Czech Technical University in Prague

Faculty of Electrical Engineering

Department of Cybernetics

DIPLOMA THESIS

Visual Path Detection for Mobile Robot Navigation

Author: Bc. Dina Sushkova

Advisor: Ing. Karel Košnar, Ph.D.

January 2017

Czech Technical University in Prague
Faculty of Electrical Engineering

Department of Cybernetics

DIPLOMA THESIS ASSIGNMENT

Student: Bc. Dina Sushkova

Study programme: Cybernetics and Robotics

Specialisation: Robotics

Title of Diploma Thesis: Visual Path Detection for Mobile Robot Navigation

Guidelines:

1. Study methods for shadow removal from image.
2. Study methods for visual path detection.
3. Design algoritms for real-time path detection and robot control for path following.
4. Verify methods on public datasets.
5. Perform experiments with real robot.
6. Evaluate preformance of designed algorithms.

Bibliography/Sources:

[1] Tomáš Krajník, Jan Blažíček, Joao M. Santos, "Visual Road Following Using Intrinsic Images", Mobile Robots (ECMR), 2015 European Conference on, vol., no., pp. 1-6, 2-4 Sept. 2015

[2] José M. Álvarez and Antonio M. López, "Road Detection Based on Illuminant Invariance", IEEE Transactions on Intelligent Transportation Systems, Vol. 2, No.1, March 2011

[3] G. Finlayson, M. Drew, and C. Lu, "Entropy minimization for shadow removal", International Journal of Computer Vision, 2009

[4] Hailing Zhou, Hui Kong, Lei Wei, Douglas Creighton, Saeid Nahavandi, "Efficient Road Detection and Tracking for Unmanned Aerial Vehicle", IEEE Transactions on Intelligent Transportation Systems, Vol. 16, No. 1, February 2015

[5] Huan Wang, Yan Gong, Yangyang Hou, and Ting Cao, "Road Detection Based on Image Boundary Prior", Springer International Publishing Switzerland, 2015

Diploma Thesis Supervisor: Ing. Karel Košnar, Ph.D.

Valid until: the end of the summer semester of academic year 2016/2017

L.S.

prof. Dr. Ing. Jan Kybic
Head of Department

prof. Ing. Pavel Ripka, CSc.
Dean

Prague, December 16, 2015

Acknowledgement

I would love to thank my thesis advisor Ing. Karel Košnar, Ph.D. for his help and guidance during my work.

I express my thanks to my parents for their support during my studies and to my brother for being a great inspiration.

I would also like to thank my course mates, professors, teachers and other CTU FEE staff for making my masters studies enlightening and interesting, and for making CTU FEE an alma mater to be proud of.

Prohlášení autora práce

Prohlašuji, že jsem předloženou práci vypracovala samostatně a že jsem uvedla veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

Author statement for undergraduate thesis:

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

Prague, date.................. ....
signature

Anotace

Schopnost najít sjízdnou cestu a zůstat na ní během pohybu je pro mobilního robota nezbytné. V této práci se zaměřuji na vizuální detekci cesty z jednoho obrázku. Naimplementovala jsem metodu založenou na vnitřní vlastnosti RGB obrázku—nezávislosti na zdroji osvětlení. Tato metoda dovolí převést barevný obrázek do odstínů šedé, ve které nejsou stíny. Detekce cesty je udělaná na šedém obrázku, kde cesta má jednotnou barvu nezávislou na osvětlení v původním obrázku. Druhá naimplementovaná metoda je řez grafem založeném na směsi Gaussovských vícerozměrných rozdělení (GMM). Tato metoda využívá barevné modely a informaci o struktuře, aby vytvořila z obrázku ohodnocený graf. Detekce cesty je pak řešena nalezením minimálního řezu. Kombinace obou metod je řešena úpravou váhy hran v závislosti na výsledcích detekce cesty na obrázku bez stínu. Implementované algoritmy byly vyzkoušeny na veřejně dostupném datasetu a během experimentů s robotem. Metoda založená na nezávislosti na osvětlení a kombinace obou metod obstály dobře v testech. Metoda řezu grafem neměla tak dobré výsledky, přesto se prokazala jako nadějná.

Klíčová slova

Nezávislost na zdroji světla, výpočet minimální entropie, obraz bez stínů, detekce oblohy, chromatický prostor, směs Gaussovských vícerozměrných rozdělení, detekce cesty, sledování cesty

Annotation

The ability to find the drivable area and stay in it during the movement is essential for mobile robot. In this thesis I focus on visual road detection in a single image. I implement a method based on the intrinsic illuminant invariant feature of the RGB image. This method allows to transform color images into the grayscale shadow free images. The road detection is performed on the grayscale image where the road area have a uniform gray level independent on the light conditions in the original image. The other implemented algorithm is a Gaussian Mixture Models (GMM) based graph cut. This method uses color models and structure information to compose a graph from the image. The road detection is based on finding the minimum cut in the graph. The combination of illuminant invariant and GMM based methods is performed by altering edge weights according to results of illuminant invariant road detection. Implemented algorithms were tested on the publicly available dataset and in experiments with real robot. The illuminant invariant and combined method performed well in tests. The graph cut method didn't have such good results, nevertheless it still looks as a promising method.

Keywords

Illuminant invariance, entropy minimization, shadow free image, sky detection, chromaticity space, Gaussian Mixture models, road detection, path following

Contents

Appendices

List of Figures

List of Tables

List of algorithms

Chapter 1

Introduction

One of the most essential robot skills in the field of mobile robotics is to be able to identify driveable area and to be able to stay in it during the movement. To perform this task, robots are equipped with variety of sensors allowing to gain information about the surroundings. Sensors which can be used for road detection include mostly cameras and laser rangefinders. Many researchers focus on visual road detection because cameras are cheap and are able to provide the robot with a lot of useful information. From a single image it's possible to extract region and boundary features [1]. Region-based features include color, texture and illuminant invariance, while the boundary is defined mostly by road borders and vanishing point. By combining two cameras it's even possible to use stereo vision to recover depth information [2]. Some studies rely only on one image feature in road detection, while other combine different approaches or utilize information from other sensors [3].

