Looking for the path: image segmentation

Jan Coufal
Faculty of Mechanical Engineering
Brno University of Technology
Brno, Czech Republic
jan.coufal.cz@gmail.com

Jiri Krejsa
Centre of Mechatronics
Institute of Thermomechanics AS CR, v.v.i.
Brno, Czech Republic
krejsa@fme.vutbr.cz

Abstract—**Processing of the images acquired from the camera attached to the mobile robot in outdoor environment can be used for feature extraction or to distinguish the path the surrounding in the environment. Such information is further used in the planner and/or position estimator. The paper gives the overview of image segmentation method used on real data gathered by mobile robot Bender II during the tests for Robotour 2009 competition.**

Keywords: vision-based navigation, image processing, image segmentation, mobile robots

I. INTRODUCTION

Robust, two-dimensional path following for autonomous robot in outdoor non-urban environment is challenging task of monocular vision navigation. Characteristic features of this problem are shadow and illumination changes, no clear boundaries, changes of road surface and little or no prior knowledge of the roads. The task itself can be simply stated as extraction of road representation features, usually of relatively low data size from the image or a sequence of images acquired by the camera mounted on the robot. Road representation is further used as an input in path planning algorithm that determines subsequent motion of the robot, or together with other sensory inputs in the position estimator in cases when certain map features are known (typically the detection of the cross road)

Motivated by the Robotour 2009 challenge, we have developed a segmentation method invariant to illumination conditions with embedded adaptive database of path color model and direction extraction.

II. RELATED WORK

Using vision like a supplement in robot and vehicles navigation has been popular research field. Areas of the research can be classified depending on road conditions, used sensors and vision algorithms. In our survey, we focused only on monocular color cameras, as the hardware cost in binocular systems capable of well performed synchronization, that is the necessary condition for restoring the 3D distance map from the stereo images is still relatively high. The computational cost of algorithms in processing images from binocular systems is also an issue.

For robots running on well-structured roads, such as roads in urban areas, the primary attention is focused on lane tracking and curve fitting. Since the road has a relatively uniform surface, techniques such as color-based road segmentation [1] and edge detection [2] are used with high percentage of successful features extraction. Even in such well defined roads the road segmentation algorithms are required to exhibit invariance to shadows and luminance. For this purpose, many approaches contain transformation of the source image into different color space [3, 4, 5]

When robot is running in an unstructured environment, terrain classification and obstacle avoidance are in primary focus. Method proposed in [4] is based on construction of 2D scene model of outdoor environment. Image is converted to illumination invariant Ohta/Gevers color space, segmented by hybrid of thresholding and region growing methods and obtained clusters are classified into predefined classes using Support Vector Machines technique by their color and texture properties. Approach gives good results, but is highly dependent on predefined classes. In our method, we wanted to avoid implementing pre-learned knowledge database about environment.

Motivated by DARPA challenge, [5] developed method for direction extraction in desert terrain. His approach uses color transformation to c1c2c3 color space published in [6] for shadow elimination and better segmentation performance. The motion planning is based on a vision vector space, which is unitary vector represents collision-free directions in the image coordinate system. This vector space is projected to a preprocessed set of trajectories and the best candidate is chosen and used for motion planning. With respect to this approach we should state that while the ultimate goal is to develop the universal road/surrounding extractor, certain apriory knowledge regarding the environment can be successfully incorporated into the method. Images acquired from desert terrain exhibit features hard to find in images taken from the park or forest path.

Interesting approach is to involve the mean-shift algorithm for road segmentation and to use graph cuts for region merging. Reference [8] proposes a novel road following method, which firstly uses the Mean-Shift algorithm with embedded edge confidence to partition the images into homogenous regions with precise boundary. Then, according to the color statistic information of the road/non-road model obtained from previous frames, the Graph Cuts algorithm is used to achieve the final binary images and update the road/non-road model simultaneously. This combination of the advantages of Graph Cuts algorithm and Mean Shift algorithm effectively solves some difficult problems of conventional methods, such as the adaptive selection of road model under complex environments, and the choice of effective criteria for the region merging. Problem and main disadvantage is enormous computation time, which makes this method not suitable for real-time. In spite of that the method seems to be reasonable to use in initialize stage or the high evaluation time can be reduced by using the computation on GPU.

