MASARYK UNIVERSITY
FACULTY OF INFORMATICS

Semantic Segmentation Based On Similarity

DISSERTATION PROPOSAL

Mgr. Roman Stoklasa

Supervisor: Prof. RNDr. Michal Kozubek, Ph.D.
Consultant: RNDr. David Svoboda, Ph.D.

Brno, September 2012

Supervisor: Prof. RNDr. Michal Kozubek, Ph.D.

Supervisor's signature: _____

Contents

Chapter 1

Introduction

Computer vision is a field of study which concentrates on acquiring, processing, analyzing and understanding of images. This field has been studied for decades but there are still many opened and unsolved problems and challenges. Probably the biggest challenge to solve is the problem that computers can't "see" what is in the image — what does the image reflect.

Why computers can't perceive what is shown in the image in the similar manner like people do? Images are for computers just matrices of numbers without any semantic information. Therefore, we need many different and sophisticated methods in order to obtain at least some semantic information about the image from the computer system. In this work, we will address the problem of finding the semantics in images, of recognizing objects and scenes in the image, which has not been still sufficiently solved yet [1].

In the past few years we are observing a big growth of multimedia data available — particularly in the form of photos and videos. People like to take photos of their life and share them with others using many social networks and web portals (e.g., Facebook, Google+, Youtube, Google Picasa, Flickr, etc.). Because the amount of multimedia content grows, there is also a demand for effective search and retrieval methods that can find photos or videos based on some criteria. If the system is able to automatically recognize objects and scenes, it will be able to store additional meta-information about each image, which will be very helpful during the search. Currently, there is no such system available, so images need to be searched either based on some accompanying text information or based on visual similarity.

Text-based image search needs some text information about each image, which can be, for example, user descriptions or tags. The disadvantage is that such information is often imprecise or missing. The second option is to search images based on visual similarity but this requires giving an example query image. Unfortunately, users very often do not have such an example close at hand. Therefore, there exist also approaches which combine both the text-based image search with the content-based similarity search. Incorporating recognition of particular objects into the process can enhance search performance and shrink so-called *semantic gap* [2]. We believe, that solving the problem of semantic segmentation and object recognition can improve search performance even more.

In order to be able to recognize objects in the images, we need to solve several difficult problems. First of all, we need to find objects of interest (or candidates for objects) in the image and localize them. This problem is called segmentation and it is a well-known problem in the field of image processing [3]. When we know where the objects (or the candidates for objects) are, we need to distinguish between different types/classes of objects (to recognize

them). This can be done using classification.

Automatic segmentation is very difficult task, which is being solved for many years and decades. The main problem why there is still no general-purpose automatic segmentation algorithm is that each application has its specific needs — each application may be interested in different parts of image. For example, when we have a photo of person, some application may be interested only in the segmentation of face, whereas other application may need the segmentation of the whole body. Moreover, there are many different domains of images — you can have specific biomedical images (e.g., images from fluorescence microscope or CT images), images from some industrial camera intended for quality assurance in a factory, or you can have pictures of real world taken by somebody during holidays. This great diversity of image types causes that there can't be designed one common algorithm or approach which would solve automatic segmentation task well in all domains. Instead, each domain has its own suitable algorithms. A brief overview of such approaches will be described in section 2.1.

After segmentation of objects we need to classify each of the regions. Classification is a decision process where we decide, into which category (categories) that particular object belongs. Classification is a well-established problem, which can be solved using many different approaches, where each approach has its own pros and cons. A summary about possible classification methods will be also described in section 2.3.

1.1 Objectives of the dissertation thesis

In our work we address the problem of semantic segmentation — the problem how to divide image into segments and assign a label to each segment. If we want to solve this problem, we need to incorporate solutions to both previously mentioned subproblems — segmentation and classification. State of the art approaches use mostly machine-learning methods for classification such as Support Vector Machines, Neural Networks or decision trees. These approaches have the main disadvantage that such systems are built (trained) only for one particular problem — for recognizing only well defined and relatively small number of classes, which were known before the training phase.

In our approach we would like to use k-Nearest Neighbor (k-NN) classifier and similarity searches in knowledge-base database. This approach does not require a training phase in the same sense as other machine-learning techniques do. Knowledge for k-NN classifier is represented and stored in the database which can be built, updated or maintained by separate entity or subsystem (i.e., the classifier itself do not influence the "training" process in any way). Such database can be enhanced in the course of time — we can say that it will be possible to enhance the database almost on-line. There are many possibilities how to utilize this property — for example such system can be connected with a dialog system that will be collecting feedback from users and reflect it into the database. In this way the database can be "taught" to recognize new classes, objects and scenes based on the user feedback.

Our goal is to develop a system which will take simple image as the input and will return labeled image as the output. This system should be applicable to various domains of images.

1.2 Outline of the document

This document is divided as follows. In Chapter 2 we describe state of the art approaches to segmentation, computation of image features and classifying them. In Chapter 3 we present already achieved results and in Chapter 4 we will discuss the aims of the dissertation thesis.

Chapter 2

State of the Art

In this section we will describe the main publications and results related to our work. This description is not exhaustive, we are focusing here just on the leading directions and results in each particular subproblem.

In our work, we will deal with the problems of segmentation, classification and their combination. Therefore, we will briefly review various approaches to these problems. We will also discuss some promising research directions and applications which can be considered as related to problem of semantic segmentation. In particular, we will also review automatic image annotation systems, content-based sub-image retrieval and semantics extraction from the image.

2.1 Segmentation

Segmentation is one of the early steps in order to process image data. Its purpose is to divide image into regions, which have strong correlation with real world objects contained in the image. A very good survey of many segmentation algorithms can be found in book [3], which describes them thoroughly.

There are two types of segmentation: *complete segmentation* and *partial segmentation* [3, Ch. 6]. Complete segmentation results in image partitioning where each region corresponds with one object in the image. Partial segmentation results in regions which do not correspond directly with objects. We need some further processing in order to achieve proper complete segmentation. It is obvious, that problem of complete segmentation is very difficult for real-world images [4, Ch. 10]. Therefore, we need to accept the fact, that many segmentation algorithms gives us just partial segmentation and we need to design consecutive processing steps which can finalize the segmentation, e.g., by merging or dividing regions.

Segmentation algorithms can be also divided into two groups based on "direction" of processing to *top-down* and *bottom-up* approaches. *Bottom-up* approach starts with the pixels of image and organize them into regions. Segments are created just according to the image data. On the other hand, *top-down* approach starts with some model of objects that are expected to be in the image. This kind of algorithms tries to fit that model to the given image data based on which the segments are established.

2.1.1 Region-based segmentation

Region-based segmentation constructs regions directly using various strategies. The basic idea is to divide an image into zones of maximal homogeneity. All methods mentioned in this section are typical examples of bottom-up approaches.

Patches

The simplest segmentation algorithm is to divide image to small parts — patches. There can be various strategies how the patches can be generated, either regularly as a small squared patches [5, 6, 7], or irregularly. When generating patches irregularly, one can apply several different strategies, for instance random sampling. Sampling density can be either uniform, or it can be higher for salient parts of image [8, 9]. For example, in [10] authors uses SIFT-like keypoint detector for finding interesting parts of image and then they extract patches around each keypoint. The final segmentation can be obtained after merging neighboring patches with the same classification.

Region growth

Region growth method is another simple approach how to obtain region-based segmentation [3, Ch. 6.3]. The method starts with initial small regions (some methods start with regions initially consisting of simple pixel) and evaluates homogeneity criterion for neighboring regions. If the criterion indicates that the homogeneity is not broken even after merging neighboring regions, these regions will be merged together.

Split and Merge

Enhancement to the region growth method represents split and merge algorithm described in details in [3, Ch. 6.3.3]. Apart from the merging part, which is similar to the one mentioned above, it adds also the opposite process — splitting. When a region does not fulfill the homogeneity criterion, it is split into smalled regions. These smaller regions then can be merged with neighboring regions afterwards again.

2.1.2 Template Matching

Template matching [11] is a basic method that can be used for locating a priori known objects in the image. This method is an example of *top-down* approach. Objects are represented with models often referred to as templates, and we search for the best possible match in the image.

However, there are many aspects that should be treated and solved. The main problem is how to deal with transformations such as scale and rotation. Naïve approach is to test the template in all possible transformation (positions, rotations, scales) but this approach is very computational intensive. Therefore, several "smart" approaches were introduced to address this issue, for example in [12, 13]. The point is to represent the template and investigated

region by some feature, for example Haar-like box feature. Using this trick one can avoid slow element-by-element floating-point computations.

Another question is how to evaluate similarity between the template and the image part. The basic way is to define different similarity measures (such as Sum of Squared Differences (SSD), Sum of Absolute Differences (SAD), Normalized Cross-Correlation (NCC), Mutual Information (MI) etc.), but there exists also some other sophisticated methods. Template can be divided into small parts (patches) which are positioned relatively to the reference point of the whole template [14]. When best possible matches of all patches are determined individually, patches' relative positions can vary from its original positions in the template. This flexibility helps to cope with decent transformation or distortion of objects in the image compared to the template.

The problem of template-matching can be very time-consuming, especially when a large set of possible transformations are taken into account.

2.1.3 Segmentation by Composition

An interesting approach to segmentation was described by Bagon et al. [15]. They define a good image segment as the one which can be easily composed from its own pieces, while it can be composed very hardly from regions in other segments. This method can be used also for class-based segmentation — we can define some sample images of object which we would like to segment and the algorithm will find all regions, that can be composed using regions from these samples.

2.1.4 Contour Detection

Contour detection is a typical task in computer vision, it is similar to the edge detection. Its purpose is to detect borders of objects in the image. The difference between contours and edges is that edges correspond to variation of intensity values, whereas contours should correspond to salient objects.

Contour detection is a dual task to segmentation. When we have segmentation, we can always obtain closed contours from the segments' boundaries. Unfortunately, contours need not be closed so the opposite process to obtain regions from contours is more complicated [16, 17].

There can be found many publications dealing with contour detection algorithms or that use contour detection for some higher-level tasks. In the last few years, various methods for contour detection algorithms were published [18, 19, 20, 21, 22, 23, 24, 25]. Arbelaez et. al [26] claimed that their contour detection method outperforms any previous approaches and reaches the state-of-the-art performance.

2.1.5 Hierarchical Segmentations

One of the biggest problem of automatic image segmentation is to determine the level of details of segmented regions, which affects mainly bottom-up approaches. Let us image that

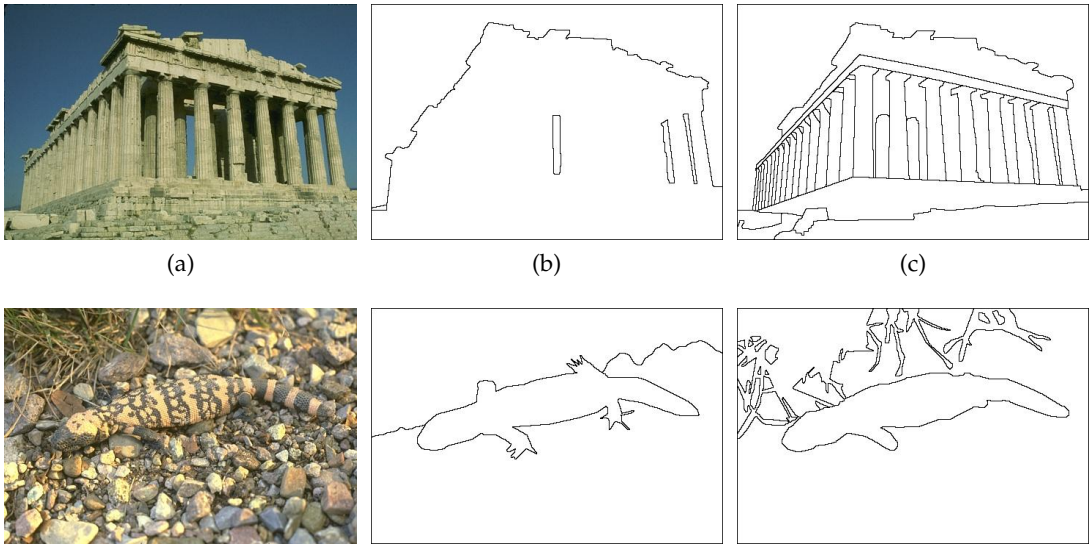


Figure 2.1: Two examples from BSDS300 dataset [27] with 2 different human-defined segmentations which show that different levels of details are important for different persons. (a,d) Original images. (b,e) Coarse level regions. (c,f) Fine level regions.

you have a photo of some object on the gravel background like in Figure 2.1d. If the image does not contain any metadata it is hard to determine if the user would like to segment each stone (for example, because he/she would like to count them), or if the gravel should be segmented as one region (background). This problem is fundamental and is present also in human-made segmentations. Figure 2.1 shows two examples from The Berkeley Segmentation Dataset (BSDS300) [27] with related segmentation from 2 different persons. It can be seen that different users may prefer different level of details even for the same image.

To deal with this issue, hierarchical segmentations were introduced. Hierarchical segmentations divide image into regions for different level of details. Typically, there is a tree structure which contains the information how regions can be merged when moving from fine-level of segmentation to more coarse-level. With this structure, one can tune desired level of details.