This thesis is focused on the visual path detection in a single image from a single camera. The main approach I've taken is to use the illuminant invariance intrinsic property of the image. This method is based on a physical theory of a color image formation. According to the theory, RGB images from narrow-band cameras have a property allowing to construct an illuminant invariant, e.g. shadow free, image. In such image, road area would have a uniform color independent of the light conditions in the original image.

The other approach I implement is a road detection based on the road color. It utilizes Gaussian Mixture Models (GMMs) in order to represent road and non-road color models. The method combines color information with edge detector to compose a graph. The road detection task is therefore transformed into finding this graph's minimum cut. Furthermore, I implement a combination of both methods in attempt to improve the results.

1.1 Thesis outline

The thesis is composed of 5 chapters. In chapter 1—Introduction—I describe the thesis structure and present state of the art methods of road detection in section 1.2.

In chapter 2 I introduce the theory of the illuminant invariance. I describe how the shadow free image is formed and how the invariant angle parameter is computed. I also describe the sky detection method. Afterwards, I present the methods of road detection in the illuminant invariant image.

In chapter 3 I describe the usage of Gaussian Mixture Models in road detection task, present the road detection technique and propose a combination of GMM method with illuminant invariance algorithms.

In chapter 4 I present performed experiments, explain the parameter estimation and evaluate the results.

1.2 State of the Art in Visual Road Detection

The field of visual road detection is based on the usage of the characteristic road properties presented in images. The most approaches found in science literature are focused on the color or/and structure properties of roads. Various combinations of both approaches are often used to increase robustness of the results.

One of the color-based approaches is to use Gaussian Mixture Models (GMMs), as described in [3]. Authors use GMMs to create the road model and use it co classify pixels in the input image. The update of road model is performed to adapt the model to the changing environment, the approach though uses laser range finder to select pixels for update.

Other example of GMM implementation can be found in [4], where road detection for aerial vehicle is considered. Authors propose an interesting approach of combining road and non-road models created using GMMs with edge detection to find the road using min-cut/max-flow algorithm. Road detection is then combined with homography based road tracking to speed up the overall algorithm.

Another approach was used in [5], where authors use algorithm composed of two modules. First uses intensity image to find candidates to the road boundary. The second module is based on the assumption, that road region color can be described by the multivariate Gaussian distribution. Here, the color information is used to find the road region and reinforce the best matching boundary.

A big area of visual road detection is composed of algorithms based on the illuminant invariance theory. This theory as described in [6] allows to create shadow free grayscale images and even reconstruct colorful shadow free images. In road detection though the grayscale image is sufficient to find road regions and color reconstruction based on given approach is rarely (if ever) used. In [7] authors create grayscale shadow free image using the mentioned approach and find the road areas according to the likelihood-based classifier. In [8] authors also create illuminant invariant image. The road classification is performed by equalization of the grayscale image and subsequent thresholding.

Authors of [2] propose to improve the results of the illuminant invariant image formation by ignoring the sky area of the image. Authors also propose another approach to road-nonroad classification based on the use of confidence intervals. To further improve results, algorithm utilizes the stereo-vision allowing to reconstruct the depth information.

Authors of [9] proposed to use image boundary prior to overcome the problems associated with usage of seeds. The assumption was made in many studies that the center-lower part of an image is a road region. In some cases this isn't necessary true, therefore authors proposed to use another assumption, that the road region has the large common boundary with the bottom part of an image. The color information is used to over-segment the image into several patches. The segmented parts are then classified as road or non-road based on the mentioned assumption. Moreover, the illuminant invariant grayscale image is embedded into the algorithm to eliminate the effect of shadows.

The approach presented in [10] proposes the method to detect unstructured roads. Authors use texture information at every pixel to detect the vanishing point of the main part of the road. The found vanishing point is then used to estimate two most dominant edges and segment the road.

Different approach is described in [11]. Authors implemented the combination of scene and pixel-based classifiers. The scene classifier uses learned road geometry models to estimate the type of the input image and create a road probability map. The computed map is then combined with the probability map of the pixel-based classifier to improve the result.

The survey comparing some of the mentioned approaches and more is presented in [1].

Chapter 2

Illuminant invariant images

One of the challenges of an autonomous robotics in the outdoor environment is the detection of the free road space. This task is complicated by various factors, including the variations in the road and background colors and textures, presence of obstacles, illumination conditions. The last is responsible for the appearance of shadows, which create the undesirable color variance on the road surface. Particularly challenging cases are when the road contains both illuminated and shaded regions, as they tend to be classified as different classes. One of the possible approaches to overcome this obstacle is to create an illuminant invariant image and use it to detect the road regions. In this chapter the method to create an illuminant invariant image using the intrinsic features of an RGB image is described.

2.1 RGB image formation

The RGB color is defined as a triple $\mathbf{R} = (R_1, R_2, R_3) = (R, G, B)$. The value of each component depends not only on actual color of an object, but is also affected by various factors such as shape and reflectance of objects, properties and location of the illuminant and observer [12]. Description of the pixel formation, accounting all these parameters is a complex task. Therefore to make problem simpler, researches use simplified models. The theory of the shadow removal considers, that the world consists of Lambertian surfaces, i.e. surfaces which reflect the light in all directions. This assumption can be used in practice, as many real surfaces are close to Lambertian, as stated in [13]. Under this simplification the pixel formation can be described [6] by the following formula:

$$R_k = \sigma \int E(\lambda)S(\lambda)Q_k(\lambda)\mathrm{d}\lambda, \qquad k = 1,\ 2,\ 3, \tag{1}$$

where λ is wavelength, σ is Lambertian shading. $E(\lambda)$ is the spectral power distribution of the illuminant, denoting the amount of power emitted at each wavelength from the interval of the visible spectrum. $S(\lambda)$ is the surface spectral reflectance function, defining the amount of the incident light which was reflected from the surface at each wavelength. $Q_k(\lambda)$ is the camera sensor sensitivity function, describing the proportion of the incident light absorbed by the sensor k at each wavelength [12, 14].

