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-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

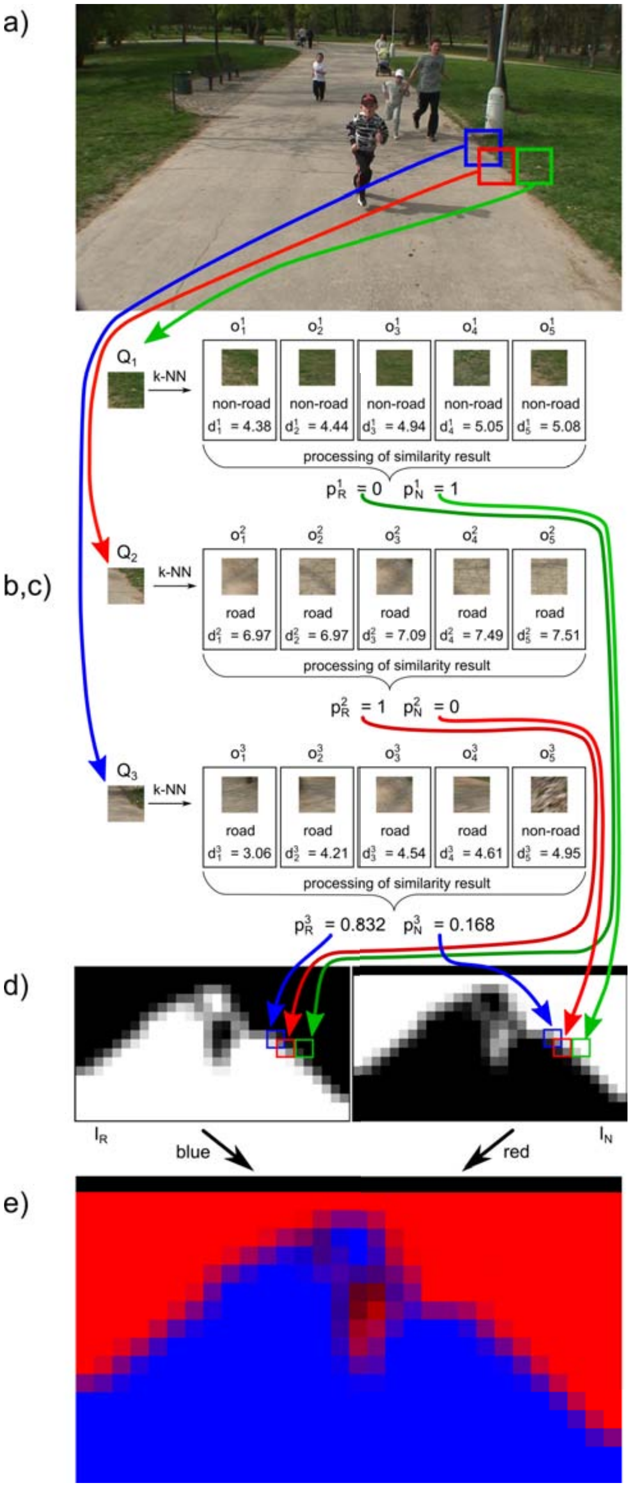


Fig. 1. Illustration of the road detection algorithm. a) Samples Q_1, Q_2 and Q_3 are extracted from the input image. b) Most similar images from database are retrieved for each sample Q_i (for $i \in 1\ 2\ 3$) using k-NN query ($k = 5$). c) Results from similarity database are combined together and likelihoods p_R^i and p_N^i are computed for each region Q_i. d) Values p_R^i and p_N^i 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:

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

  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):

  $$ {}_{j=1}^{k} w_j^i = 1 \qquad (1)$$

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:

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

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:

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

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

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 $\alpha(d)$:

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

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 $T_d, 2T_d$ 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:

$$p_R^i = {}_{j \quad x\ c_x^i = 1} \alpha(d_j^i) \cdot w_j^i \qquad (5)$$

$$p_N^i = {}_{j \quad x\ c_x^i = 0} \alpha(d_j^i) \cdot w_j^i \qquad (6)$$

With these definitions, numbers p_R^i and p_N^i can have a value only from interval $0, 1$ 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.