III. PROPOSED METHOD

We accept a few assumptions and simplifications for our solution. We assume that the vehicle maintains the contact with ground at all the time during its travel. We also assume that ground surface is relatively flat. Therefore, it can be treated as a ground plane.

The overall scheme of the method is drawn on Fig. 1. Image data are represented by the image matrix of particular color components extracted from the camera, usually in RGB color space.

A. Preprocessing

The purpose of preprocessing is to filter the noise in the image and to cut upper part of the screen. In [8] the road-following technique was presented, which can be used also as a horizon detector. Because of flat ground assumption, we don't have to implement horizon detector. The position of the horizon is constant and it depends only on the known construction parameters of the robot. Data included in the part of the image up to the horizon are useless for path detection and segmentation. Therefore we can cut this part off the image to speed up the performance of the algorithm.

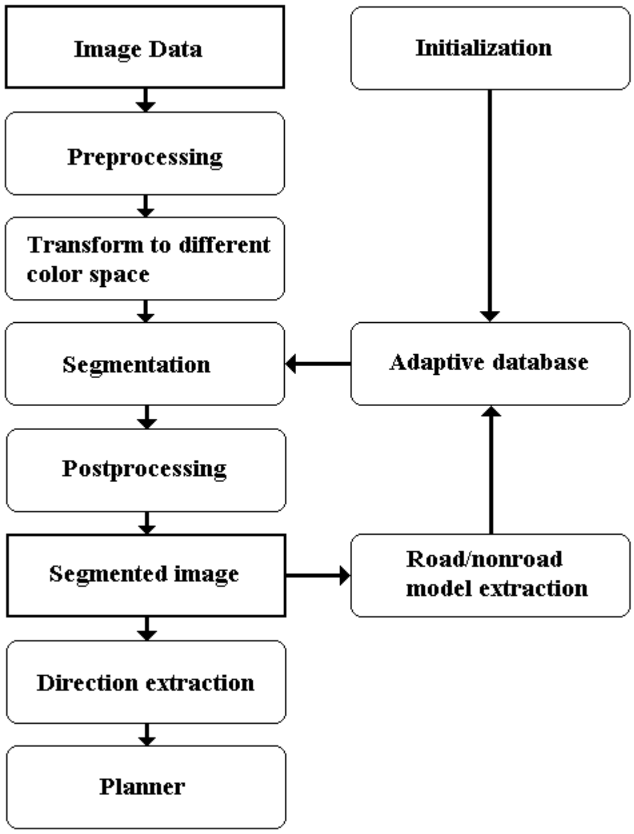

Figure 1. Principle of vision-based navigation algorithm.

Fast convolution with gauss kernel optimized in assembler is used as a noise filter.

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \qquad x, y \in \langle -n, n \rangle \tag{1}$$

where $G(x,y)$ is radial symmetric function for a x-th and y-th element of the kernel matrix of size $(2n+1)^2$.

B. Color correction

The main purpose of color correction is to reduce the shadow and illumination change in the scene that effects acquired original image, as segmentation must be robust against such phenomena. We tested several color models e.g. popular HSI color model, Otha's color space [9], c1c2c3 and l1l2l3 color space described in [6]. Our segmentation experiments show, that c1c2c3 color model is the best shadow and illumination invariant color model for outdoor vision algorithms.

$$\begin{aligned} c_1 &= \operatorname{atan}\left(\frac{R}{\max(G,B)}\right) \\ c_2 &= \operatorname{atan}\left(\frac{G}{\max(R,B)}\right) \\ c_3 &= \operatorname{atan}\left(\frac{B}{\max(R,G)}\right) \end{aligned} \tag{2}$$

The coordinates of c1c2c3 are rescaled to byte values and stored in the bitmap data structure for further use.