If the camera sensor sensitivity function is equal to Dirac delta function $Q_k(\lambda) = q_k\delta(\lambda - \lambda_k)$, then equation (1) is simplified to:

$$R_k = \sigma E(\lambda_k)S(\lambda_k)q_k. \tag{2}$$

In practice, camera sensors aren't manufactured to be sensitive to one specific wavelength. Typical camera sensor is sensitive to a range of wavelengths with one maximum. The example sensor response is shown in figure 1 (a), where the three sensors, Red, Green and Blue, of Sony DXC930 camera are depicted. Authors of [6] claim, that although real sensors aren't described precisely by Dirac delta function, as in figure 1 (b),

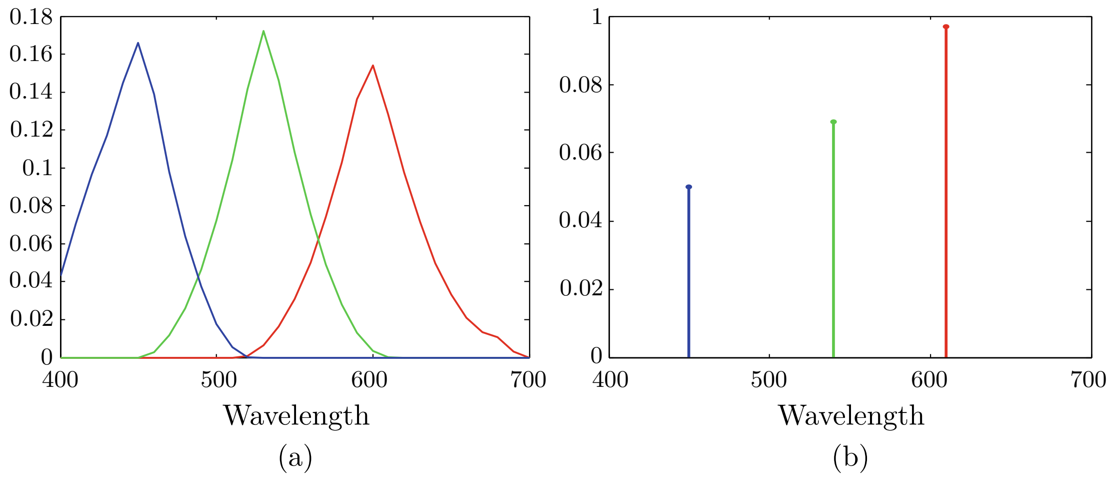


Figure 1 **(a)**: Typical RGB camera sensors—Sony DXC930 camera. **(b)**: Theoretical narrow-band RGB camera sensors.
From International Journal of Computer Vision, "Entropy Minimization for Shadow Removal", Volume 85, Issue 1, p 41, by Graham D. Finlayson, Mark S. Drew, Cheng Lu. ©Springer Science+Business Media, LLC 2009. Reprinted with permission of Springer.

the theory can still be used if sensors are sufficiently narrow-band. Even in case, when sensors are broad-band and described theory doesn't apply, good results may still be obtained if sensors are appropriately transformed. The possible approach may be spectral sharpening proposed in [15]. The goal of this approach is to create new sensor sensitivities by finding a linear combination of the given set of sensors, for which the result is the most narrow-band. Other possible technique is to use sensor transforms deliberately to enhance the resulting invariant image, as proposed in [16].

The next assumption is that the illumination can be described by Planck's law in Wien's approximation as in [17]:

$$E(\lambda, T) \approx \frac{a_1}{n^2\lambda^5} e^{-\frac{a_2}{nT\lambda}} = \frac{2\pi h c_0^2}{n^2\lambda^5} e^{-\frac{-hc_0}{nkT\lambda}} \text{ W/m}^2, \tag{3}$$

where $a_1 \approx 3.742 \times 10^{-16}$ Wm^2 and $a_2 \approx 1.439 \times 10^{-2}$ mK are constants, h is Planck constant, k is Boltzmann constant, c_0 is the speed of light in vacuum. n is the refractive index, T is the temperature of a black-body radiator in Kelvins. Since $n = 1.0001$ in air, it can be omitted. The Wien's approximation can be used if $e^{(a_2/(n\lambda T))} \gg 1$. According to [18], temperatures up to $T = 10000\ K$ can be expected in real applications, and the visible spectrum consists of wavelengths in range from 400 to 700 nm. Therefore, the minimum expected value is $e^{(a_2/(n\lambda T))} \approx 7.810$ and the Wien's approximation can be used. The authors have also compared Planck's equation and the Wien's approximation of several black-body radiants expected in real-life applications, claiming the results are similar.

Equation (3) describes shapes of the power distribution, but doesn't take into account illuminant power. Therefore the light intensity constant I is added to Wien's approximation to model varying power:

$$E(\lambda, T) \approx I a_1 \lambda^{-5} e^{-\frac{a_2}{T\lambda}}. \tag{4}$$

After substituting equation (4) into equation (1), components of the RGB color **R** are given as:

$$R_k = \sigma I a_1 \lambda_k^{-5} e^{\left(-\frac{a_2}{T\lambda_k}\right)} S\left(\lambda_k\right) q_k, \qquad k = 1,\ 2,\ 3, \tag{5}$$

2.2 Chromaticity space

The chromaticity is a color specification, that isn't dependent on its luminance [6], i.e. "quotient of the luminous intensity and the projected area of the source in a given direction" as defined in [14]. According to the same source, this quantity "correlates with perceived brightness". In chromaticity specification individual pixels lose information about their luminance. But that doesn't mean, that such image is shadow free, because luminance isn't the only parameter, which affects shadow formation. As stated in [6], the difference between non-shadowed and shadowed regions isn't that ones are illuminated and others aren't. In most cases they both are illuminated, but by different sources. For example, at daylight shadow is illuminated by sky only, while non-shadowed areas by both sky and sun. These two illuminants have different temperatures, which also affect pixel formation, according to equation (5). Since temperature information can still be present in chromaticity space, shadows don't disappear.

The chromaticity is defined in such a way, that many different descriptors can comply with it. In this section three of them are described, namely, L$_1$ norm, Band-ratio log chromaticity and Geometric mean log chromaticity.