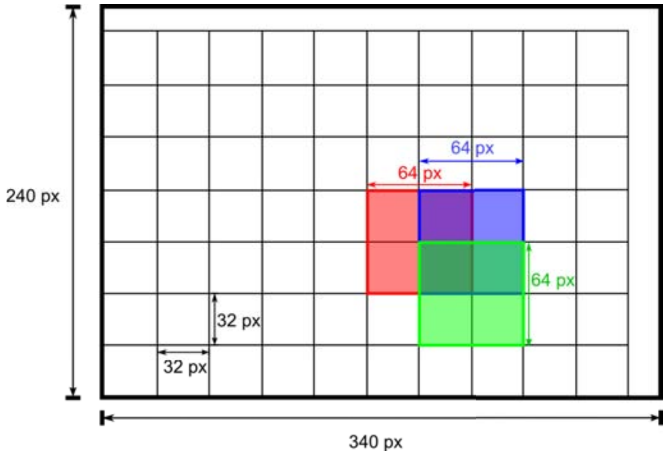


Fig. 2. Segmentation of input image into tiles for similarity search. We used samples of size 64 × 64 px with overlaps of 32 px.

C. Creation of classification map

From the previous step, we have a set of triples $\{(r^1\ p^1_R\ p^1_N) \qquad (r^n\ p^n_R\ p^n_N)\}$, where r^i is i-th region (corresponding to i-th query image Q_i) and p^i_R and p^i_N are the likelihoods defined above. Because regions from $\{r^1 \qquad r^n\}$ 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.

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

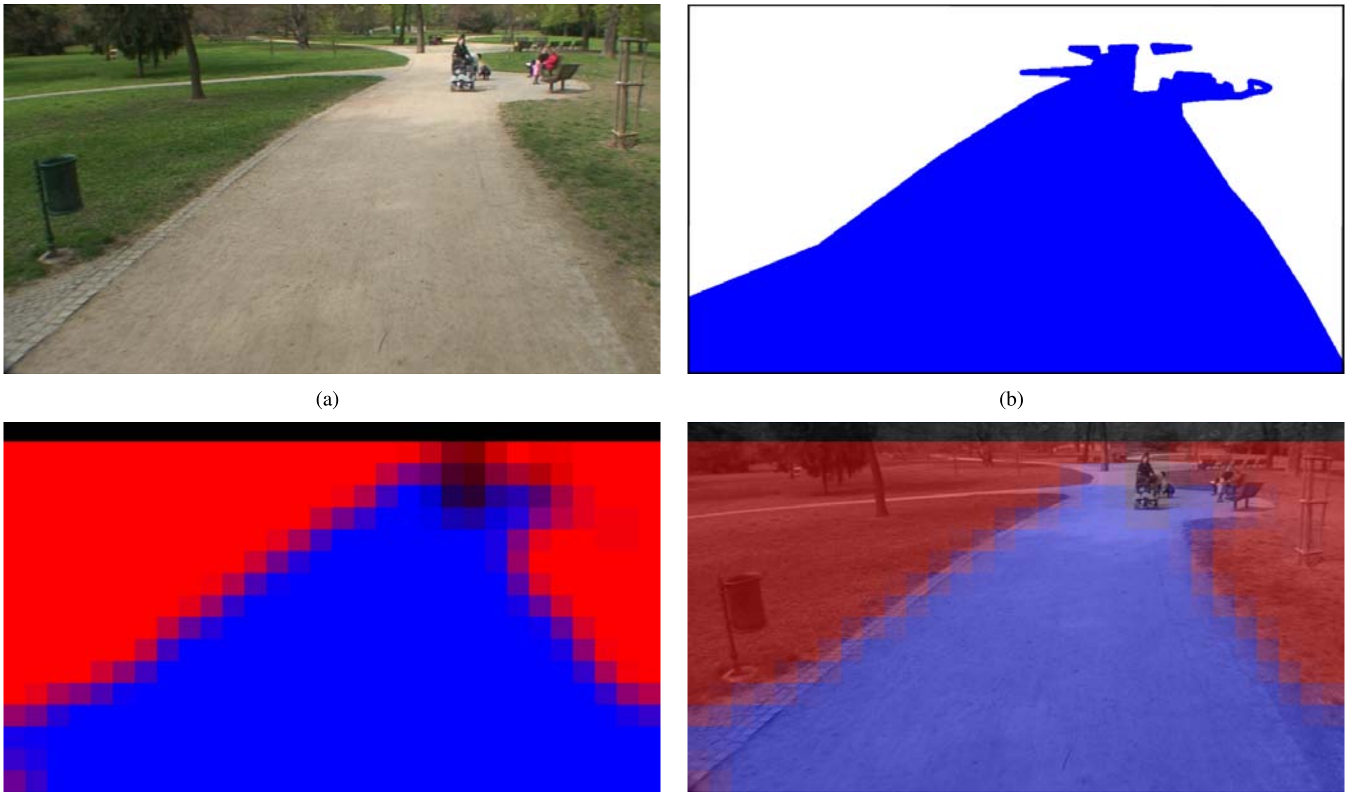
We defined several error metrics in order to evaluate precision of our algorithm in a quantitative way. The amount of an error depends on the two factors: size of the area on which we obtained other than expected result; and also on the difference between expected and actual result.