C. Segmentation

The essential task of the whole process is to distinguish a road from nonroad surface and this is done by image segmentation. We have tested three bottom-up segmentation principles: simple thresholding, region growing and mean-shift clustering. Our experiments show, that satisfactory results are provided by thresholding method. Region growing and mean-shift methods have better segmentation results, but they are more computationally expensive, as in their nature is to classify all the pixels in the image to the class containing similar points with reference to their spatial domain. In our case, complete segmentation is not necessary. Problem of dealing with wrongly classified pixels is efficiently solved during post-processing.

Thresholding method is computationally fast and efficient segmentation technique. The distance between actual pixel and road representative pixel is computed using chosen metric for

Figure 2. Example of thresholding segmentation and postprocessing. Image was converted to c1c2c3 color space and segmented by thresholding technique. Reference point was obtained from trapezoid region in front of a robot and threshold value is 20. After that, combination of postprocessing methods was applied.

each pixel in image. If the distance between both pixels is less than distance criteria, then the pixel is classified as a road. The condition can be described as follows:

$$dist\|px,\ ref\| \le t_{\max} \tag{3}$$

where *px* and *ref* are points in three dimensional space determined by c1c2c3 color coordinates, *dist* is metric and t_{max} is maximum distance between pixels in chosen metric. For our purposes the Euclidean metric is sufficient, but other metrics can be implemented, e.g. Mahalanobis metric [10].

In our approach, road representative pixel is provided by adaptive database, which stores road representatives from previous images, therefore initialization phase is needed when the first representative is chosen.

Segmented data are stored in array of bit values, where value of one represents road while zero represents the background. This data structure is very useful in post-processing.

D. Post-Processing

In this section, data obtained in segmentation stage are further adjusted. The combination of dilatation and erosion methods is used to delete incorrectly classified pixels out of road and merge larger areas together. Sequence of postprocessing methods is dependent on the magnitude of the threshold value. If threshold value is smaller than hypothetical optimum, number of unrecognized road pixels will be bigger. In this case is better to use “close” method first (dilatation, erosion). If threshold value is higher, number of nonroad pixels classified as a road will be larger. For that reason is better to use “open” method first (erosion, dilatation).

Our experiments shows, that robust results are provided by relatively higher threshold value and corresponding postprocesing combination of methods. In our case we use a small number of iterations of the open method (erode-dilate) and larger amount of iterations of the close method (dilate-erode).

After that, we need to fill small regions in the image. For that purpose we used the connected component labeling technique. It is an algorithmic application of graph theory, where sets of connected regions (components) with the same value in the bit map are uniquely labeled. Number of elements with the same label is counted and if component have less than minimum number of elements, then its value is inverted and component is merged with closest one. Result data structure of the post-processing is bit mask.

E. Direction extraction

In this section we present our approach to achieve the control angle further utilized by mobile robot planner module. Our inputs are a bit mask, which contain information about collision-free space, and angle φ between GPS coordinates of next waypoint (goal) and actual mobile robot coordinates. It is assumed that camera is mounted in the middle plane of the robot, so the vertical center line in the image is the projection of robot actual heading angle. We also assume that velocity of robot is not significant. Therefore the trajectory of robot motion in the image can be represented by the straight line.

Now we can formulate motion planning quantitatively. Lets w_α is a weight function for trajectory in direction determined by steering angle α . We can compute it as follows:

$$w = d(\alpha)^k \cdot \cos(\alpha - \Phi) \tag{4}$$

where k is constant determining nonlinearity between distance of two points in image and distance between two points in real world. Angle ϕ is a difference between waypoint angle φ and heading angle of the robot.