L$_1$ norm

One of the most known chromaticity spaces is defined using the L$_1$ norm. Components $\mathbf{r}$ of this space are created by normalizing components of a RGB pixel $\mathbf{R}$ according to the formula:

$$\mathbf{r} = (r_1, r_2, r_3) = (r, g, b) = \left(\frac{R_1}{R_1 + R_2 + R_3}, \frac{R_2}{R_1 + R_2 + R_3}, \frac{R_3}{R_1 + R_2 + R_3}\right). \tag{6}$$

As emerged from the equation (5), the light intensity I and the Lambertian shading σ are canceled out in each component of $\mathbf{r}$:

$$r_k = \frac{\sigma I a_1 \lambda_k^{-5} e^{\left(-\frac{a_2}{T\lambda_k}\right)} S\left(\lambda_k\right) q_k}{\sigma I a_1 \sum_{i=1}^{3} \left(\lambda_i^{-5} e^{\left(-\frac{a_2}{T\lambda_i}\right)} S\left(\lambda_i\right) q_i\right)}, \quad k = 1, 2, 3. \tag{7}$$

The image formed by $\mathbf{r}$ is thus intensity free. Unfortunately, shadows are not removed by this simple operation, as luminant-dependent parameter (temperature T) is present in equation (7). L$_1$ norm removes luminance, but due to its properties, it's not useful in illuminant invariance image formation, although it's still used in color recovery algorithms.

Band-ratio log chromaticity

Contrarily, band-ratio and geometric mean log chromaticity are defined in such a way, that they can be used in shadow removal task. The band-ratio chromaticity space is defined by obtaining band-ratios $\boldsymbol{c} = (c_1, c_2)$ as:

$$c_k = \frac{R_k}{R_p}, \tag{8}$$

where p is one of the channels $p \in \{1, 2, 3\}$ and $k = 1, 2$ indexes the remaining two channels .

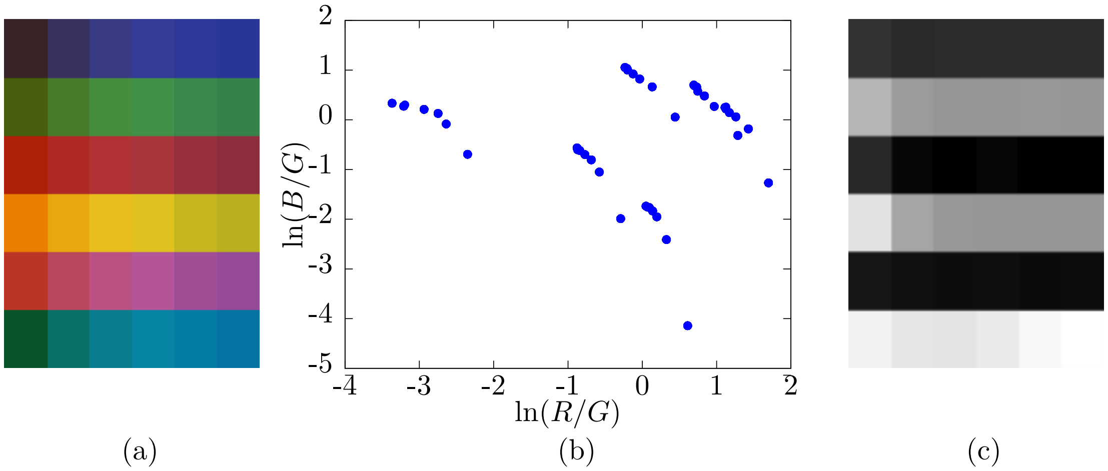


Figure 2 **(a)** The set of 6 patches under different Planckian lights. **(b)** Log-chromaticity plot of image (a). **(c)** The resulting illuminant invariant image.

The created chromaticity space is free of intensity and Lambertian shading information, since according to the equations (8) and (5), corresponding components are canceled out:

$$c_k = \frac{R_k}{R_p} = \frac{a_1 \lambda_k^{-5} S\left(\lambda_k\right) q_k e^{\left(-\frac{a_2}{T\lambda_k}\right)}}{a_1 \lambda_p^{-5} S\left(\lambda_p\right) q_p e^{\left(-\frac{a_2}{T\lambda_p}\right)}}. \tag{9}$$

The band-ratio log chromaticity space $\boldsymbol{\rho} = (\rho_1, \rho_2)$ is obtained by taking the logarithm of $\boldsymbol{c}$. The components of this chromaticity space have a form

$$\rho_k = \log\left(c_k\right) = \log\left(\frac{s_k}{s_p}\right) + \log\left(\frac{e^{(e_k/T)}}{e^{(e_p/T)}}\right) = \log\left(\frac{s_k}{s_p}\right) + \frac{e_k - e_p}{T}, \tag{10}$$

where $s_{k,p} = a_1 \lambda_{k,p}^{-5} S\left(\lambda_{k,p}\right) q_{k,p}$ and $e_{k,p} = -a_2/\lambda_{k,p}$. Evidently, this space is still luminance-free.

The illuminant invariant space can be obtained by extracting the last luminant-dependent parameter T from definition (10) of ρ_1 and substituting it to the ρ_2, giving

$$\rho_2 = \log\left(\frac{s_2}{s_p}\right) - \frac{e_2 - e_p}{e_1 - e_p} \log\left(\frac{s_1}{s_p}\right) + \frac{e_2 - e_p}{e_1 - e_p} \rho_1. \tag{11}$$

This equation shows, that projections of pixels to the band-ratio log-chromaticity space form parallel lines with the same slope. The offsets are dependent on surfaces. The individual points on the same line represent the difference in the illumination and shading, according to [7].

The example of how the log-chromaticity plot is formed is shown in figure 2. In figure (a) the set of 6 patches from the Macbeth color checker is shown under 6 different Planckian lights in a range from 2500 K to 10000 K. Figure (b) shows where these patches are projected in the log-chromaticity plot. Here, the straight lines formed due to the change of the illuminant can be seen. In figure (c) the resulting illuminant invariant image is shown.

The illuminant invariant image can be obtained by projecting points of the band-ratio log-chomaticity space onto the direction $\mathbf{v}^{\perp}$ perpendicular to the vector $\mathbf{v} = (e_2 - e_p, e_1 - e_p)^T$. From the resulting set of scalars the grayscale shadow invariant image can be produced.

Geometric mean log chromaticity

The band-ratio log chromaticity can be used to create shadow invariant images. But it's necessary to choose, which channel should be used as a divisor. If an image happens to have all pixels with small red values and the red channel is chosen as a divisor, then outliers may occur. Various light conditions such as bright sun, clouds or shadows may result in different colors to be more or less intense. Additionally if the camera's color balance isn't properly set, then it may happen that some of the channels lack intensity. In the changing environment it's hard to predict what light conditions can be expected and how would they affect the color balance. The possible method to overcome these limitations is to divide by the geometric mean of all channels, instead of preferring one of them. Such divisor increases the robustness to noise [19], as the small value in one of the channels can be compensated by the remaining channels, effectively reducing the number of the outliers.