There exist several methods how to compute hierarchical segmentations. One of them is based on the watershed transform [28]. Disadvantage of the watershed transform is that it tends to over-segment the image [3, Ch. 6.3.4], thus various techniques were introduced to deal with this problem [29]. Basically, there are several variants of methods for region merging and their description can be found, e.g., in [30]. Another possibility is to create the hierarchical regions from the output of any contour detector as was shown in [31]. Arbelaez et al. [26] published its state-of-the-art algorithm based on their (hierarchical) contour detector algorithm.

2.2 Image features

After we have extracted image regions, we need to describe their properties, which are called features. As a feature one can take almost any numerical property that can be computed based on the image data. It can be, for example, some statistical information (such as mean value of pixel intensities, statistical moments, etc.), texture properties (such as gradient histogram, homogeneity, etc.) or shapes of the regions (such as curvature, smoothness of boundary, etc.).

Features are extracted using image descriptors, which can be generally divided into two groups: global descriptors and local descriptors. Global image descriptor extracts features based on the whole input image while local descriptor typically describes just a small region surrounding interesting point in the image.

In this section we will review very briefly some types of features that can be used for describing regions either from real-world images or from biomedical ones. We will pay special attention to the texture descriptors, because they are often used for characterizing individual regions from the initial segmentation of image.

2.2.1 Global features

There is a range of different types of descriptors, from well-known and well-established to the proprietary ad-hoc solutions applicable for a particular application only.

Probably the most famous global descriptors are a part of the MPEG-7 standard [32]. This standard defines many different descriptors suitable for multimedia content (images, video, audio). Visual part of standard contains tools for description of color information, shapes, texture properties and other. This type of descriptors is used quite commonly, e.g., in [33], [34] or [6].

Ad-hoc descriptors are used also quite often, for example, to compute different special characteristics of images from selected application domain. If the dataset one is working with contains images with some common property, it can be very useful to design ad-hoc descriptor for this particular property, because general well-known descriptors are not able to describe that characteristic properly.

2.2.2 Local features

Apart from the global features mentioned above, local descriptors describe only a small portion of image near some specified point, usually termed as keypoint.

Computation of local features consists of two basic stages: 1) key-point detection and 2) region description. In the first stage, the set of interesting points is extracted from the image. Decision what is an interesting point depends on many factors and is also dependent on application — it can be, for example, corners, blobs or some extremal regions (with minimal or maximal intensities). Many detectors can be found in the survey by Tuytelaars and Mikolajczyk [35]. However, there are several common and favorite key-point detectors widely

used: Harris-Affine & Hessian-Affine [36], Maximally Stable Extremal Regions (MSER) [37], Intensity extrema based detector (IBR) [38], Edge based detector (EBR) [38] and Salient region detector [39].

In the second stage, the surrounding region near to each key-point is described. Common local descriptors often seen in the literature include: Scale-Invariant Feature Transform (SIFT) [40, 41], Speeded Up Robust Features [42] and Histogram of Oriented Gradients [43, 44].

Local features are very popular for content-based image retrieval and also for classification (e.g. [45, 46]).

2.2.3 Texture-based features

Texture based features are quite common and their purpose is to describe some properties of textured areas. These descriptors can be applied either to the whole image (e.g., when the image is small and contains just one object of interest) or just to the selected region (such as one segment from initial segmentation).

Several texture descriptors were published that can be used also for biomedical images (often for purpose of classification). These include, for instance, Zernike features [47] and Haralick features [48], which were successfully used for example for classification of cellular protein localization patterns [49] or to statistical description of rock texture [50].

Another texture-based descriptor is called Tamura features [51] which consists of 6 different attributes: coarseness, contrast, directionality, line-likeness, regularity and roughness.

Local Binary Pattern (LBP) [52] technique is based on the idea of texture units which were introduced by Wang and He [53]. They stated that a texture image can be decomposed into a set of essential small units called texture units. Texture unit is typically represented by 3×3 window. Using these texture units, each texture can be described by its texture spectrum.

Hung et al. in their work [54] compared the performance of Texture Spectrum and LBP in texture classification. Both features performed with approximately identical accuracy.

2.3 Classification

Classification is a process of identification a category (or set of categories), which an examined object belongs to. Classification is closely related with object recognition — if we can recognize some object, we can also classify it; and when we can classify some object, we can say that we are able to recognize it in some way.

Classification is a common method used in various types of applications. For example in biomedical image domain, Boland and Murphy presented that it is possible to use classifier successfully for different subcellular patterns on the images from fluorescence microscope [55]. What is more interesting, Murphy et al. [56] showed, that automatic classification can reach better performance than the classification performed by humans in certain domains. Clearly, this result motivated further development of classification methods for new appli-

cation areas.

Classifiers can be divided into two basic groups: supervised and unsupervised [3]. In supervised approach the classifier is provided with the samples (or definition) of each class. This type of approach is suitable when we know "what we are looking for". On the other hand, unsupervised classifiers are helpful when we do not know exactly how classes should be defined. Classes are extracted from training data set which should be partitioned into several different classes.

2.3.1 Overview of Classification Methods

Cluster Analysis

Cluster analysis is one group of methods for unsupervised classification, which means that such methods do not need any "teacher" during learning phase. Instead of the teaching, these methods learn themselves from the dataset. There exists basically two types of cluster analysis methods: hierarchical and non-hierarchical clustering. Hierarchical methods constructs a tree of clusters — each cluster (subset of dataset) can be divided into smaller clusters. Non-hierarchical methods just divide dataset into desired number of clusters. Non-hierarchical algorithms can be either parametrized or not.

k-Means

Non-parametric non-hierarchical cluster analysis is quite popular and a simple approach. One can use for example simple MacQueen k-means cluster analysis algorithm [57] to partition dataset into k distinct clusters. Basically, the parameter k needs to be known in advance to the processing. Moreover, there exist also some methods which can estimate k from the dataset [3, Ch. 9.2.5].

SVM

Probably the most commonly used supervised classification method is a Support Vector Machine (SVM) [58]. SVM was originally designed for binary classification — for linear separation of vectors into two classes. However, later research reported extensions allowing for non-linearly separable classes, non-separable classes and combination of multiple binary classifiers to perform multi-class classification. An interesting idea how to add multi-class support by combining k-Nearest Neighbor classifier was published in [59].

Basic principle of binary SVM classifier is to find such linear discrimination function which will separate all vectors (called *support vectors*) into the two classes. The discrimination function forms a hypersurface for general n-dimensional feature space, i.e. line in 2-dimensional feature space, surface in 3D feature space, etc.). Discrimination function maximizes the margin between both classes [3, Ch. 9.2.4], where margin is defined as the distance between the discrimination hypersurface and the closest training sample.

When somebody wants to build a SVM classifier, one needs to select proper kernel function (which transforms the problem of linear separability to non-linear one) and also needs to supply training samples for both classes. In practice, positive and negative examples are often used because SVM classifier often serves as a decision-maker whether the object belongs to particular class or not.

It is important to notice, that the training phase should be carried out before the classifier is able to classify first query object. Unfortunately, it is not possible to extend the training samples during the life of SVM classifier — the only possibility is to re-run the training phase from the beginning for the whole updated training features set.

The main disadvantage of the SVM method we find in the following points: non-natural support of multi class classification (which can be very prohibitive for large number of classes) and the inability to update the training set during the "life" of the classifier (in order to add new examples or to add brand new class).

Neural networks

Neural networks can also be used for classification. Its ability to learn specific patterns can be utilized with advantage as was shown in [49] or [60]. Details about neural networks for object recognition can be found, for example, in the book [3].

Feed-forward neural networks have similar disadvantages to the SVM classifiers: one has to train classifier completely prior to its first usage. Moreover, training phase for neural network can converge very slowly, so one needs a lot of training samples and it is time-consuming. Neural networks are also sensitive to overfitting problem [61].

However, there exist also self-organized neural networks that are able of unsupervised learning, e.g. Kohonen feature map [62]. This type of networks is performing the role of clustering, so similar inputs produce the same output [3, Ch. 9.3.2].

Decision trees, Random forests

Decision tree [63] is a data structure that can be used for classification. Leaves of such tree represent classes (or class labels) and all internal nodes represent some decision criterion. It is typical that one variable is evaluated at each level of the tree — in general we can interpret each internal node as a classifier according to only one feature. Classification process runs from the root to the leaf through internal nodes.

Random forest is a multi-way classifier, which consists of several different trees. These trees differ from previously defined decision trees in the value of leaves — leaf of each tree in the random forest contains posterior distribution over all classes. Each tree is built with some form of randomization, which can be basically of two different types: either the trees can be "grown" (learned) on a different subsets of training set or there can be differences in the evaluation nodes (e.g., different subset of features evaluated, different order of evaluation, etc.). An example of using random forest classifier for classification of images is available in [64].

Genetic Algorithms

Genetic algorithms are well-known as optimization techniques, but this type of algorithms can also be used for the image understanding problems (as shown, for example, in [3]) in image processing. This method works on the basis of hypothesize and verify principle. Genetic algorithm is responsible for generation of new segmentations, each of which bound with some hypothesis. Some objective function verifies which of the hypotheses are good and which are not. Therefore we can look at it as it is an optimization problem for the objective function.

At the beginning the image is over-segmented — regions of this segmentation are called *primary regions*. Primary regions are repeatedly merged together to form current segmentation. Genetic algorithm forms new feasible segmentation from the the current population and forms new hypothesis (assignment of labels to segments).

An example application of such evolutionary principles for image classification tasks can be seen in [65] and [66].

k-Nearest Neighbor

Nearest-Neighbor (NN) classifier is a simple non-parametric classifier that relies on distance evaluation between objects (features). In the simplest form, this classifier works as follows: for the query object (feature) the closest neighbor is found in the training set. Query object is assigned the same label as has its nearest neighbor. Some of the biggest advantages are: (i) there is no training phase and the training set (or the "knowledge base" for classification) can be updated anytime. (ii) NN classifier is also resistant to overfitting problem [67]. (iii) Can naturally handle a huge number of classes.

However, this type of classifier seems not to be very popular (compared to, e.g., SVM). In spite of this fact Boiman et al. [67] showed that Nearest-Neighbor classifier can achieve as good performance as other methods.

k-Nearest Neighbor classifier is a modification that takes into account up to k neighbors of query object. Label for the query object will be derived from labels of neighbors. An example of a possible aggregation function which produces classification estimate is described in [6]. An example of k-NN classifier based on local features is shown in [46].

2.3.2 Feature selection

A common problem in classification is that one can design arbitrary number of different image features but one doesn't know which of them are better than others for describing the classes, which of them do have the best discriminative property and which of them are noisy. In general, feature selection addresses the problem of selecting the most informative features among all given ones.

When a large number of features is used it leads to high-dimensional problems whose solutions are often inefficient. Moreover, one can encounter a *course of dimensionality* — a term used to express the exponential growth of the complexity as a function of dimensionality

(this term was firstly described by Bellman in [68]).

Principal Component Analysis

Several techniques were introduced to reduce the dimensionality in order to fight this problem with high-dimensional data. The basic one is called principal component analysis (PCA) (also known as *Karhunen-Loève transform* or *Hotelling transform*). PCA simplifies high-dimensional data by identifying new coordinate system in which the vectors will be expressed. This new coordinate system has the property, that its basis vectors follow modes of greatest variance in the data. Then one can take into account just the most important basis parameters and the rest can be left out because they provide least significant information.

Visual Words

Visual Words (or visual dictionaries) is a concept how to deal with a huge number of high-dimensional features. It is often used in conjunction with local features like SIFT, SURF or HOG. The problem is, that one can extract hundreds or even thousands of local features from a typical real-world image. Each feature is a vector with high number of dimensions (for example 128 in case of SIFT), which leads to relatively sparse density of extracted features inside the whole feature space. The huge amount of features for large collection of images is a problem for efficient evaluation, indexing and searching — e.g., when somebody wants to index and search in a collection consisting of 100 millions images and from each image in average 300 features is extracted, one needs to handle 30 billion of features).

Therefore a technique of deriving visual words (visual dictionaries) was introduced [69]. This method is based on clustering. When all local features are extracted from the training set, features are clustered into many clusters (typically several hundreds or thousands). Each local feature is then represented by its cluster to which it belongs. These clusters are called "visual words" because there is a similarity with textual documents — each image contains many visual words (one visual word for one detected local feature) in the same way like each textual document consists of many words. This principle is often referred to as *bag of words*.

One of the biggest advantages of this approach is the ability to use inverted files for indexing of images in the similarity database and the possibility for very efficient image retrieval. Usage of visual vocabulary method can be seen in many publications including [70, 71, 72].

Fuzzy-Rough Feature Selection

Another way to select features is, so called, Fuzzy-Rough Feature Selection method (FRFS) [73]. It is based on fuzzy [74] and rough [75] sets. It reduces discrete and/or real-valued noisy data. An example of using FRFS in combination with neural networks and k-NN classifier can be found in [76].

Boosting

Boosting is a technique how to combine several weak classifiers in order to obtain the strong one [77]. It is related to the feature selection in the sense that, while the feature selection algorithm is choosing the most discriminative features, boosting algorithm is choosing how to combine basic classifiers. A popular algorithm for boosting is called AdaBoost and was introduced in [78]. There exist also many variations and extensions to the basic boosting algorithm, for example, probabilistic boosting-tree [79] and multi-modal and hierarchical boosting [80]. Liu et al. [81] uses boosting for combining features extracted from image content and from the EXIF metadata in their "indoor/outdoor" image classifier.