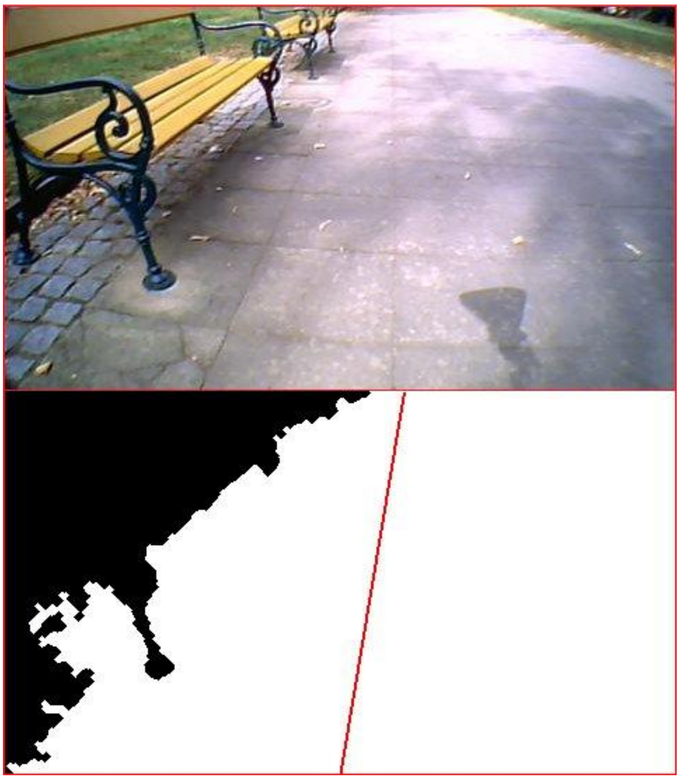

Figure 3. Result of direction extraction algorithm.

Function *d(α)* is number of collision-free pixels in the bit map in direction determined by angle α. We can compute *d(α)* as follows:

$$d(\alpha)=\sum B(x,y) \tag{5}$$

where B is the bit mask obtained during the post-processing and *x, y* are image coordinates. Value of function *B(x,y)* is 1, when value in bit mask at *x,y* coordinate is true, otherwise *B(x,y)* is 0. The value of function *d(α)* is sum of the pixels classified as the road.

Let's denote that *u* and *v* are coordinates in trajectory coordinate space τ with zero point in the center of the bottom edge of screen. Transformation formula between x,y and u,v:

$$\begin{aligned} y &= height - u \\ x &= \frac{width}{2} + v \\ v &= u \cdot \tan(\alpha) \end{aligned} \tag{6}$$

where height and width are the dimensions of the image. The trajectory is fully described by angle α. Therefore, we formulate the motion planning problem as an optimization problem. We are looking for trajectory angle α that maximizes weight function *w*.

There is countless number of potential trajectories and we can't evaluate them all. Therefore we have pre-computed set of trajectories for different α and final trajectory is obtained as their weighted average.

$$\alpha_{final} = \frac{\sum w_i \alpha_i}{\sum w_i} \tag{7}$$

F. Extraction of reference point

At the end of one loop of the algorithm, the resulting bit mask is placed to original image. The result of multiplication of both images is a new image with valid data for extraction of new reference point. With inversion of bit mask we can acquire non-road reference points. Both points (road/non-road) are stored in database.

Using recognized path from last images gives to algorithm adaptability to continuously changing conditions.

G. Initialization stage

In initialization stage we pre-compute all necessary parameters, trajectories and reference point for path recognition. We assume that robot is standing during initialization on the path, heading towards next waypoint.

To extract the reference point the trapezoid region in front of the robot is used [5]. The image for reference extraction must be static. To exclude moving object in image during initialization we implemented autocorrelation function. If two images in sequence are static, autocorrelation function reach maximum and image is suitable for computing of road reference point.

IV. Experiment

We tested our navigation algorithm on real data gathered by mobile robot Bender II during Robotour 2009 competition. The camera used to acquire the images was Megapixel USB2 Wide Angle Webcam Live WB-6200p.