The geometric mean chromaticity space is created using vectors $\boldsymbol{c} = (c_1, c_2, c_3)$, where the components are

$$c_k = \frac{R_k}{\sqrt[3]{R_1 R_2 R_3}}, \tag{12}$$

and $k = 1, 2, 3$. Such definition of $\mathbf{c}$ is the chromaticity, since the intensity and shading information are canceled out:

$$c_k = \frac{\sigma I a_1 \lambda_k^{-5} \exp\left(-\frac{a_2}{T\lambda_k}\right) S\left(\lambda_k\right) q_k}{\sigma I a_1 \sqrt[3]{\prod_{i=1}^{3} \left(\lambda_i^{-5} e^{\left(-\frac{a_2}{T\lambda_i}\right)} S\left(\lambda_i\right) q_i\right)}}, \tag{13}$$

where $k = 1, 2, 3$.

The geometric mean log chromaticity space is derived from $\boldsymbol{c}$ by taking the natural logarithm of individual components

$$\rho_k = \ln\left(c_k\right) = \ln\left(\frac{s_k}{\sqrt[3]{\prod_{i=1}^{3} s_i}}\right) + \frac{1}{T}\left(e_k - \frac{1}{3}\sum_{i=1}^{3} e_i\right) = \ln\left(\frac{s_k}{s_m}\right) + \frac{e_k - e_m}{T}, \tag{14}$$

where $s_{k,i} = a_1 \lambda_{k,i}^{-5} S\left(\lambda_{k,i}\right) q_{k,i}$ and $e_{k,i} = -a_2/\lambda_{k,i}$. This equation looks similar to the equation (10). However, the pixel from geometric mean log chromaticity space is composed from three components, not two. To obtain the illuminant invariant image from this space, additional computations presented in the next section are necessary.

2.3 Obtaining the grayscale shadow invariant image

Due to the property of the band-ratio log-chromaticity space to form straight lines from the RGB pixels formed on the same surface, it can be used to create shadow free images. As will be shown below, the geometric mean log chromaticity space can be transformed to have similar properties.

Band-ratio log chromaticity space

The situation described by the equation (11) is depicted in the figure 3. The individual points in the log-log plot form parallel lines. The slope is dependent only on the camera parameters and is defined by vector $\mathbf{v} = (e_2 - e_p, e_1 - e_p)^T$. The offset is dependent

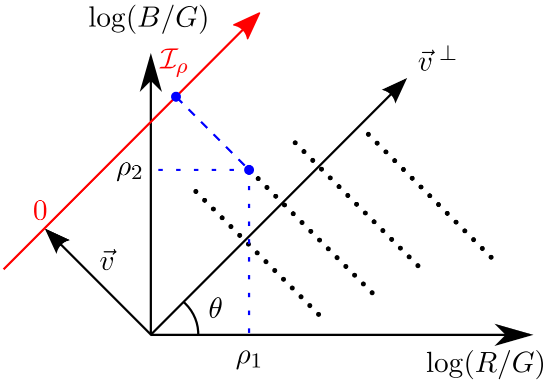


Figure 3 Obtaining the 1D shadow invariant image from the band-ratio log-chromaticity space.

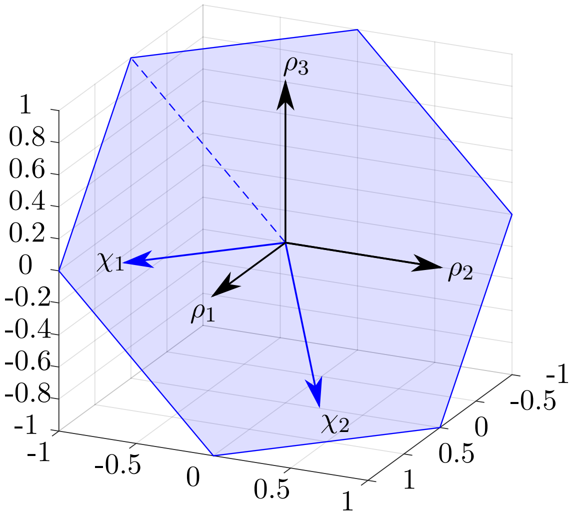


Figure 4 The plane on which components $\boldsymbol{\rho}$ of geometric mean log chromaticity space are placed.

on the surface, but not on the shading, illumination nor temperature. Thus projecting points on the line $\mathbf{v}^{\perp}$ perpendicular to the vector $\mathbf{v}$ loses the shading information, while still distinguish individual surfaces. The slope of the line $\mathbf{v}^{\perp}$ is given by the camera dependent parameter called illuminant-invariant angle θ [7]. With such definition of θ, the projection of the point $\rho = (\rho_1, \rho_2)$ onto the line $\mathbf{v}^{\perp}$ can be calculated using the formula

$$\mathcal{I}_\rho = \rho_1 \cos(\theta) + \rho_2 \sin(\theta)\,. \tag{15}$$

The angle θ isn't known and has to be correctly estimated, so that the resulting grayscale image is shadow free. The methods to find the illuminant-invariant angle are described in section 2.4.

Geometric mean log chromaticity space

Unlike the previous case, the geometric mean log chromaticity pixels are composed of three components. In this case the grayscale image is obtained by firstly projecting from the 3D space onto a 2D and then to 1D.

The points $\boldsymbol{\rho} = (\rho_1, \rho_2, \rho_3)$ aren't independent [6]. In fact, for each pixel, the calculated geometric mean log chromaticity is placed on the plane $\boldsymbol{u}^{\perp}$ depicted in figure 4.

This plane is constrained by equation

$$\begin{aligned}\rho_1 + \rho_2 + \rho_3 &= \ln\left(\frac{R_1}{\sqrt[3]{R_1 R_2 R_3}}\right) + \ln\left(\frac{R_2}{\sqrt[3]{R_1 R_2 R_3}}\right) + \ln\left(\frac{R_3}{\sqrt[3]{R_1 R_2 R_3}}\right) \\ &= \ln\left(\frac{R_1 R_2 R_3}{R_1 R_2 R_3}\right) = \ln(1) = 0,\end{aligned} \tag{16}$$

and is perpendicular to the vector $\boldsymbol{u} = \frac{1}{\sqrt{3}}(1,1,1)^T$, since $\boldsymbol{\rho} \cdot \boldsymbol{u} = 0$.