2.3.3 Similarity Evaluation

Evaluation of image similarity can also be used for image classification. Typical situation is that one has a database of known images (image samples) which are already classified or labeled. In order to classify unknown image, one finds the most similar image (images) in the database and infer resulting class from them.

The most simple approach is to use one-to-one similarity — i.e., compare the query image to each image in the database separately. More advanced approach (which is claimed to be more accurate [82, 67]) is to compute the one-to-multiple similarity (also referred as *Image-to-Class* distance) and was introduced by Boiman et al.

Shechtman and Irani [83] reported approach for evaluation of similarity between images based on the local self-similarity descriptor applied to both images. They showed promising results as their approach is able to cope with large differences of photometric properties between images.

2.4 Semantic Segmentation and Objects Recognition

Semantic segmentation of an image is such segmentation which groups pixels together by their common semantic meaning. Each segment is assigned a label that denotes a semantic meaning of that segment. In this section, we will briefly review several possible approaches published in the recent years.

Region-based

A majority of published approaches are so called region-based. A common property of region-based semantic segmentation methods is that they start with some initial segmentation (even with possibly oversegmented image).

Arbelaez et al. [1] use hierarchical segmentation algorithm for generating initial regions. After that, they are use several SVM classifiers to assign labels to regions, which leads to final semantic segmentation.

A slightly different approach based also on regions was published in [84]. They created bag of regions for each object (generated from region tree of their hierarchical algorithm)

and use generalized Hough voting scheme to estimate object position in the image.

In [85], authors used initial segmentation that splits image into many almost homogeneous regions (called fragments). These fragments are then labeled and merged together based on the classification of each fragment.

It is obvious that broader visual context can be very helpful when classifying some region. For example, in [86] authors used ancestral sets of regions (also obtained from hierarchical segmentation algorithm) to enhance precision of classification. As a classification algorithm they used 3 different methods: Logistic Regresion, SVM and Rank Learning.

Contour-based

Apart from region-based approaches there were published also several methods that use contours instead of regions. In [87], authors used contours to detect objects and background in images. Hariharan et al. [88] proposed semantic contour detection algorithm.

2.5 Related applications

In this section, we would like to mention additional research areas that are somehow related or can be exploited with advantage when dealing with the problem of semantic segmentation. Namely, we will discuss automatic image annotation systems and content-based sub-image retrieval.

2.5.1 Automatic annotation systems

Visual Concept Detection and Annotation task (VCDA) [89] is a relatively new field of research that has emerged in recent years. Visual Concept Detection is closely related to the semantic segmentation and image classification task. However, Visual Concept Detection works at a coarse level and processes image as a whole while semantic segmentation works on finer level — objects need to be detected and labeled within the image. Visual Concept Detection often uses many global cues and is not so tightly related to objects in the image.

For an overview of present state-of-the-art VCDA methods one can look into the report from ImageCLEF 2011 contest [90]. We believe that an automatic image annotation system can be very helpful as an "oraculum" because it can provide valuable context of the image.

2.5.2 Content-based search and retrieval

Content based image retrieval task deals with the problem how to find visually similar images from some (potentially large) collection (e.g., database). Similarity between images can be defined using many different descriptors or metrics. A survey about different recent approaches can be seen in [91]. There are also attempts to develop methods for efficient search for sub-images — i.e., to find all images in collection that contains query image as its sub-image [92, 93].

Similarity search for images and sub-images offers promising possibilities also for classification, especially in conjunction with k-NN classifier and large collections of annotated image databases. ImageNet [94] is an example of such database — it is a largescale ontology of images built upon the backbone of the WordNet structure [95].

Chapter 3

Achieved Results

In this chapter, we would like to summarize and show our current results that are relevant to the future research. In the first one — the road detection application — we have demonstrated that a combination of segmentation and classification for each region is useful and can achieve good accuracy (even for very simple segmentation methods). In the second project — the HEp-2 Cell Classifier — we have proven that relatively precise k-NN classifier can be built also for biomedical images. In the third one — Sister Cells Classification — we designed visual similarity measure between cells in the images from microscope.

3.1 Road detection

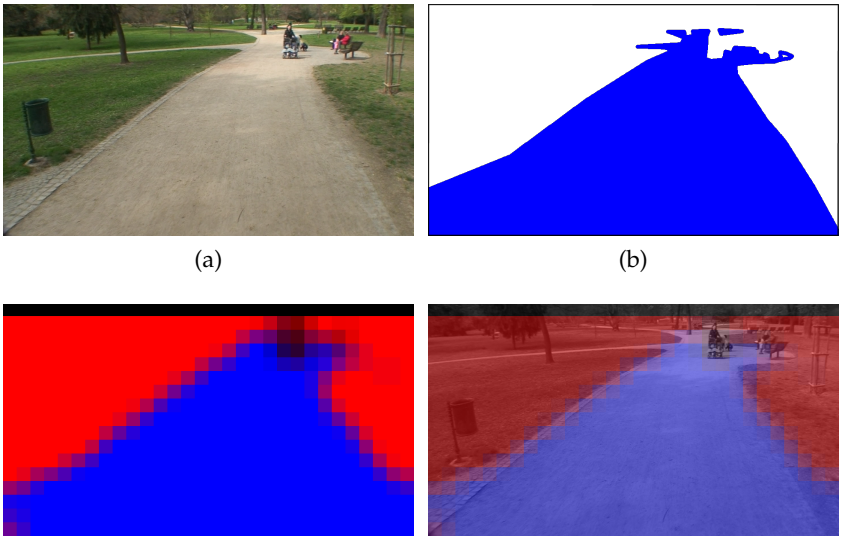
In this project, we were dealing with the problem of road detection for an autonomous robot. The superordinate problem is to develop an autonomous robot which will be able to navigate and travel in the natural outdoor environment from point A to point B. This problem would not be so difficult nowadays when GPS receivers are very common. The difficult part is the rule that such robot has to follow roads and pathways and is not allowed to slip out of the road. In this situation one needs to ensure that the robot is able to follow the roads accurately. It appeared that best possibility was to process visual information for this purpose.

We developed new method for road detection from the visual information based on the similarity search. In our approach, we combined naïve image segmentation with the k-NN image classifier. During the segmentation, the image was divided into regular overlapped squared regions. Each region was then classified whether it is more similar to road or non-road. This classification was based on the similarity search for most similar examples in the database ("knowledge-base"). In this database, we had samples of various road and non-road textures.

Figure 3.1 shows an example result from our algorithm. Detailed explanation of the algorithm and achieved results can be found in [6], which is also included in Appendix B of this thesis proposal.

3.2 HEp-2 Cell Classifier

In this project, we tried to develop and build a classifier, which will be competitive and will achieve good result in the international contest on HEp-2 Cells Classification hosted by the



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

Figure 3.1: Results of our road detection algorithm: (a) Input frame from the camera. (b) Manually defined ground-truth for the frame (blue area represents road). (c) Computed classification map. Topmost black bar is unclassified margin of the image. (d) Classification map overlaid over input frame.

21th International Conference on Pattern Recognition, ICPR 2012[1]. Aim of this competition was to compare different state-of-the-art methods for classification of this type of data on the common large dataset.

The dataset contained 6 different classes of cells (see Fig. 3.2) which were already pre-segmented (each cell image was provided with the segmentation mask). The evaluation criterion for the contest was to achieve the highest accuracy.

We developed two versions of our classifier[2] which differ in the set of used image descriptors. In the first version, only global descriptors were used to describe image features while in the second version we also tried to use local features. These local features were computed by the combination of MSER keypoint detector and SIFT region descriptor. We used the following global descriptors: LBP, Haralick features, Color Structure (from MPEG-7 specification), granulometry-based descriptor (which expresses the distribution of structures of different sizes in the image) and one ad-hoc descriptor which described intensity differences in neighboring pixels.

Both versions of classifier were an implementation of the k-NN approach with the training examples stored in the database. The classification itself consisted of 4 stages: preprocessing; k-NN search for similar training images in the database; aggregation of information

1. Contest homepage: http://mivia.unisa.it/hep2contest/index.shtml
2. Software homepage: http://cbia.fi.muni.cz/projects/hep-2-cells-classifier.html

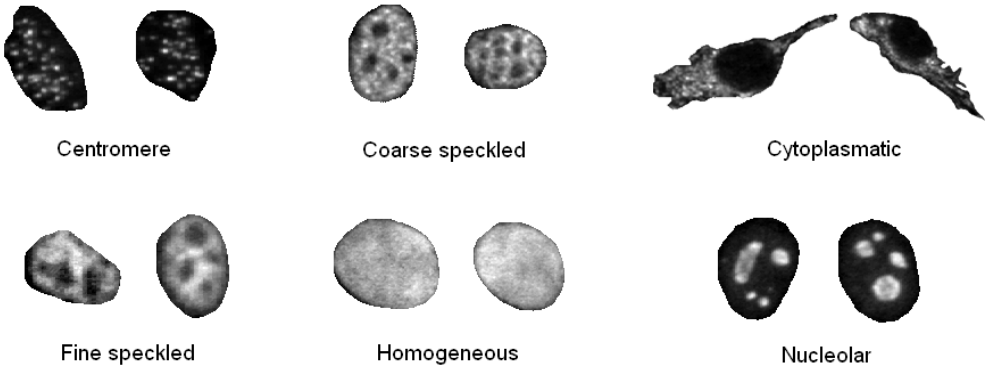


Figure 3.2: Examples of different HEp-2 cell classes (shown in the grayscale)

obtained from the database to compute partial class estimate; and finally we combine several partial class estimates into the final one.

Unfortunately, the performance of our classifier on the testing set as compared to others is not known at the time of writing of this thesis proposal. However, our classifier was able to classify approx. 97% of images from the training dataset. This result is not biased because each tested image was excluded from the database for that particular test.

3.3 Sister Cells Classification

Our task in this project was to develop a measure that would be able to confirm a claim that two cells are sisters based on their visual similarity.

We were given a set of images from fluorescence microscope and we developed a similarity measure based on our tools and knowledge experienced in HEp-2 Cell Classifier project. We computed several descriptors based on image texture, namely local binary patterns [52] and Haralick features [48], and on the shape of image structures. We also computed a granulometric curve [96], which expresses the distribution of structures of different sizes in the image. We defined several aggregated distance functions for different combinations of these descriptors.

Such similarity measure turned out to be able to distinguish between sister and non-sister cells (verified on ground-truth examples), so it was used as a support for the claim that some cells were sisters because they contained similar nuclear structures.

Full article [97] can be found in Appendix C.

Chapter 4

Aims of the Thesis

4.1 Objectives

Basically, solving image recognition problem is not only very challenging but also a very difficult and still open task. There is no common approach which would have reasonable performance and results across different image domains. There exist, however, a number of different approaches and methods but each solves just some narrow area of interest. Moreover, each problem has its suitable methods for solving.

Although an idea of general image recognition software is very tempting, we need to start with small steps and specify our domain of interest as a narrow field. However, we would like to keep in mind the idea for possible generalization, which may have some influence on our future decisions. There are basically two reasonable domains which can be taken into consideration in the research group I am a part of — real world images or biomedical images.

In the nearest future of our research we plan to continue with development of our ideas presented in the road detection algorithm [6]. This algorithm is already a prototype implementation of more general principle which can be used for semantic segmentation. We plan to generalize and enhance the method with several ideas such as (i) addition of multi-class classification support; (ii) enhancing of segmentation, possibly with the use of hierarchical segmentation method; (iii) speed optimization in order to enable deployment to a real autonomous robot. All this effort will be done with the aim of publication in impacted journal or in high-quality international conference on pattern recognition or robotics applications.

Proposed principle, which we believe is suitable for semantic segmentation and is already used in our road detection solution, can be described as follows. Processing will start with chosen segmentation method to generate candidate regions. In our research, we will not focus on development of a segmentation algorithm but rather use some already available one which will provide sufficiently good results. Which results will be "good enough" will depend on the application (for one application a simple grid-segmentation can be sufficient, for other a more sophisticated and more precise segmentation may be needed).

After that we will ask a simple question for each region: "What does this region look like?" The answer to this question will be obtained using k-NN classifier, which will answer it based on the evidences in some annotated database (e.g., in training samples). There is also a possibility of other, more sophisticated questions we may ask in the future: "What does this merged region look like?"; "Can we find more known patterns when we merge

these regions?"; "If there is a region of type A in the image, can we confirm a hypothesis that some other region is of type B?" (for example: "If there is a human face in the image, can this neighboring region be a human body?").

After answering such questions for each region, we can make final conclusion about scene in the image. We can label recognized objects as well as we mark some regions as unknown objects/structures. By using k-NN classifier, we are able to evaluate reliability of classification. If the classifier is "not sure", we can avoid the risk of misclassifying such region.

We believe that this principle is universal enough to be applied also to other image domains, in particular to biomedical images. In fact, our second goal in the research will be to show its usability for the purpose of detecting and classifying different cell types in the images from microscope. This work will be based on our results in the HEp-2 Cell Classification project (see Section 3.2). Results from this project proved that we are able to achieve decent performance using our approach with the k-NN classifier in the biomedical image domain as well.