The images on Figure 4 show pictures randomly selected from the robot track. Each figure is composed of four subframes illustrating the output of individual algorithm steps: top left the original image, top right the image converted to c1c2c3 color space, bottom left the threshold segmentation result and bottom right direction extraction.

First image shows the performance of direction extraction algorithm on border of the path. Algorithm reacts correctly. Next two images show segmentation in difficult lighting conditions. Both results are satisfying, i.e. usable by the planner to keep the robot on the path. Last image shows the drawback of presented algorithm. Because the weight is a sum of all positively classified pixels in tested direction, we can expect, that some obstacles will be ignored. This property was implemented on purpose, because small areas of different terrain type may occur and true obstacles are detected by other means (laser rangefinder). Nevertheless, with further research taking into account the shape(s) of segmented portion of the image such issue can be addressed.

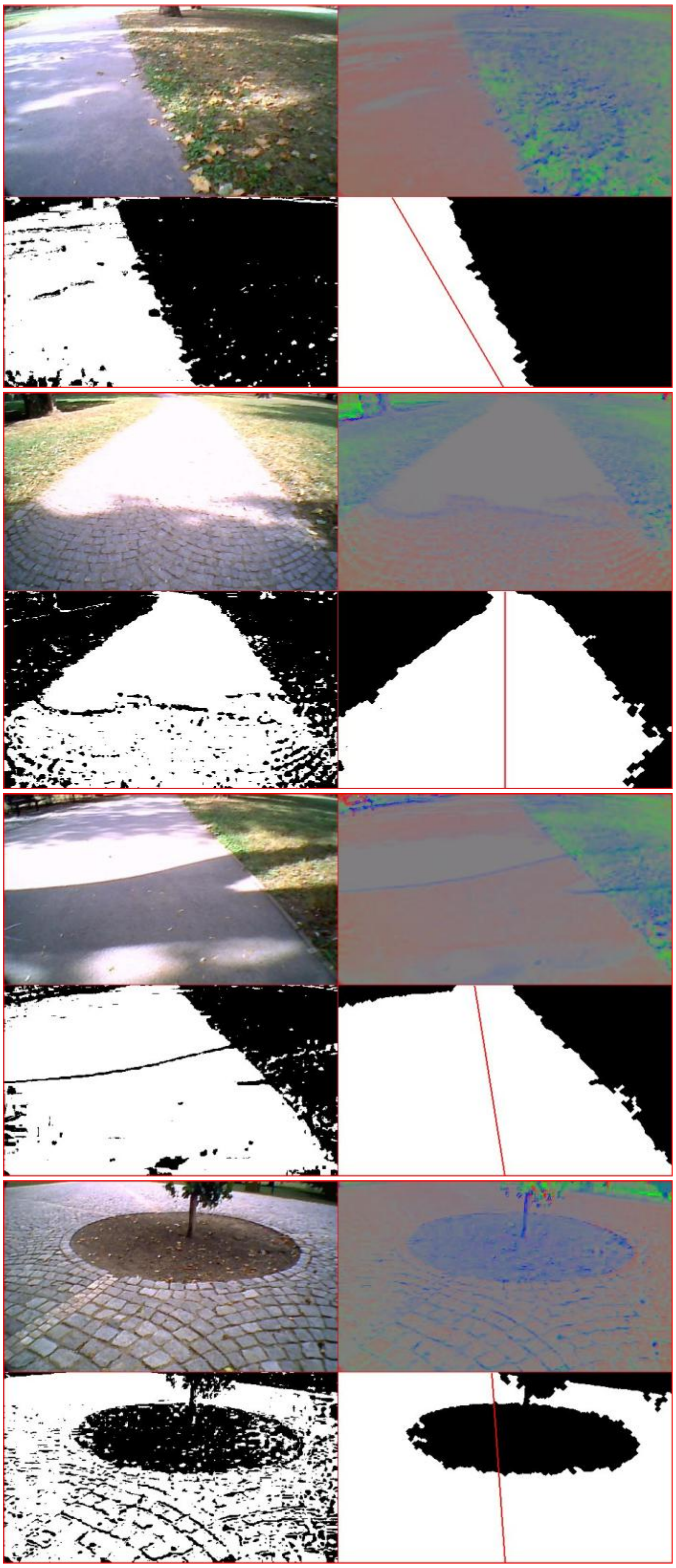

Figure 4. Examples of algorithm performance.