To characterize the 2D space, the projector $P_u^\perp$ onto this plane is considered [6]. The projection of $\boldsymbol{\rho}$ onto a vector $\boldsymbol{u}$ is defined by a projection matrix $P_u = \frac{\boldsymbol{u}\boldsymbol{u}^T}{\boldsymbol{u}^T\boldsymbol{u}} = \boldsymbol{u}\boldsymbol{u}^T$, according to [20]. The projection of $\boldsymbol{\rho}$ onto a space orthogonal to $\boldsymbol{u}$ is $P_u^\perp = I - P_u = I - \boldsymbol{u}\boldsymbol{u}^T$, where I is the identity matrix. On the other hand, the projection matrix can be decomposed as $P_u^\perp = U^T U$, where U is the 2×3 matrix with orthonormal rows. Defined this way, matrix U is used to transform points $\boldsymbol{\rho}$ into the coordinate system $\boldsymbol{\chi}$ on the plane $\boldsymbol{u}^\perp$:

$$\boldsymbol{\chi} = (\chi_1, \chi_2)^T = U\boldsymbol{\rho}. \tag{17}$$

The matrix U can be composed in any way so that $U^T U = I - \boldsymbol{u}\boldsymbol{u}^T$ and the conditions above hold true. The value proposed in [6] is used in this thesis:

$$U = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{6}} & -\frac{2}{\sqrt{6}} \end{bmatrix}. \tag{18}$$

Deriving from the equations (14) and (17), individual points in the discussed 2D space $\boldsymbol{\chi}$ have coordinates

$$\begin{aligned}\chi_1 &= \frac{1}{\sqrt{2}}\left(\ln\left(\frac{s_1}{s_m}\right) + \frac{e_1 - e_m}{T}\right) - \frac{1}{\sqrt{2}}\left(\ln\left(\frac{s_2}{s_m}\right) + \frac{e_2 - e_m}{T}\right) \\ &= \frac{1}{\sqrt{2}}\ln\left(\frac{s_1}{s_2}\right) + \frac{1}{\sqrt{2}}\frac{e_1 - e_2}{T} \\ \chi_2 &= \frac{1}{\sqrt{6}}\ln\left(\frac{s_1 s_2}{s_3^2}\right) + \frac{1}{\sqrt{6}}\frac{e_1 + e_2 - 2e_3}{T}.\end{aligned} \tag{19}$$

The equations above have similar properties to the equation (10). Similarly to the equation (11) points $\chi = (\chi_1, \chi_2)$ form parallel lines with slope dependent only on the camera parameters and offsets dependent on the surfaces:

$$\chi_2 = \frac{1}{\sqrt{2}}\ln\left(\frac{s_1 s_2}{s_3}\right) - \frac{1}{\sqrt{6}}\frac{e_1 + e_2 - 2e_3}{e_1 - e_2}\left(\frac{s_1}{s_2}\right) + \frac{1}{\sqrt{3}}\frac{e_1 + e_2 - 2e_3}{e_1 - e_2}\chi_1. \tag{20}$$

Obtaining of the grayscale image is similar to the situation in the band-ratio log chromaticity space, shown in the figure 3. The projection of points (χ_1, χ_2) onto the 1D space is calculated using the equation (15), where $\rho = \chi$.

2.4 Obtaining the illuminant-invariant angle

Knowing the correct illuminant-invariant angle θ is essential to the creation of the shadow free image. If θ isn't correct, then the situation depicted in figure 5 can occur. In figure 5(a) θ is correctly estimated. Here, shading, illumination and luminant temperature are removed, since the points forming one line have similar value in the 1D space. In figure 5(b) the illuminant-invariant direction isn't correctly found, which

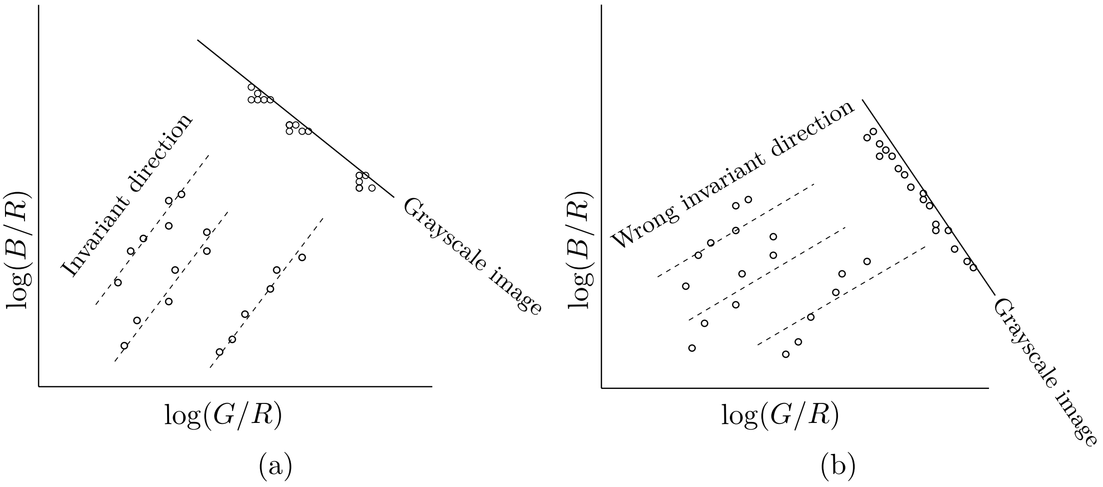


Figure 5 Intuition for finding best direction via minimizing the entropy. **(a)**: Log-ratio feature space values for paint patches fall along parallel lines, as lighting is changed. Each patch corresponds to a single probability peak when projected in the direction orthogonal to the direction of lighting change. **(b)**: Projecting in the wrong direction leads to a 1D pdf which is less peaked, and hence of larger entropy.

From International Journal of Computer Vision, "Entropy Minimization for Shadow Removal", Volume 85, Issue 1, p 37, by Graham D. Finlayson, Mark S. Drew, Cheng Lu. ©Springer Science+Business Media, LLC 2009. Reprinted with permission of Springer.

causes, that points with different illumination are projected onto different values in 1D space, causing shadows to appear in the grayscale image.