To achieve this goal, we will need to extend the classifier with suitable segmentation algorithm, which will be able to detect and segment each cell from original image from microscope. For testing and evaluation of our solution, we will use the same dataset [98] that was used in the HEp-2 Cell Classification Contest, but we will work with non-segmented original images. This dataset will be publicly available after the contest ends in November 2012. This work is also intended to be published either in journal or in one of the conferences on biomedical image processing or pattern recognition.

After successful achievement of above goals, we plan to move forward to more general domain of problems, such as segmenting and labeling of real-world images. We would like to show that our concept is a competitive variant to the semantic segmentation algorithm published by Arbelaez et al. [1]. As a benchmark we will use the same PASCAL [99] dataset which was also used in [1]. We hope that such comparison and our results would allow us to publish our work in some high-quality research platform such as IEEE Conference on Computer Vision and Pattern Recognition.

We plan to achieve all of the mentioned goals thanks to flexible design of processing framework. Our system will be divided into modules which will be cooperating together. For example, there will be modules dealing with segmentation, modules dealing with extraction of keypoints, computing features of regions, etc. Obviously, this allows to have several modules designed with the same purpose, e.g., segmentation algorithms for extracting several cell types, so that we can employ optimal configuration of modules for the given application. There is also a possibility to use some automatic image annotation tool as one of the modules, which can serve as an "oraculum" for visual concept detection.

4.2 Future visions

Our research will be done with respect to our distant vision of a complex automated system, which would be able to recognize and describe what is contained in the images. We know

that this goal is very ambitious and it can take many years to solve it. It is clearly beyond one doctoral study. However, we would like to build at least some basic stones in this way.

One of the biggest advantages of our proposed approach compared to the solution in [1] will be the long-term usability. Approaches based on SVM or other machine-learning methods need to be trained prior to their life and it is very difficult to update their "knowledge" (you often need to run the whole training phase again). This property makes them less suitable for implementation in a long-term applications, such as web portals or web services for image recognition.

Instead, k-NN classifier depends only on the knowledge-base (database of known samples and metadata) which can be taught and updated almost on-line. We can imagine that, in the future, there can exist web application which will allow users to describe their images (i.e., application which will allow users, or other computer systems, to submit arbitrary image and this application will return labeled image with recognized objects) and whose knowledge-base could be improved as the time goes. We can also imagine that such web service would have great potential and added value for many 3rd party applications working with multimedia content.

The ability of learning during the lifetime of such service also offers the possibility of connecting with some feedback-giving system. We can imagine that in connection with a dialog system it would be possible to learn new types of objects and visual concepts as well as enhance precision of detection for already known objects based on user feedback.

However, all these visions and ideas are very complex and they are not parts of our dissertation thesis. We are discussing them here just to show our motivation for dealing with this research area.

4.3 Study plan

In this section, we provide the following schedule of the future research:

Autumn 2012

- Evaluation of possible segmentation algorithms
- Analyzing possibilities to utilize existing sub-image search and image annotation tools
- Enhancing the Road Detection algorithm
- Publication about HEp-2 Cell Classifier submitted to international conference on pattern recognition or biomedical image processing

Spring 2013

- Erasmus stay at Birmingham City University (BCU) for 5 months
- Cooperation with experts on image segmentations at BCU

- Publication about enhanced road detection algorithm submitted to journal or international conference

Autumn 2013

- Implementation of segmentation and classification tool for biomedical images
- Implementation of semantic segmentation tool comparable to [1]

Spring 2014

- Publication about application for biomedical images and/or comparison of our approach with [1]
- Extensive testing of developed software
- Completing the dissertation thesis

Bibliography

[1] P. Arbelaez, B. Hariharan, C. Gu, S. Gupta, L. Bourdev, and J. Malik, "Semantic segmentation using regions and parts," *2012 IEEE Conference on Computer Vision and Pattern Recognition*, vol. 0, pp. 3378–3385, 2012.

[2] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 12, pp. 1349–1380, Dec. 2000. [Online]. Available: http://dx.doi.org/10.1109/34.895972

[3] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision*. Thomson-Engineering, 2007.

[4] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2nd ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2001.

[5] A. Kårsnäs, A. L. Dahl, and R. Larsen, "Learning histopathological patterns," *Journal of Pathology Informatics*, no. 2:12, 2012.

[6] R. Stoklasa and P. Matula, "Road detection using similarity search," in *2nd International Conference on Robotics in Education*, R. Stelzer and K. Jafarmadar, Eds., Vienna, 2011, pp. 95–102, ISBN 978-3-200-02273-7.

[7] *Discriminative training for object recognition using image patches*, vol. 2, Jun. 2005. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2005.134

[8] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, Nov. 1998. [Online]. Available: http://dx.doi.org/10.1109/34.730558

[9] L. Itti and C. Koch, "A saliency-based search mechanism for overt and covert shifts of visual attention," *Vision Research*, vol. 40, pp. 1489–1506, 2000.

[10] D. Gokalp and S. Aksoy, "Scene classification using bag-of-regions representations," in *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, june 2007, pp. 1 –8.

[11] R. Brunelli, *Template Matching Techniques in Computer Vision: Theory and Practice*. Wiley Publishing, 2009.

[12] J. Miao, B. Yin, K. Wang, L. Shen, and X. Chen, "A hierarchical multiscale and multiangle system for human face detection in a complex background using gravity-center template," *Pattern Recognition*, vol. 32, no. 7, pp. 1237 – 1248, 1999. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320398001563

[13] F. Tang and H. Tao, "Fast multi-scale template matching using binary features," in *In IEEE WACV*, 2007.

[14] O. Boiman and M. Irani, "Detecting irregularities in images and in video," *Int. J. Comput. Vision*, vol. 74, no. 1, pp. 17–31, Aug. 2007. [Online]. Available: http://dx.doi.org/10.1007/s11263-006-0009-9

[15] S. Bagon, O. Boiman, and M. Irani, "What is a good image segment? a unified approach to segment extraction," in *Computer Vision – ECCV 2008*, ser. LNCS, D. Forsyth, P. Torr, and A. Zisserman, Eds., vol. 5305. Springer, 2008, pp. 30–44.

[16] J. H. Elder and S. W. Zucker, "Computing contour closure," in *Proceedings of the 4th European Conference on Computer Vision-Volume I - Volume I*, ser. ECCV '96. London, UK, UK: Springer-Verlag, 1996, pp. 399–412. [Online]. Available: http://dl.acm.org/citation.cfm?id=645309.757447

[17] D. W. Jacobs, "Robust and efficient detection of salient convex groups," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 1, pp. 23–37, Jan. 1996. [Online]. Available: http://dx.doi.org/10.1109/34.476008

[18] J. Malik, S. Belongie, T. Leung, and J. Shi, "Contour and texture analysis for image segmentation," *International Journal of Computer Vision*, vol. 43, no. 1, pp. 7–27, 2001.

[19] X. Ren, "Multi-scale improves boundary detection in natural images," in *Proceedings of the 10th European Conference on Computer Vision: Part III*, ser. ECCV '08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 533–545. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-88690-7_40

[20] G. D. Joshi and J. Sivaswamy, "A simple scheme for contour detection," in *VISAPP (1)'06*, 2006, pp. 236–242.

[21] P. Dollar, Z. Tu, and S. Belongie, "Supervised learning of edges and object boundaries," in *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*, ser. CVPR '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 1964–1971. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2006.298

[22] J. Mairal, M. Leordeanu, F. Bach, M. Hebert, and J. Ponce, "Discriminative sparse image models for class-specific edge detection and image interpretation," in *Proceedings of the 10th European Conference on Computer Vision: Part III*, ser.

ECCV '08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 43–56. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-88690-7_4

[23] P. Felzenszwalb and D. McAllester, "A min-cover approach for finding salient curves," in *Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop*, ser. CVPRW '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 185–. [Online]. Available: http://dx.doi.org/10.1109/CVPRW.2006.18

[24] Q. Zhu, G. Song, and J. Shi, "Untangling cycles for contour grouping," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, oct. 2007, pp. 1 –8.

[25] X. Ren, C. Fowlkes, and J. Malik, "Scale-invariant contour completion using conditional random fields," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 2, oct. 2005, pp. 1214 –1221 Vol. 2.

[26] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2010.161

[27] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th Int'l Conf. Computer Vision*, vol. 2, July 2001, pp. 416–423.

[28] J. Serra, *Image Analysis and Mathematical Morphology*. Orlando, FL, USA: Academic Press, Inc., 1983.

[29] S. Beucher, "Towards a unification of waterfalls, standard and p algorithms," Center of Mathematical Morphology, Mines ParisTech, Tech. Rep., 2012. [Online]. Available: http://cmm.ensmp.fr/~beucher/publi/Unified_Segmentation.pdf

[30] F. Meyer, "An overview of morphological segmentation," *IJPRAI*, pp. 1089–1118, 2001.

[31] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, "From contours to regions: An empirical evaluation," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. Ieee, 2009, pp. 2294–2301.

[32] P. Salembier and T. Sikora, *Introduction to MPEG-7: Multimedia Content Description Interface*, B. Manjunath, Ed. New York, NY, USA: John Wiley & Sons, Inc., 2002.

[33] J. Pakkanen, A. Ilvesmäki, and J. Iivarinen, "Defect image classification and retrieval with mpeg-7 descriptors," in *Proceedings of the 13th Scandinavian conference on Image analysis*, ser. SCIA'03. Berlin, Heidelberg: Springer-Verlag, 2003, pp. 349–355. [Online]. Available: http://dl.acm.org/citation.cfm?id=1763974.1764028

[34] E. Spyrou, H. L. Borgne, T. Mailis, and E. Cooke, "Fusing mpeg-7 visual descriptors for image classification," in *In International Conference on Artificial Neural Networks (ICANN*. Springer, 2005, pp. 847–852.

[35] T. Tuytelaars and K. Mikolajczyk, "Local invariant feature detectors: a survey," *Found. Trends. Comput. Graph. Vis.*, vol. 3, no. 3, pp. 177–280, Jul. 2008. [Online]. Available: http://dx.doi.org/10.1561/0600000017

[36] K. Mikolajczyk and C. Schmid, "Scale and affine invariant interest point detectors," *International Journal of Computer Vision*, vol. 60, no. 1, pp. 63–86, 2004. [Online]. Available: http://lear.inrialpes.fr/pubs/2004/MS04

[37] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in *In British Machine Vision Conference*, vol. 1, 2002, pp. 384–393. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.7.2484

[38] T. Tuytelaars and L. Van Gool, "Matching widely separated views based on affine invariant regions," *Int. J. Comput. Vision*, vol. 59, no. 1, pp. 61–85, Aug. 2004. [Online]. Available: http://dx.doi.org/10.1023/B:VISI.0000020671.28016.e8

[39] T. Kadir, A. Zisserman, and M. Brady, "An affine invariant salient region detector," 2004. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.4.5690

[40] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, ser. ICCV '99. Washington, DC, USA: IEEE Computer Society, 1999, pp. 1150–. [Online]. Available: http://dl.acm.org/citation.cfm?id=850924.851523

[41] ——, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004. [Online]. Available: http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94

[42] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," in *In ECCV*, 2006, pp. 404–417.

[43] S. Bileschi and L. Wolf, "Image representations beyond histograms of gradients: The role of gestalt descriptors," in *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, june 2007, pp. 1 –8.

[44] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, ser. CVPR '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 886–893. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2005.177

[45] T. Homola, V. Dohnal, and P. Zezula, "Proximity-based order-respecting intersection for searching in image databases," *Adaptive Multimedia Retrieval. Context, Exploration, and Fusion*, pp. 174–188, 2011.

[46] G. Amato and F. Falchi, "knn based image classification relying on local feature similarity," in *Proceedings of the Third International Conference on SImilarity Search and APplications*, ser. SISAP '10. New York, NY, USA: ACM, 2010, pp. 101–108. [Online]. Available: http://doi.acm.org/10.1145/1862344.1862360

[47] M. R. Teague, "Image analysis via the general theory of moments," *Journal of the Optical Society of America (1917-1983)*, vol. 70, pp. 920–930, Aug. 1980. [Online]. Available: http://www.opticsinfobase.org/viewmedia.cfm?id=57703&seq=0

[48] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. SMC-3, no. 6, pp. 610 –621, nov. 1973.

[49] M. Boland, M. Markey, R. Murphy *et al.*, "Automated recognition of patterns characteristic of subcellular structures in fluorescence microscopy images," *Cytometry*, vol. 33, no. 3, pp. 366–375, 1998.

[50] L. Lepistö, L. Kunttu, J. Autio, and A. Visa, "Rock image classification using non-homogenous textures and spectral imaging," in *Spectral Imaging", WSCG SHORT PAPERS proceedings, WSCG'2003, Plzen, Czech Republic*, 2003.

[51] H. Tamura, S. Mori, and T. Yamawaki, "Textural features corresponding to visual perception," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 8, no. 6, pp. 460 –473, june 1978.

[52] T. Ojala, M. Pietikainen, and D. Harwood, "Performance evaluation of texture measures with classification based on kullback discrimination of distributions," in *Pattern Recognition, 1994. Vol. 1 - Conference A: Computer Vision Image Processing., Proceedings of the 12th IAPR International Conference on*, vol. 1, oct 1994, pp. 582 –585 vol.1.