The computation time of the algorithm for a single image of (480x360) pixels without code optimization is around 200ms. The algorithm was implemented in C#. Such speed is sufficient for relatively high velocities of the robot, as the distance traveled during the processing can be kept in reasonable values.

V. CONCLUSION AND FUTURE WORK

In this paper, we report our development of robust vision-based algorithm used for motion planning of autonomous mobile robot. To achieve good performance, data are first filtered to remove the additive noise. We transform regular RGB color coordinates to c1c2c3 color model for better segmentation result as this color model is less sensitive to the changes in lighting conditions. Then threshold segmentation method classifies the image data to road or non-road class with respect to the representative point of the road. Segmented data are stored as bit array and are post-processed. The holes in image are filled and only few continuous clusters are obtained. Bit array is applied on original data and new reference point for segmentation in next frame is stored in the database. Finally, the direction extraction algorithm computes the best angle to achieve desired direction. Such image processing algorithm is not difficult to implement and can be used by other researchers not only in robotics competitions.

Our future work will be focused on reduction of the computation time of complex algorithms via OpenCL for GPU programming, as most of the image processing routines can be easily transferred to semi-parallel versions, ideal for multiple processors used on GPU. With this technology, we will be able to increase processed images frame rate to high values on commonly available hardware.

ACKNOWLEDGMENT

Authors thank for the support of the Academy of Sciences of the Czech Republic under the research plan AV0Z20760514 and for the support of the Operational Programme Education for Competitiveness, project number CZ.1.07/2.3.00/09.0162.

REFERENCES

[1] M. Sotelo, F. Rodriguez, L. Magdalena, L. Bergasa and L. Boquete, "A color vision-based lane tracking system for autonomous driving in unmarked roads" *Autonomous Robots*, vo.16, no. 1, 2004

[2] M. Ekinci, F. W. J. Gibbs, B. T. Thomas, "Konwledge-based navigation for autonomous road vehicles" Turkish Journal of Electrical Engineering & Computer Sciences, 2000

[3] Alvarez, J.M & Lopez, A.M. & Baldrich, R., "Shadow resitstant road segmentation from a mobile monocular system" *Iberian Conference on Pattern Recognition and Image Analysis*, vol. II, 9-16, 2007

[4] Juan Gabriel Aviña-Cervantes, Michel Devy, "Scene Modeling by ICA and Color Segmentation" MICAI 2004: 574-583

[5] D. Song, H. N. Lee, J. Yi, and A. Levandowski, "Vision-based motion planning for an autonomous motorcycle on ill-structured roads" *Auton Robot*, vol. 23, pp. 197–212, 2007.

[6] Gevers, T., & Smeulders, A.W. M. (1999). "Colour based object recognition" *Pattern Recognition*, 32, 453–464.

[7] Lin, L. Zhou, W., "A Robust and Adaptive Road Following Algorithm for Video Image Sequence", *Lecture Notes in Computer Science*, 2007, NUMB 4681, 1041-1049

[8] Rasmussen, C. "Grouping dominant orientations for illstructured road following" In *Proceedings of IEEE computer society conference on computer vision and pattern recognition*, June 2004.

[9] Ohta, Y., "Knowledge-Based Interpretation of Outdoor Natural Color Scenes" *Research Notes in Artificial Intelligence 4*, Pitman Advanced Publishing Program (1985)

[10] Grudic, G., & Mulligan, J., "Outdoor path labeling using polynomial mahalanobis distance" *In Robotics: science and systems*, Philadelphia, PA, June 2006.