Several methods to estimate θ are used in the literature. These methods are mostly based on the two approaches—camera calibration and entropy minimization.

The camera calibration method is described in [6]. It's based on the idea, that since the illuminant-invariant angle is dependent only on the camera parameters, it can be estimated off-line. This method requires to perform a calibration using the camera, can't be used on the single image made by unknown camera and isn't robust to changes.

Therefore methods based on the entropy minimization were proposed. These methods' main idea is that the grayscale image calculated with the correct value θ will have smaller entropy, than images calculated with wrong value of θ. The best estimation of θ is found by comparing grayscale images generated with different angles. In [6] two methods—entropy minimization and information potential maximization—were described. The authors of [8] use entropy minimization, but differ in the detection of outliers. Authors of [21] also use entropy minimization, but adopt different to [6] and [8] rule for histogram bin width calculation. Authors of [7] use entropy minimization, but propose another way to determine outliers and claim to find more stable results of θ by minimizing the average entropy distribution of a set of input images instead of using the single image.

2.4.1 Entropy Minimization

The main principle of entropy minimization is described in algorithm 1. The illuminant invariant direction is found by comparing entropies of grayscale images corresponding to all possible angles. Firstly, the 2D log-chromaticity space is computed for the given image. Then, for each angle $\theta = \{1, \dots, 180\}$, the 1D illuminant invariant image $\mathcal{I}$ is computed. After careful outlier detection the entropy is calculated. The angle corresponding to the minimal entropy is chosen as the correct illuminant-invariant angle.

Algorithm 1: The main principle of the entropy minimization

```
Input  : RGB image
Output: Grayscale image corresponding to $\theta$ with minimal entropy
Create 2D log-chromaticity image $\chi$
$minEntropy = \infty$
$minTheta = 0$
for $\theta = 1 \ldots 180$ do
    Compute illuminant invariant image $\mathcal{I}$ from $\chi$ with angle $\theta$
    Choose inliers from $\mathcal{I}$
    Calculate the appropriate histogram bin width
    Create histogram from inliers
    Calculate entropy $entropy$ from histogram
    if $entropy < minEntropy$ then
        $minEntropy = entropy$
        $minTheta = \theta$
Compute illuminant invariant image $\mathcal{I}$ from $\chi$ with angle $minTheta$
```

In entropy minimization algorithm the comparison of different grayscale images is done with Shannon entropy

$$\eta = \sum_{i=1}^{n} -p_i \operatorname{lb} p_i, \tag{21}$$

where n is the number of histogram bins, p_i is the probability of the bin and lb denotes the binary logarithm. However, such definition of entropy is sensitive to the choice of the histogram bin width. If the bin width is too small, then algorithm is prune to noise. On the contrary, if the bin width is too big, then the precision is lost. The bad choice can lead images with different values and standard deviations to not being treated equally and the wrong image $\mathcal{I}_\theta$ to be favored. Therefore, methods, which adapt bin width to the individual image, are used in many studies.

In the illuminant invariance theory the most used formula to compute the appropriate bin width h is Scott's normal reference rule for normally distributed data:

$$h = \frac{3.5\sigma}{N^{1/3}}, \tag{22}$$

where N is the number of inliers and σ is their standard deviation. Implementation of this rule needs careful detection of outliers. I implement this rule in my algorithm.

The other possibility is to use Freedman and Diaconis' bin width, as proposed in [21]:

$$h = \frac{2\,\mathrm{IQR}}{N^{1/3}}, \tag{23}$$

where N is the number of samples and IQR is the interquartile range. The above equation can be more robust, than Scott's reference rule, as IQR is less sensitive to the outliers.

2.4.2 Outlier detection

Scott's normal reference rule for bin width calculation assumes normally distributed data, which makes it sensitive to outliers. Often outliers are pixels with very small or large value of R, G and/or B. Such pixels correspond to infinite or very big numbers in log chromaticity spaces. If not deleted from computations these pixels contribute to calculations of mean and standard deviation. The resulting mean may shift from the position dictated by inliers. The standard deviation may be bigger, making histogram bin width to lose precision.

In literature different methods of outlier detection are presented. Finlayson, Drew and Lu in [6] implement a simple rule to differentiate between inliers and outliers. They propose to use only middle 90% of all pixels to create a histogram. This method assumes that outliers are presented only in 5% of the lowest and 5% of the biggest values. The problem with this method is that if the number of actual outliers is much bigger than 10% it can still be prone to errors.

Another approach is used by Krajník, Blažíček and Santos in [8]. Authors also use a simple rule: values i which are lower than one threshold i_{min} or bigger than the other i_{max} are called outliers. Authors use values $i_{min} = 0.05$ and $i_{max} = 0.95$ for normalized image. Note, that unlike the previous method the number of inliers is variable between different histograms. This method solves the issue of the big number of outliers. Although, it may have troubles adapting to histograms, where inliers are shifted to one of the sides. It may not be the problem, if the shift is expected to be the same in all of the images. Unfortunately, in my application it's not the case—inliers change their values from dark to light depending on the invariant angle.

Alternatively, in [7] Álvarez and López proposed a method, which addresses both of the issues. They utilize the algorithm capable of adapting to the unique range of values of each image. The algorithm is presented in detail in [22]. It's based on the Chebyshev's inequality in a form:

$$P\left(|X - \mu| \geq k\sigma\right) \leq \left(\frac{1}{k^2}\right), \tag{24}$$