[53] L. Wang and D.-C. He, "Texture classification using texture spectrum," *Pattern Recogn.*, vol. 23, no. 8, pp. 905–910, Aug. 1990. [Online]. Available: http://dx.doi.org/10.1016/0031-3203(90)90135-8

[54] C.-C. Hung, M. Pham, S. Arasteh, B.-C. Kuo, and T. Coleman, "Image texture classification using texture spectrum and local binary pattern," in *Geoscience and Remote Sensing Symposium, 2006. IGARSS 2006. IEEE International Conference on*, 31 2006-aug. 4 2006, pp. 2750 –2753.

[55] M. V. Boland and R. F. Murphy, "A neural network classifier capable of recognizing the patterns of all major subcellular structures in fluorescence microscope images of hela

cells." *Bioinformatics*, vol. 17, no. 12, pp. 1213–1223, 2001. [Online]. Available: http://dblp.uni-trier.de/db/journals/bioinformatics/bioinformatics17.html#BolandM01

[56] R. F. Murphy, M. Velliste, and G. Porreca, "Robust numerical features for description and classification of subcellular location patterns in fluorescence microscope images." *VLSI Signal Processing*, vol. 35, no. 3, pp. 311–321, 2003. [Online]. Available: http://dblp.uni-trier.de/db/journals/vlsisp/vlsisp35.html#MurphyVP03

[57] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, L. M. L. Cam and J. Neyman, Eds., vol. 1. University of California Press, 1967, pp. 281–297.

[58] V. N. Vapnik, *Statistical learning theory*, 1st ed. Wiley, Sep. 1998. [Online]. Available: http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0471030031

[59] H. Zhang, A. Berg, M. Maire, and J. Malik, "Svm-knn: Discriminative nearest neighbor classification for visual category recognition," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 2, 2006, pp. 2126 – 2136.

[60] C.-F. Tsai, K. McGarry, and J. Tait, "Image classification using hybrid neural networks," in *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, ser. SIGIR '03. New York, NY, USA: ACM, 2003, pp. 431–432. [Online]. Available: http://doi.acm.org/10.1145/860435.860536

[61] B. Everitt, *The Cambridge dictionary of statistics*. Cambridge [u.a.]: Cambridge Univ. Press, 1998. [Online]. Available: http://gso.gbv.de/DB=2.1/CMD?ACT=SRCHA&SRT=YOP&IKT=1016&TRM=ppn+244907668&sourceid=fbw_bibsonomy

[62] T. Kohonen, M. R. Schroeder, and T. S. Huang, Eds., *Self-Organizing Maps*, 3rd ed. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2001.

[63] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, Mar. 1986. [Online]. Available: http://dx.doi.org/10.1023/A:1022643204877

[64] A. Bosch, A. Zisserman, and X. Muñoz, "Image classification using random forests and ferns," in *ICCV*. IEEE, 2007, pp. 1–8. [Online]. Available: http://dblp.uni-trier.de/db/conf/iccv/iccv2007.html#BoschZM07

[65] W. Smart and M. Zhang, "Classification strategies for image classification in genetic programming," in *Proceeding of Image and Vision Computing Conference*, D. Bailey, Ed., Palmerston North, New Zealand, 2003, pp. 402–407.

[66] D. Agnelli, A. Bollini, and L. Lombardi, "Image classification: an evolutionary approach," *Pattern Recogn. Lett.*, vol. 23, no. 1-3, pp. 303–309, Jan. 2002. [Online]. Available: http://dx.doi.org/10.1016/S0167-8655(01)00128-3

[67] O. Boiman, E. Shechtman, and M. Irani, "In defense of nearest-neighbor based image classification," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, june 2008, pp. 1 –8.

[68] R. E. Bellman, *Adaptive control processes - A guided tour*. Princeton, New Jersey, U.S.A.: Princeton University Press, 1961.

[69] J. Yang, Y.-G. Jiang, A. G. Hauptmann, and C.-W. Ngo, "Evaluating bag-of-visual-words representations in scene classification," in *Proceedings of the international workshop on Workshop on multimedia information retrieval*, ser. MIR '07. New York, NY, USA: ACM, 2007, pp. 197–206. [Online]. Available: http://doi.acm.org/10.1145/1290082.1290111

[70] M.-E. Nilsback and A. Zisserman, "A visual vocabulary for flower classification," in *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*, ser. CVPR '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 1447–1454. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2006.42

[71] A. Bosch, A. Zisserman, and X. Muñoz, "Scene classification via plsa," in *Computer Vision – ECCV 2006*, ser. Lecture Notes in Computer Science, A. Leonardis, H. Bischof, and A. Pinz, Eds. Springer Berlin Heidelberg, 2006, vol. 3954, pp. 517–530. [Online]. Available: http://dx.doi.org/10.1007/11744085_40

[72] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *In Workshop on Statistical Learning in Computer Vision, ECCV*, 2004, pp. 1–22.

[73] R. Jensen and Q. Shen, "New approaches to fuzzy-rough feature selection," *Fuzzy Systems, IEEE Transactions on*, vol. 17, no. 4, pp. 824 –838, aug. 2009.

[74] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, pp. 338–353, 1965. [Online]. Available: http://www-bisc.cs.berkeley.edu/Zadeh-1965.pdf

[75] Z. Pawlak, *Rough Sets: Theoretical Aspects of Reasoning about Data*. Norwell, MA, USA: Kluwer Academic Publishers, 1992.

[76] C. Shang, D. Barnes, and Q. Shen, "Facilitating efficient mars terrain image classification with fuzzy-rough feature selection," *Int. J. Hybrid Intell. Syst.*, vol. 8, no. 1, pp. 3–13, Jan. 2011. [Online]. Available: http://dl.acm.org/citation.cfm?id=1971737.1971739

[77] R. E. Schapire, "The boosting approach to machine learning: An overview," 2002.

[78] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55,

no. 1, pp. 119 – 139, 1997. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S002200009791504X

[79] Z. Tu, "Probabilistic boosting-tree: learning discriminative models for classification, recognition, and clustering," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 2, oct. 2005, pp. 1589 –1596 Vol. 2.

[80] J. Fan, Y. Gao, and H. Luo, "Hierarchical classification for automatic image annotation," in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, ser. SIGIR '07. New York, NY, USA: ACM, 2007, pp. 111–118. [Online]. Available: http://doi.acm.org/10.1145/1277741.1277763

[81] X. Liu, L. Zhang, M. Li, H. Zhang, and D. Wang, "Boosting image classification with lda-based feature combination for digital photograph management," *Pattern Recogn.*, vol. 38, no. 6, pp. 887–901, Jun. 2005. [Online]. Available: http://dx.doi.org/10.1016/j.patcog.2004.11.008

[82] O. Boiman and M. Irani, "Similarity by composition," in *In NIPS*, 2006.

[83] E. Shechtman and M. Irani, "Matching local self-similarities across images and videos," in *IEEE Conference on Computer Vision and Pattern Recognition 2007 (CVPR'07)*, June 2007.

[84] C. Gu, J. Lim, P. Arbeláez, and J. Malik, "Recognition using regions," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. Ieee, 2009, pp. 1030–1037.

[85] Y. Schnitman, Y. Caspi, D. Cohen-Or, and D. Lischinski, "Inducing semantic segmentation from an example," in *Proceedings of the 7th Asian conference on Computer Vision - Volume Part II*, ser. ACCV'06. Berlin, Heidelberg: Springer-Verlag, 2006, pp. 373–384. [Online]. Available: http://dx.doi.org/10.1007/11612704_38

[86] J. J. Lim, P. Arbelaez, C. Gu, and J. Malik, "Context by region ancestry." in *ICCV*. IEEE, 2009, pp. 1978–1985. [Online]. Available: http://dblp.uni-trier.de/db/conf/iccv/iccv2009.html#LimAGM09

[87] X. Ren, C. Fowlkes, and J. Malik, "Figure/ground assignment in natural images," *Computer Vision–ECCV 2006*, pp. 614–627, 2006.

[88] B. Hariharan, P. Arbelaez, L. Bourdev, S. Maji, and J. Malik, "Semantic contours from inverse detectors," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, nov. 2011, pp. 991 –998.

[89] M. J. Huiskes, B. Thomee, and M. S. Lew, "New trends and ideas in visual concept detection: the mir flickr retrieval evaluation initiative," in *Proceedings of the international conference on Multimedia information retrieval*, ser. MIR

'10. New York, NY, USA: ACM, 2010, pp. 527–536. [Online]. Available: http://doi.acm.org/10.1145/1743384.1743475

[90] S. Nowak, K. Nagel, and J. Liebetrau, "The clef 2011 photo annotation and concept-based retrieval tasks," in *CLEF (Notebook Papers/Labs/Workshop)*, V. Petras, P. Forner, and P. D. Clough, Eds., 2011.

[91] Y. Liu, D. Zhang, G. Lu, and W.-Y. Ma, "A survey of content-based image retrieval with high-level semantics," *Pattern Recognition*, vol. 40, no. 1, pp. 262 – 282, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320306002184

[92] T. Homola, V. Dohnal, and P. Zezula, "Searching for sub-images using sequence alignment." in *ISM*. IEEE Computer Society, 2011, pp. 61–68. [Online]. Available: http://dblp.uni-trier.de/db/conf/ism/ism2011.html#HomolaDZ11

[93] ——, "Sub-image searching through intersection of local descriptors," in *Proceedings of the Third International Conference on SImilarity Search and Applications*. ACM, 2010, pp. 127–128.

[94] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, june 2009, pp. 248 –255.

[95] C. Fellbaum, Ed., *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*, illustrated edition ed. The MIT Press, May 1998. [Online]. Available: http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/026206197X

[96] P. Soille, *Morphological Image Analysis: Principles and Applications*, 2nd ed. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2003.

[97] D. Y. Orlova, L. Stixová, S. Kozubek, H. J. Gierman, G. Šustáčková, A. V. Chernyshev, R. N. Medvedev, S. Legartová, R. Versteeg, P. Matula, R. Stoklasa, and E. Bártová, "Arrangement of nuclear structures is not transmitted through mitosis but is identical in sister cells," *Journal of Cellular Biochemistry*, pp. n/a–n/a, 2012. [Online]. Available: http://dx.doi.org/10.1002/jcb.24208

[98] (2012) Biomedical images database. [Online]. Available: http://nerone.diiie.unisa.it/zope/mivia/databases/db_database/biomedical/

[99] M. Everingham, L. Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *Int. J. Comput. Vision*, vol. 88, no. 2, pp. 303–338, Jun. 2010. [Online]. Available: http://dx.doi.org/10.1007/s11263-009-0275-4

Appendix A

Summary of the Study

A.1 Passed Courses

- PA170 Digital Geometry (Autumn 2010)
- PA173 Mathematical Morphology (Autumn 2010)
- MV011 Statistics I (Spring 2011)
- MA012 Statistics II (Autumn 2011)
- PA171 Digital Image Filtering (Spring 2012)

A.2 Participations

- August - September 2011: *Advanced Methods on Biomedical Image Analysis* (AMBIA) Summer School, Brno, Czech Republic (*14 days*)
- October 2011: *Image Acquisition and Processing in Biomedical Microscopy* Course, Prague, Czech Republic (*5 days*)

A.3 Publications

- R. Stoklasa and Pe. Matula, "Road detection using similarity search," in *2nd International Conference on Robotics in Education*, R. Stelzer and K. Jafarmadar, Eds., Vienna, 2011, pp. 95–102, ISBN 978-3-200-02273-7.
  — my contribution was approximately *85%*. I have developed the main idea, implemented the algorithm and wrote the majority of the paper.
- R. Stoklasa, T. Majtner, D. Svoboda, and M. Batko, "HEp-2 Cells Classifier," (software). 2012
  — my contribution was approximately *45%*. I was responsible for design and development of the classifier, data pre-processing and other tasks. Details about this software can be found on http://cbia.fi.muni.cz/projects/hep-2-cells-classifier.html. Working software can be obtained after contacting of authors.

- D. Y. Orlova, L. Stixová, S. Kozubek, H. J. Gierman, G. Šustáčková, A. V. Chernyshev, R. N. Medvedev, S. Legartová, R. Versteeg, P. Matula, R. Stoklasa, and E. Bártová, "Arrangement of nuclear structures is not transmitted through mitosis but is identical in sister cells," *Journal of Cellular Biochemistry*, pp. n/a–n/a, 2012. [Online]. Available: http://dx.doi.org/10.1002/jcb.24208
  — my contribution was approximately *3%*. I worked on evaluation of similarity between sister cells in order to support the claim that some particular cells are sisters.

A.4 Given Presentations

- Oral presentation *Road Detection Using Similarity Search* presented at *2nd International Conference on Robotics in Education*, Vienna, September 2011
- Presentation at Seminar of Searching and Dialog Laboratory, Spring 2011
- Several presentations at Center for Biomedical Image Analysis, one presentation each semester

A.5 Teaching

In addition to my research activities, I have assisted in teaching of the course PB069 *Desktop Application Development in C#/.NET* in Spring 2011 and Spring 2012 (2 seminar groups, 4 hours per week).