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:

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

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

[2] park Lužánky, Brno, Czech Republic



(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.

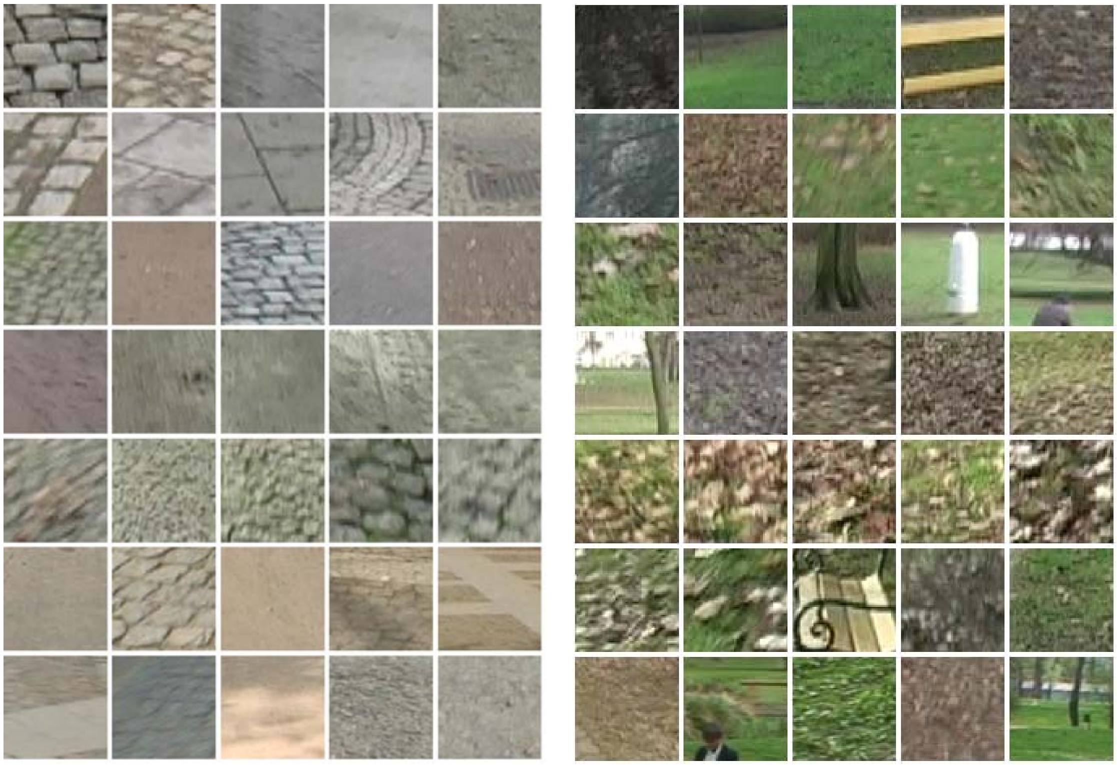
Let I denote the whole classification map encoded as image, I_B denote the blue channel of image I (which contains values of p_R), I_R denote the red channel of image I (which contains values of p_N), GT denote manually defined ground-truth, which contains value 1 for pixels, which represent road and 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.