A.6 Supervising

- I supervised Ketan Bacchuwar during his 3-month stay at the CBIA group.
- I'm supervising one bachelor thesis which will be defended in the February 2013.

Appendix B

Publication about Road Detection

The following pages contain the publication about road detection algorithm. My contribution to this publication is approximately *85%*. I have developed the main idea, implemented the algorithm and wrote the majority of the paper.

Road Detection Using Similarity Search

Roman Stoklasa
Faculty of Informatics
Masaryk University
Brno, Czech Republic
Email: xstokla2@fi.muni.cz

Petr Matula
Faculty of Informatics
Masaryk University
Brno, Czech Republic
Email: pem@fi.muni.cz

***Abstract*—This paper concerns vision-based navigation of autonomous robots. We propose a new approach for road detection based on similarity database searches. Images from the camera are divided into regular samples and for each sample the most visually similar images are retrieved from the database. The similarity between the samples and the image database is measured in a metric space using three descriptors: edge histogram, color structure and color layout, resulting in a classification of each sample into two classes: road and non-road with a confidence measure. The performance of our approach has been evaluated with respect to a manually defined ground-truth. The approach has been successfully applied to four videos consisting of more than 1180 frames. It turned out that our approach offers very precise classification results.**

***Index Terms*—road detection, similarity search, navigation, image classification, autonomous robot, Robotour**

I. Introduction

Robotour—robotika.cz outdoor delivery challenge[1] is a Czech competition of autonomous robots navigating on park roads, the aim of which is to promote development of robots capable of transporting payloads completely autonomously in a natural environment. Development of the approach presented here have been motivated by this competition.

For a successful navigation some kind of environment perception is necessary. The perception can either be based on *non-visual* techniques, such as odometry, infrared sensors, usage of a compass and GPS signal, or based on *visual* information obtained by a camera (or several different cameras). The non-visual techniques are in general more sensitive to outdoor environment and the information content is not so rich as in the case of visual navigation. Efficient analysis of visual information is very challenging.

Two notable approaches to navigation using visual information have been used by winner teams in the previous years of Robotour competition. The basic principle of the first approach described in [1], [2] is to find a set of interesting points on the camera image [3], which represents some significant points in 3D space. It is essential to have a special "map" that contains a huge number of these points with their position in the environment. This map must be created before the navigation process itself and it is typically built during a series of supervised movements of a robot through all possible roads. All detected points are stored in a database with their estimated position. When the robot navigates autonomously in such mapped environment, interesting points are extracted from the image and compared to the points in the "map". The position and orientation of the robot is determined according to the matching points. The main disadvantage of this approach is the need of creating an ad hoc map of the whole environment where the navigation process would take place. Because building of ad hoc maps is impractical for large environments, this kind of approaches is not allowed from the year 2010 on.

The second navigation approach used by Eduro Team [4]—winner of Robotour 2010—combines a road detection with an OpenStreetMap map. For the road detection they used an algorithm based on the principle described in [5]. The idea is to track similar visual pattern that appears in the bottom of the image. It is assumed that there is a road in the bottom part of the image and everything that looks similar is also the road. This simplification brings a big disadvantage because when a robot gets to an difficult situation (for example when it arrives to an edge of the road) this method can easily be confused and start to follow a non-road visual pattern, or, vice versa, it can cause problems on the boundaries between two different road surfaces.

In this paper we address a subtopic of the whole navigation problem of autonomous robots in the natural environment based on similarity searches (Section II), which does not build any ad hoc map before the navigation. In particular we present a novel approach for road detection from the input images taken by robot's camera (Section III), which can detect roads even with different surfaces. We show (Section IV) that the proposed approach can reliably detect roads under various light and environment conditions and that it can also detect unpredictable situations not present in the training data, which could otherwise negatively influence the navigation process.

II. Similarity Search

Content-based image retrieval is a process of finding images in some image collection or database that are visually similar to the specified query image. We need to represent images using objects in some metric space in order to be able to define some (dis)similarity measure between them [6]. It is very common to use a vector space with an appropriate metric function as a metric space. In such a case, we have to represent images as vectors in this vector space.

[1] http://robotika.cz/competitions/robotour/en

Visual descriptors are used to describe some image characteristics in a form of vectors. There are many different image characteristics which can be described, for example, color properties, textures or shapes. In our case, we are using global descriptors from the MPEG-7 standard [7], namely: edge histogram, color layout and color structure. Edge histogram descriptor (EHD) is a sort of texture descriptor describing the spatial distribution of edges in the image. It produces an 80-dimensional vector and is partially invariant to image resolution. Color layout descriptor (CLD) describes spatial distribution of colors in the image and is resolution-invariant. CLD works in YCbCr color space and produces a 12-dimensional vector. Color structure descriptor (CSD) represents an image by both the color distribution of the image and the local spatial structure of the color. This color descriptor works in HMMD color space. CSD produces 64-dimensional vectors.

In general, every descriptor uses its own vector space with a different metric function due to different dimensionalities. In order to compare images according to multiple criteria, it is possible to combine multiple descriptors together using an aggregation function (e.g., a weighted sum or a product). We used weighted sum as the aggregation function for combining the dissimilarity values for each single descriptor.

There are two basic types of similarity queries: *range query* and *k-nearest neighbor (k-NN) query*. Range query $R(q, r)$ returns all images whose distance from the query image q is smaller than range r. k-nearest neighbor query $k\text{-NN}(q, k)$ returns up to k nearest images to the query q. We use k-NN query type in our approach.

In the training phase, we store different samples of categories of interest into a database with a label (attribute) specifying their class. We use two classes: *road* and *non-road*.

Similarity search engine is implemented using MESSIF similarity search engine framework [8].

III. Road Detection

Input of our road detection algorithm are images from a robot's camera. Output of the algorithm is a *classification map*.

Classification map is an image with the same dimension as the input image, which contains for each pixel a likelihood that the pixel belongs to a particular class. In our case, this map contains 2 values for each pixel: (1) the likelihood that the pixel belongs to the *road* class and (2) the likelihood that the pixel belongs to the *non-road* class. In Fig. 1, the classification map is visualized with blue (road) and red (non-road) colors and the likelihood is represented with their brightness. The darker the color the lower the likelihood.

Our road detection algorithm can be divided into the following steps (see Fig. 1).

1) Sampling of the input image—input image is divided into suitable rectangular regions (called samples), which are processed individually
2) For each sample from the input image:
   a) Retrieve the most similar samples of known surfaces from the database using k-NN query

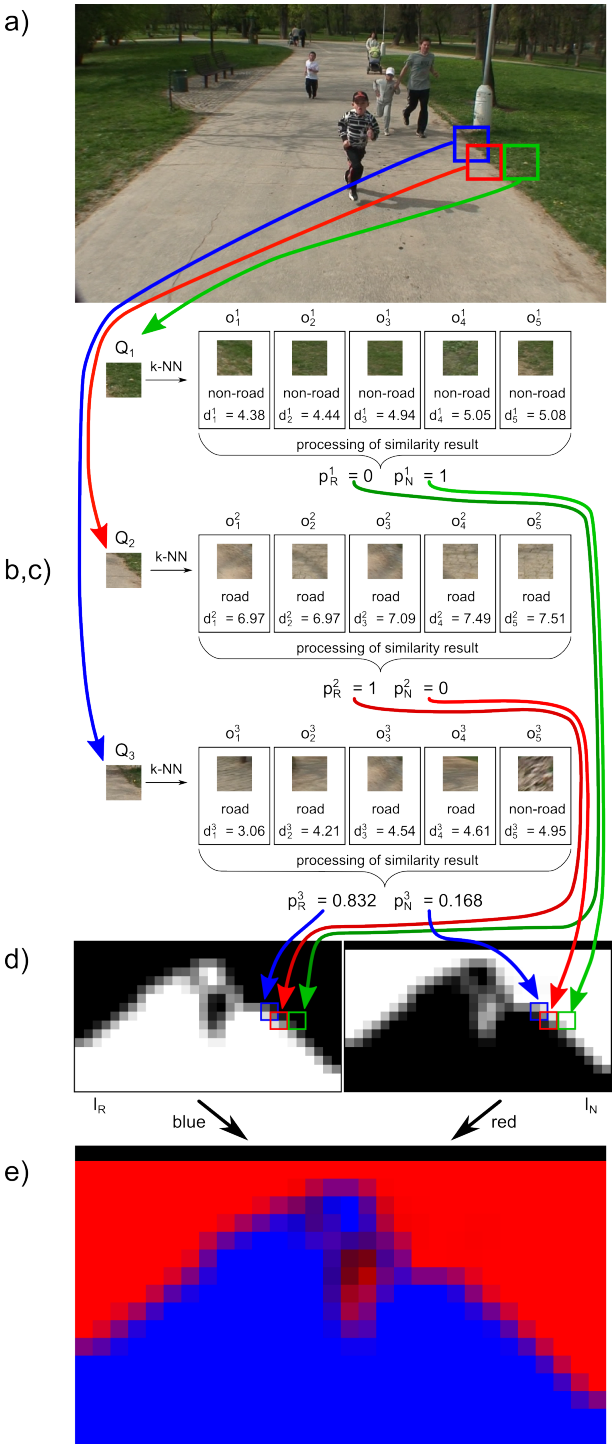


Fig. 1. Illustration of the road detection algorithm. a) Samples Q_1, Q_2 and Q_3 are extracted from the input image. b) Most similar images from database are retrieved for each sample Q_i (for $i \in 1, 2, 3$) using k-NN query ($k = 5$). c) Results from similarity database are combined together and likelihoods p^i_R and p^i_N are computed for each region Q_i. d) Values p^i_R and p^i_N are stored separately in the classification map. e) Classification map, where the likelihood that the pixel belongs to road and non-road classes is visualized with blue and red colors, respectively.

b) Process the retrieved information from the similarity database and estimate the likelihood that the sample from input image contains road or non-road

3) Combine classification result of each sample from input image and create the whole classification map

A. Sampling of input image

We divided the input images into regular rectangular regions with some overlaps. For images of size 720×576 px and 960×540 px, we used samples of size 64×64 px with an overlap of 32 px. The procedure is illustrated in Fig. 2.

With this sampling strategy it can happen that a sample contains both road and non-road areas. However, this is not a problem because the similarity search engine can return the most similar samples from the database and the similarities are combined together. In order to reduce uncertainties in the classification map we use the overlaps.

We use segmentation into regular tiles of same sizes due to strightforward implementation. The size of samples was determined empirically for our testing data set as the compromise between the resolution of classification and the computational complexity. Every single sample should contain enough characteristic visual clues with discrimination power for classification of the particular type of surface. Too small samples would not contain enough visual clues and the total amount of samples would be very high; too large samples would tend to contain more than one type of surface, which would decrease the precision of classification.

B. Similarity query and processing of similarity result

For each sample from an input image we search for k most similar samples in the database using k-NN query. Let Q_i denote i-th sample from the input image. Response of the k-NN(Q_i, k) query contains (up to) k objects $\{o_1^i, o_2^i, \ldots, o_k^i\}$. Each response object o_j^i can be written in a form of triple $o_j^i = (img_j^i, d_j^i, c_j^i)$, where img_j^i denotes the image from the database, d_j^i represents distance from the query image Q_i and c_j^i is the class to which the sample img_j^i belongs. Based on this response we determine the likelihood p_R^i that the sample Q_i contains road and the likelihood p_N^i that it contains non-road.

In order to determine likelihoods p_R^i and p_N^i we combine results of k-NN query based on the information from the search engine. Both probabilities are computed as a weighted combination of $\{c_1^i, \ldots, c_k^i\}$.

1) Weights: Let $\{w_1^i, \ldots, w_k^i\}$ denote weights for classes $\{c_1^i, \ldots, c_k^i\}$ that belongs to objects $\{o_1^i, \ldots, o_k^i\}$. We require that following properties hold:

- If an object o_m^i is λ-times closer to Q_i than an object o_n^i, then classification information c_m^i should have λ-times higher weight than c_n^i:

$$d_m^i = \frac{1}{\lambda} d_n^i \Longrightarrow w_m^i = \lambda w_n^i$$

  Note that this rule is consistent also in a situation, when the distance d_m^i is equal to 0 and distance d_n^i is non-zero. In such case c_m^i will be considered as the only one relevant class information, because weight w_m^i will be infinite.
- Sum of all weights should be equal to 1 (except the special case that some of the distances d_j^i would be 0):

$$\sum_{j=1}^{k} w_j^i = 1 \tag{1}$$

Assume that we have a set $\{(d_1^i, c_1^i), \ldots, (d_k^i, c_k^i)\}$ as the input for the aggregation function. Assume that this set is ordered ascending according to the distance so that d_1^i is the lowest distance and d_k^i is the biggest one. We define a normalizing term for the weights as:

$$N_w^i = \sum_{j=1}^{k} \frac{d_k^i}{\max(d_j^i, \epsilon)} \tag{2}$$

Because the distance d_j^i can be in general equal to 0, we need the term $\max(d_j^i, \epsilon)$ in the denominator to avoid division by zero. ϵ is some arbitrary small positive value (for example 10^{-6}). Then we can define weight w_j^i as:

$$w_j^i = \frac{1}{N_w^i} \cdot \frac{d_k^i}{\max(d_j^i, \epsilon)} \tag{3}$$

It holds, that $\sum_{j=1}^{k} w_j^i = 1$

2) Confidence factor: As we have mentioned above, we want to estimate some factor of confidence, that the similarity results are relevant. We define a function $\alpha(d)$:

$$\alpha(d) = \begin{cases} 0 & \text{for } d > 2T_d; \\ 1 & \text{for } d < T_d; \\ 1 - \frac{d - T_d}{T_d} & \text{for } T_d \leq d \leq 2T_d; \end{cases} \tag{4}$$

which define the confidence that the object class in the database with distance d from query q is relevant also for query image q itself. T_d is a threshold of "absolute confidence". If the distance between an object o and a query q is less then T_d, confidence value is equal to 1. If the distance is in the range $\langle T_d, 2T_d \rangle$ confidence value decreases linearly, and if the distance is greater than $2T_d$, the confidence is equal to 0.

3) Final likelihoods: If we define that $c_j^i = 1$ when the image img_j^i represents road and $c_j^i = 0$ when the image img_j^i represents non-road then we can compute final likelihoods p_R^i and p_N^i using:

$$p_R^i = \sum_{j \in \{x | c_x^i = 1\}} \alpha(d_j^i) \cdot w_j^i \tag{5}$$

$$p_N^i = \sum_{j \in \{x | c_x^i = 0\}} \alpha(d_j^i) \cdot w_j^i \tag{6}$$

With these definitions, numbers p_R^i and p_N^i can have a value only from interval $\langle 0, 1 \rangle$ and it must hold that $p_R^i + p_N^i \leq 1$. The inequality can happen if sample Q_i is not similar enough to any of the samples in the database. These definitions allows us to work with a confidence in similarity search results and are a key part of our approach. Note that these definitions can easily be extended to any number of classes.

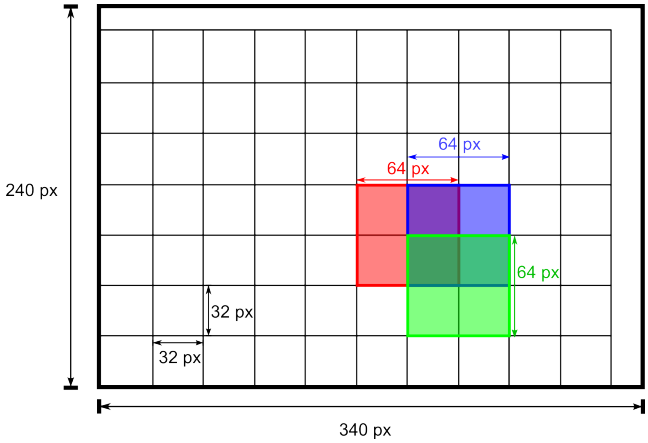


Fig. 2. Segmentation of input image into tiles for similarity search. We used samples of size 64×64 px with overlaps of 32 px.

C. Creation of classification map

From the previous step, we have a set of triples $\{(r^1, p_R^1, p_N^1), \ldots, (r^n, p_R^n, p_N^n)\}$, where r^i is i-th region (corresponding to i-th query image Q_i) and p_R^i and p_N^i are the likelihoods defined above. Because regions from $\{r^1, \ldots, r^n\}$ may overlap, we define a final classification of pixel p in the classification map as the average of classifications of all regions that contain the pixel p.

Fig. 3 shows an example of the final classification map computed by our algorithm. The value of p_R is encoded into the blue color channel, the value of p_N is encoded into the red channel. Dark areas in the image means that the algorithm was unable to reliably determine the classification of that areas, because that areas are not visually similar to any of the known samples in the database (in those areas the sum of $p_R + p_N$ is lower than 1).

IV. Evaluation and Results

A. Test data-sets

We tested our method on videos from a real outside environment recorded in a park[2] in the same way as would be recorded when the camera would be carried by an autonomous robot. The test videos were recorded on 2 different days with different light conditions.

We present results on 4 different video sequences (called "walks"). The first and the second videos (called *walk-01* and *walk-02*) were recorded using Canon XM2 camcorder on an autumn day with an overcast weather. These videos were recorded with resolution of 720×576 px. The third and the fourth videos (called *walk-03* and *walk-04*) were recorded with Sony HDR HC-3 camera on a sunny spring day. The videos were recorded in HD resolution (1920×1080 px), but we worked with downsampled images with resolution 960×540 px.

All videos together had a total length of more than 28 minutes. For the evaluation of classification precision we used 364 frames, which were picked evenly in intervals ranging from 0.8 to 8 seconds for different *walks*.

We defined ground-truth manually for each frame in the testing set. Ground-truth for each frame was created as a mask of road area in the frame. We draw the mask manually using a bitmap editor.

[2]park Lužánky, Brno, Czech Republic

B. Knowledge base

Content of our knowledge base was generated semi-automatically. We picked some frames from our testing set, for which we had defined ground-truth. From these frames we extracted several samples of road and non-road regions in the following way. A computer generated several random positions of the sampling window. Each sample whose domain overlapped with road or non-road area in the ground truth for more than 93% was included into the knowledge base. The threshold of 93% was determined empirically.

We picked 53 frames from videos *walk-01* and *walk-02* and then we generated 50 samples of size 64×64 px from each frame. We have manually discarded samples that contained some image abnormality, e.g., over-exposed regions. After this processing we got 2635 samples. The size of our testing knowledge base turned out to be sufficient in our case. We did not rigorously test the minimum size of the knowledge base and did not study the relation between its size and the environment variability in which the navigation should occur.

From videos *walk-03* and *walk-04* we picked 15 and 11 frames respectively and from each frame we generated 20 samples. Using this process we obtained additional 520 samples.

Some examples of such samples stored in our knowledge base are shown in Fig. 4.

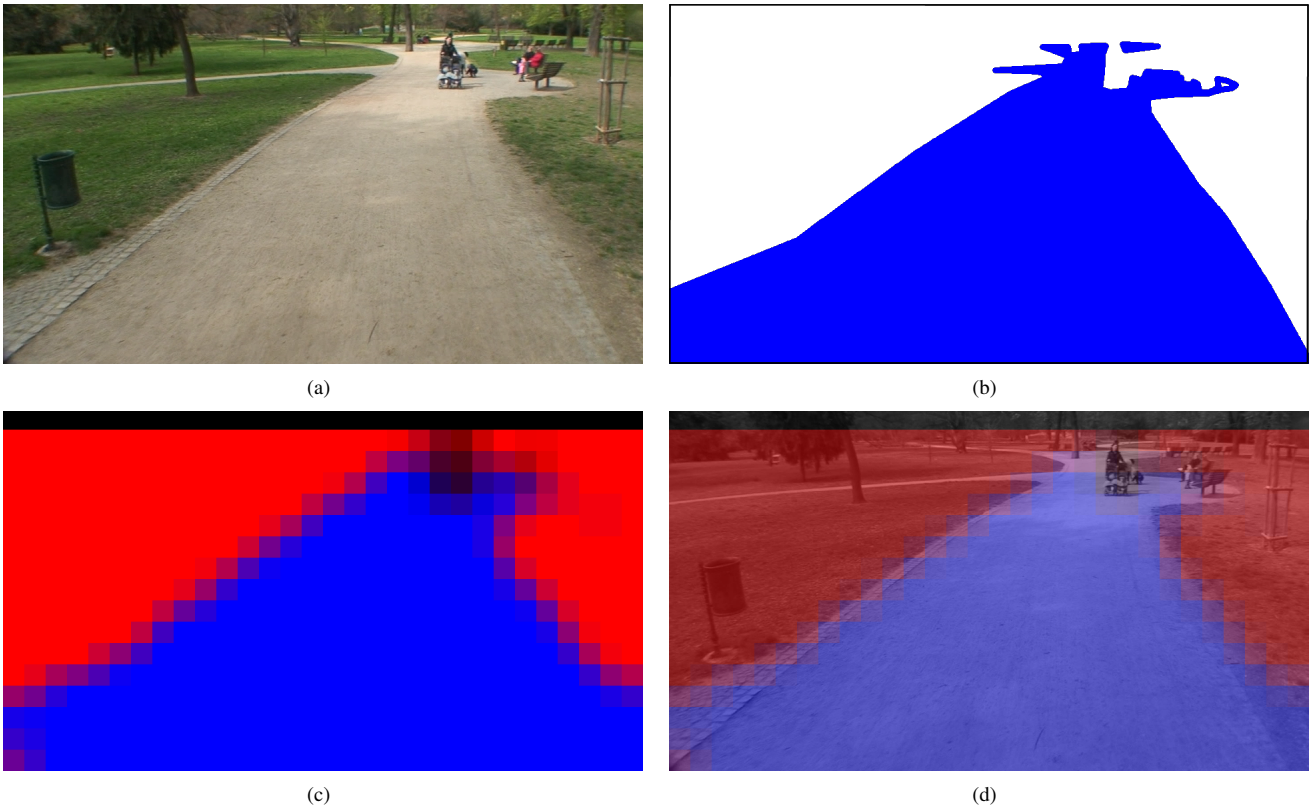
C. Precision Evaluation

We defined several error metrics in order to evaluate precision of our algorithm in a quantitative way. The amount of an error depends on the two factors: size of the area on which we obtained other than expected result; and also on the difference between expected and actual result.

We define two measures: "absolute amount of intensity under the mask" (denoted by S_A) and a "relative amount of intensity under the mask" (denoted by S_R). Both measures are evaluated with respect to the ground-truth image GT (which serve as an mask) and a gray-scale image I. Let GT image be a binary image that contains only values 0 or 1. Let I be a gray-scale image, which contains values from interval $\langle 0, 1 \rangle$. Let both images have the same dimensions over a domain Ω. Expressions $GT(p)$ and $I(p)$ denote intensity value of pixel p within the image GT and I respectively. Let the numbers w and h be the width and the height of the images. Then we can define S_A and S_R using:

$$S_A = \frac{\sum_{p \in \Omega} \min(GT(p), I(p))}{w \cdot h}$$

$$S_R = \frac{\sum_{p \in \Omega} \min(GT(p), I(p))}{\sum_{p \in \Omega} GT(p)}$$



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

Fig. 3. (a) Input frame from the camera. (b) Manually defined ground-truth for the frame (blue area represents road). (c) Computed classification map. Notice the dark area in the upper part of the classification map—this area contains visually unknown pattern and thus the confidence of the classification is low. Topmost black bar is unclassified margin of the image. (d) Classification map overexposed over input frame.

In these equations a sum of minimal pixel values at corresponding positions in images I and GT are calculated and they are either normalized with respect to the surface of the whole image (in case of S_A) or with respect to the surface of the mask (in case of S_R). Both S_A and S_R have values from the interval $\langle 0, 1 \rangle$.

A value of S_A expresses the ratio between the sum of intensities under the mask and the maximally possible sum of intensities in the whole image; a value of S_R expresses the ratio between the sum of intensities under the mask and the maximally possible sum of intensities under the mask.

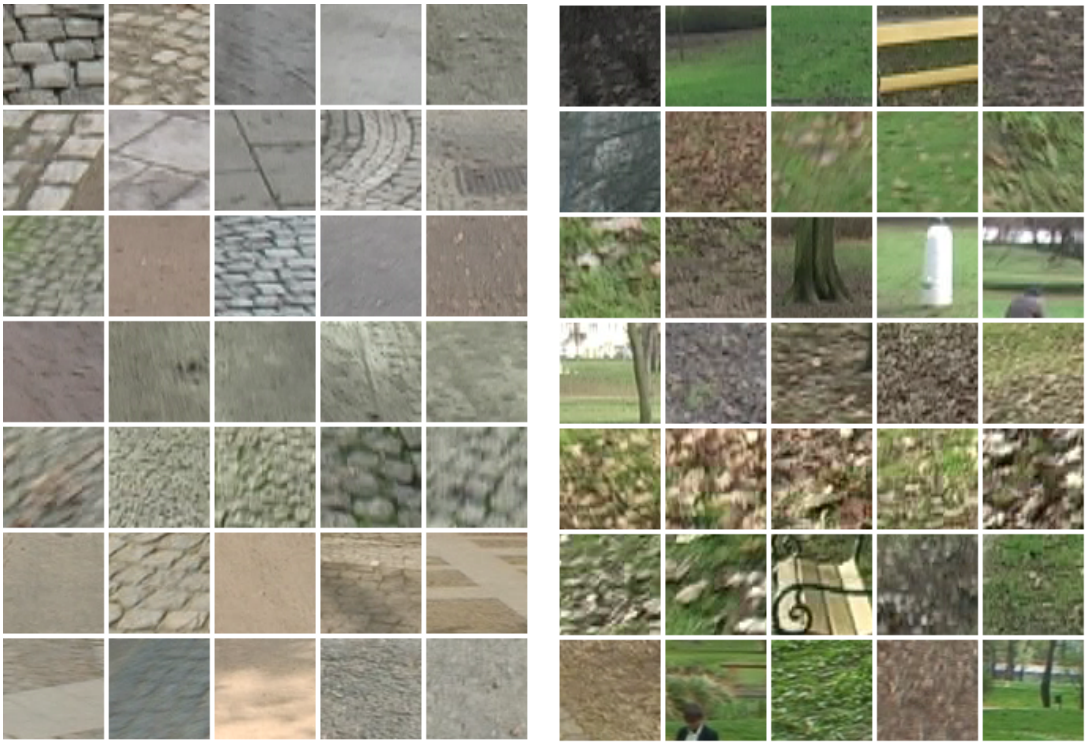
Let I denote the whole classification map encoded as image, I_B denote the blue channel of image I (which contains values of p_R), I_R denote the red channel of image I (which contains values of p_N), GT denote manually defined ground-truth, which contains value 1 for pixels, which represent road and 0 for those, which represent non-road. Let $\overline{X}$ denote complement (i.e., negative) of the image X.