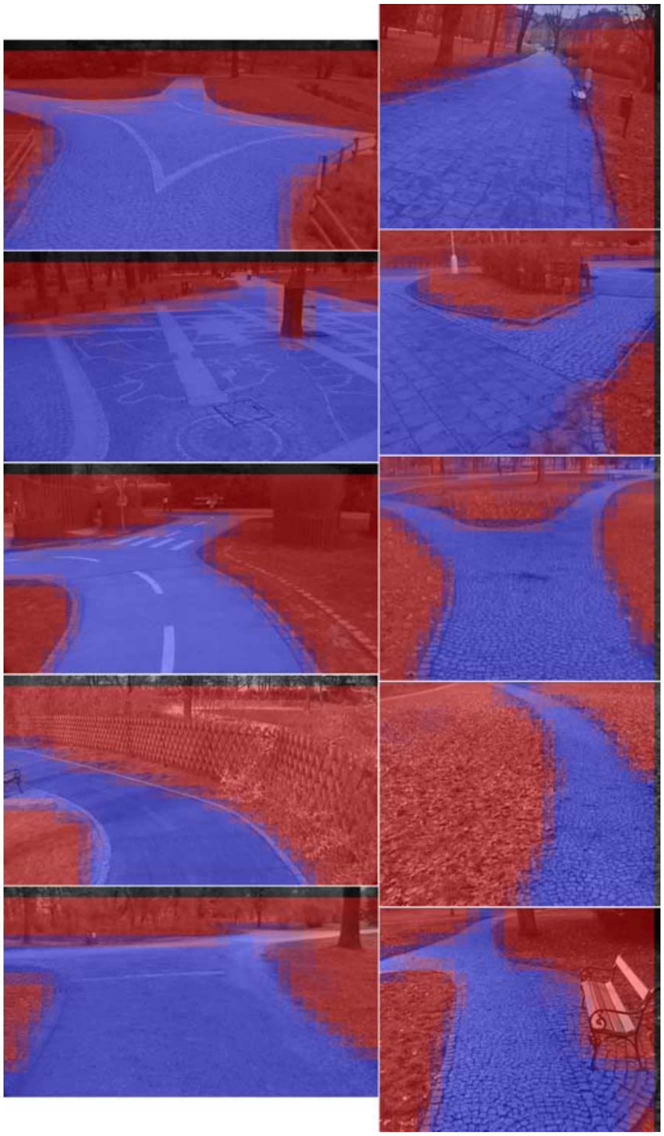


Fig. 5. Examples of classified frames exported from all videos. Frames in the left column are from *walk-03* and *walk-04*, frames in the right column are from *walk-01* and *walk-02*

REFERENCES

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

[2] T. Krajn´k 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. Krajn´k, 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. Roub´ček, and J. Roub´ček, "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*

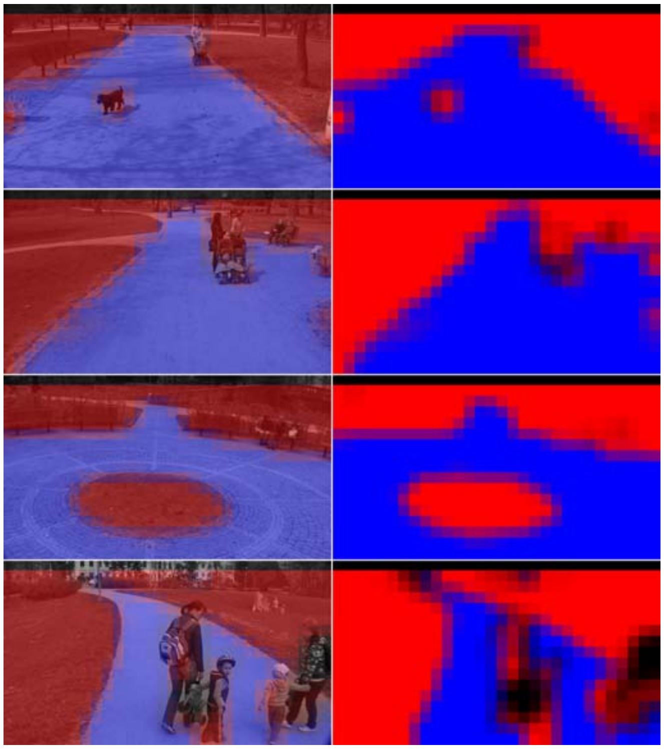


Fig. 6. Examples of classified frames which contain some obstacles. Left column shows frames with classification overlays, right column contains pure classification maps.

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.