We define several error metrics:

- Error of type FP (*False Positive*) – quantifies the proportion of pixels classified as road within non-road regions
  Defined as: $FP(I) = S_A(I_R, \overline{GT})$
- Error of type FN (*False Negative*) – quantifies the proportion of pixels classified as non-road within road regions.
  Defined as: $FN(I) = S_A(I_N, GT)$
- Error of type NP (*Non-Positive*) – quantifies the proportion of pixels not classified as road within road regions.
  Defined as: $NP(I) = S_A(\overline{I_R}, GT)$
- Error of type NN (*Non-Negative*) – quantifies the proportion of pixels not classified as non-road within non-road regions.
  Defined as: $NN(I) = S_A(\overline{I_N}, \overline{GT})$
- Precision of type PA (*Positive Accuracy*) – quantifies the proportion of pixels that were correctly classified as road regions.
  Defined as: $PA(I) = S_R(I_R, GT)$
- Precision of type NA (*Negative Accuracy*) – quantifies the proportion of pixels that were correctly classified as non-road regions.
  Defined as: $NA(I) = S_R(I_N, \overline{GT})$

D. Error induced on the road boundary

It is obvious that there must always be some inaccuracy caused by the used sampling strategy, where we divide the input image into the regular rectangular regions with the smallest possible resolution of 32×32 px. Because of this discretization we cannot precisely classify pixels near the border between road and non-road regions. Therefore we evaluated all error metrics also in a variant which ignores errors on the boundary between road and non-road regions.



(a) (b)

Fig. 4. Examples of images stored in knowledge base: (a) samples of road class and (b) samples of non-road class.

E. Results

Statistics of the achieved results are summarized in Table I. Values in the table are the average values of a particular error metric for all frames from a particular walk.

As seen in the table, an average error of type FP was approximately 3% (only for video *walk-01* reached almost the value of 10%). This can be interpreted in a way that 3% of the image area was classified incorrectly as a road. When we disregard an error induced on the border between road and non-road, all error metrics FP, FN, NP, NN became smaller by approx. 2.5%. Thus, when we ignore errors on the borders borders, we can say that our classification method failed to correctly classify regions in less than 1% of image surface.

Because we allow "unknown" classification in our approach (Section III-B), we also evaluated, how often this "uncertainty" happens. The amount of the "unknown" classification in the road and non-road areas can be computed as the difference NP−FN and NN−FP respectively. We can see from the Table I that this difference is mostly less then 0.5%.

It is also seen, that our road detection method was able to detect more than 85% of road area in the input images, therefore we think it should be possible to easily navigate robot through the real roads based on the result obtained from our algorithm.

Table II shows the results achieved for *walk-03* and *walk-04* which were classified using "knowledge base" based only on samples from *walk-01* and *walk-02*. As we can see that the

TABLE I

EVALUATION OF ERROR METRICS FOR ALL FOUR WALKS. VARIANT I SHOWS ERROR FOR THE WHOLE IMAGE, VARIANT II SHOWS ERROR WITHOUT THE ERROR ON THE BORDER BETWEEN ROAD AND NON-ROAD. ALL VALUES ARE AVERAGE VALUE OF PARTICULAR ERROR METRIC FOR ALL FRAMES OF THAT PARTICULAR WALK.

Walk	walk-01		walk-02	
Number of frames	76		82	
Variant	I	II	I	II
FP	9.86%	5.33%	2.95%	0.39%
FN	1.86%	0.34%	3.19%	0.36%
NP	1.91%	0.34%	3.20%	0.37%
NN	10.08%	5.48%	2.97%	0.40%
PA	93.90%	95.45%	85.46%	90.69%
NA	76.30%	85.42%	93.41%	99.16%

Walk	walk-03		walk-04	
Number of frames	98		108	
Variant	I	II	I	II
FP	2.57%	0.38%	3.19%	0.71%
FN	3.50%	0.28%	0.71%	0.26%
NP	3.68%	0.29%	2.99%	0.30%
NN	3.07%	0.57%	3.83%	1.06%
PA	92.47%	99.42%	93.98%	99.23%
NA	93.86%	98.75%	91.70%	97.18%

TABLE II

EVALUATION OF ERROR METRICS FOR CLASSIFICATION OF *walk-03* AND *walk-04* USING DATABASE GENERATED FROM *walk-01* AND *walk-02*. ALL VALUES ARE AVERAGE VALUE OF PARTICULAR ERROR METRIC FOR ALL FRAMES OF THAT PARTICULAR WALK. THESE RESULTS SHOW THAT THE DATABASE OF SAMPLES CAN BE "PORTABLE" (I.E. IT IS NOT BOUND TO THE PARTICULAR ENVIRONMENT AND THE PARTICULAR CAMERA).

Walk	walk-03		walk-04	
Number of frames	98		108	
Variant	I	II	I	II
FP	1.99%	0.30%	2.76%	0.54%
FN	4.88%	0.90%	3.45%	0.57%
NP	5.11%	0.92%	3.72%	0.68%
NN	2.61%	0.56%	3.48%	0.96%
PA	89.86%	98.00%	92.48%	98.29%
NA	95.10%	98.85%	92.40%	97.45%

results are still very precise. This shows that our method works well also for images that have not been used for building the database and which were taken by a different camera on a day with different weather conditions.

In Fig. 5, there are shown examples of computed classification maps related to the input images. Classification map is encoded as red-blue image and is superimposed over the corresponding frame from the camera for a better illustration. Fig. 6 shows some examples with obstackles, which were correctly classified as non-road.

F. Final remarks

We did not have to introduce any complex preprocessing steps before road detection because the fully automatic modes adjusting exposure time, color balance, etc., that we have used on the camcorders (Canon XM2 and Sony HDR HC-3) worked sufficiently well. Many low-end cameras would not be able to deal with these tasks and their produced images could be degraded in some way. In such case, additional image preprocessing may be necessary to achieve comparable results.

V. CONCLUSIONS AND FUTURE WORK

We have proposed a new method of road detection for robot navigation in natural environment. We have tested it on real data sets recorded with two different cameras under different conditions. The obtained results indicate that our algorithm could be applicable also for a real robotic implementation. Error in surface classification is less than 1% in average and the average classification error of the whole frame from camera is less then 5%.

The most computationally intensive part of the algorithm is the processing of all samples from input images and searching for visually similar images for each of them. One can easily see, that processing of one sample is independent of the others, so all samples can be processed in parallel.

This method can be easily extended to recognize multiple classes of surfaces.

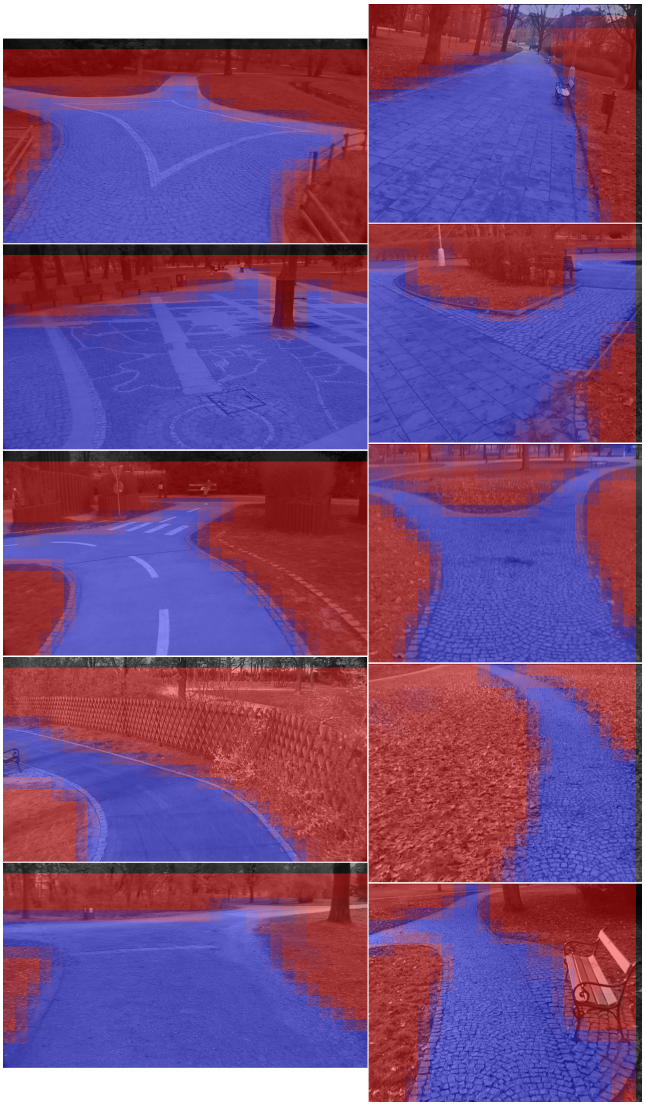


Fig. 5. Examples of classified frames exported from all videos. Frames in the left column are from *walk-03* and *walk-04*, frames in the right column are from *walk-01* and *walk-02*

REFERENCES

[1] T. Krajník, J. Faigl, M. Vonsek, V. Kulich, K. Košnar, and L. Přeučil, "Simple yet stable bearing-only navigation," *J. Field Robot.*, 2010.

[2] T. Krajník and L. Přeučil, "A Simple Visual Navigation System with Convergence Property," in *Proceedings of Workshop 2008*. Praha: Czech Technical University in Prague, 2008, pp. –.

[3] J. Šváb, T. Krajník, J. Faigl, and L. Přeučil, "FPGA-based Speeded Up Robust Features," in *2009 IEEE International Conference on Technologies for Practical Robot Applications*. Boston: IEEE, 2009, pp. 35–41.

[4] J. Iša, T. Roubíček, and J. Roubíček, "Eduro Team," in *Proceedings of the 1st Slovak-Austrian International Conference on Robotics in Education*, Bratislava, 2010, pp. 21–24.

[5] L. M. Lorigo, R. A. Brooks, and W. E. L. Grimson, "Visually-guided obstacle avoidance in unstructured environments," in *IEEE Conference on Intelligent Robots and Systems*, 1997, pp. 373–379.

[6] P. Zezula, G. Amato, V. Dohnal, and M. Batko, *Similarity Search: The Metric Space Approach (Advances in Database Systems)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005.

[7] P. Salembier and T. Sikora, *Introduction to MPEG-7: Multimedia Content*

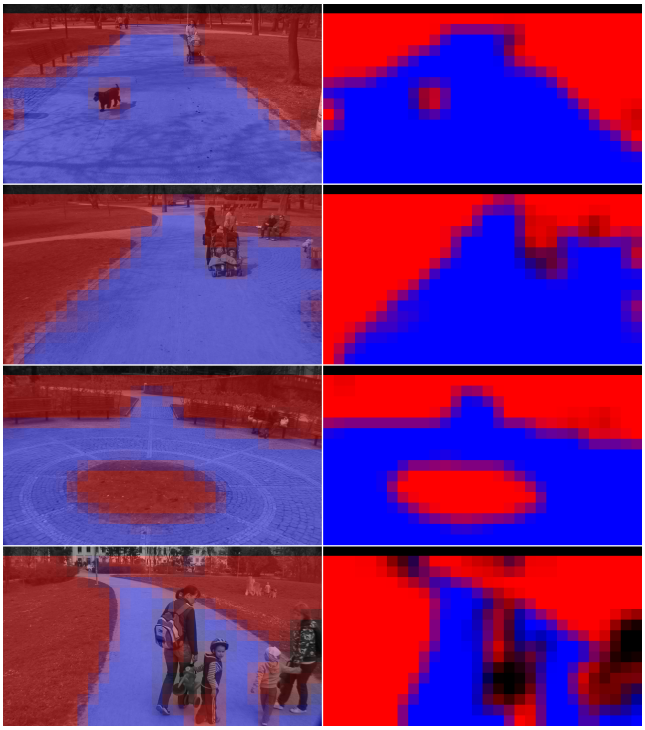


Fig. 6. Examples of classified frames which contain some obstacles. Left column shows frames with classification overlays, right column contains pure classification maps.

Description Interface, B. Manjunath, Ed. New York, NY, USA: John Wiley & Sons, Inc., 2002.

[8] M. Batko, D. Novak, and P. Zezula, "Messif: Metric similarity search implementation framework," in *Digital Libraries: Research and Development*, ser. Lecture Notes in Computer Science, C. Thanos, F. Borri, and L. Candela, Eds. Springer Berlin / Heidelberg, 2007, vol. 4877, pp. 1–10.

Appendix C

Publication about Sisters Classification

The following pages contain the publication, in which the classification of sister cells was used. My contribution to this publication was in evaluation of similarity between sister cells and to support the claim that some particular cells are sisters based on the similarity.