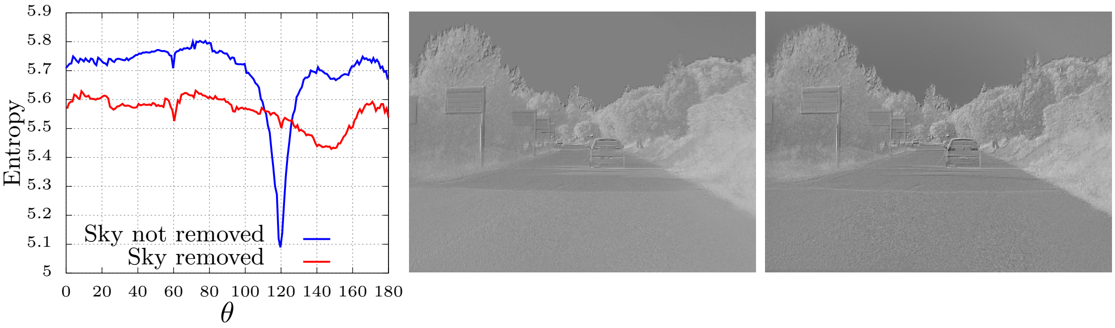
where X represents the random variable, μ is the expected or mean value, σ is the standard deviation and k is the number of standard deviations from the mean. The equation shows that less than $\frac{1}{k^2}$ of data lies further than k standard deviations from the mean μ. Chebyshev's inequality assumes that data distribution isn't known, so authors propose two methods—to work with unknown distribution and an extension of the algorithm to work with unimodal data. In my algorithm I don't use the unimodal assumption, since pixel of grayscale image may gain various values depending on the original color $\mathbf{R}$ and where this color is projected onto the 1D space by invariant angle θ. In outdoors applications colors $\mathbf{R}$ presented in the image are unpredictable. If the distribution happens to be unimodal, algorithm will still work, but the results wouldn't be as precise as if dedicated algorithm was used.

The goal of the algorithm is to find an Outlier Detection Value ODV. It distinguishes between the upper value ODV_U and lover value ODV_L, bounding the distribution. Every value i within the boundaries $ODV_L \leq i \leq ODV_U$ is an inlier, everything else is an outlier.

The calculation of ODV is divided into two stages. The first stage roughly estimates ODV and determines which values aren't outliers and should be used in calculations in next step. In the second stage these values are used to more accurately calculate final ODV.

The first stage is divided into several steps:

1. Decide the probability p_1 of pixel being a potential outlier. It is a rough estimation and authors recommend it to be bigger than the expected probability of pixel being an outlier. Authors propose values similar to $p_1 = \{0.1, 0.05, 0.01\}$ in this stage.



(a) (b) (c)

Figure 6 **(a)** The difference between entropy plot calculated from image with sky and from image with sky removed. **(b)**, **(c)** The grayscale images corresponding to minimal entropy if sky wasn't removed (b) and if sky was removed (c).

2. Calculate k from equation (24):

$$k = \frac{1}{\sqrt{p_1}} \tag{25}$$

3. Use the histogram $\mathcal{H}_\theta$ to calculate μ and σ.
4. Calculate ODV from equation (24):

$$\begin{aligned} ODV_{1U} =& \quad \mu + k\sigma \\ ODV_{1L} =& \quad \mu - k\sigma \end{aligned} \tag{26}$$

5. Create $\mathcal{H}_{\theta 2}$ from values in $\mathcal{H}_\theta$ within bounds of ODV_{1U} and ODV_{1L}

The second stage is similar to the first stage. It repeats operations from the first stage, but computations are made with other values:

1. Decide the expected probability p_2 of pixel being an outlier. Authors recommend values like $p_2 = \{0.01, 0.001, 0.0001\}$ in this stage.
2. Calculate k from equation (24):

$$k = \frac{1}{\sqrt{p_2}} \tag{27}$$

3. Use the histogram $\mathcal{H}_{\theta 2}$ to calculate μ and σ.
4. Calculate ODV from equation (24):

$$\begin{aligned} ODV_U &= \mu + k\sigma \\ ODV_L &= \mu - k\sigma \end{aligned} \tag{28}$$

Values ODV_U and ODV_L are the final result of the algorithm. They are used on the original histogram $\mathcal{H}_\theta$ for distinguishing between inliers and outliers.

2.5 Sky detection

Experiments have shown, that though entropy minimization algorithms are generally good in finding the right invariant angle, they may (often) fail in images with large sky area. The example is shown in figure 6 (b), where invariant angle is incorrectly computed resulting in shadows still be visible in grayscale image. According to [2]

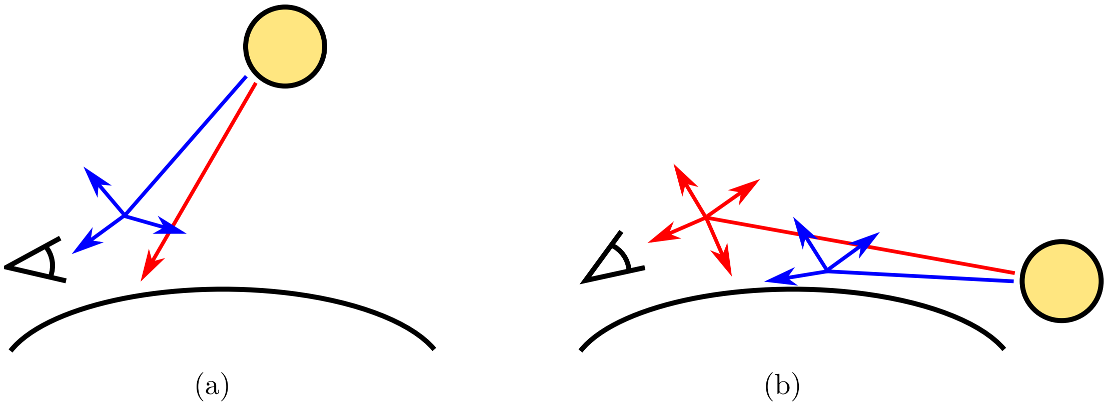


Figure 7 The principle of Rayleigh scattering **(a)** at daylight and **(b)** at sunset.

the reason for these errors can be Rayleigh scattering. This phenomenon is a scattering of electromagnetic radiation by particles with dimensions smaller than radiation wavelength [23]. The light is scattered by Earth atmosphere according to formula:

$$I\left(\lambda\right)_{sc} \propto \frac{I\left(\lambda\right)_{inc}}{\lambda^4}, \tag{29}$$

where $I\left(\lambda\right)_{sc}$ is the intensity of the scattered light, $I\left(\lambda\right)_{inc}$ is the intensity of the incident light and λ is the light wavelength.

According to this formula, the scattering of the light is dependent on the wavelength. Colors with smaller wavelengths (e.g. blue) are scattered more strongly than ones with bigger wavelengths (e.g. red). Due to Rayleigh scattering individual colors in the white light traveling through the atmosphere are not diffused equally. The intuition for this phenomenon is shown in figure 7. During daylight the blue color is scattered in the atmosphere, while red color travels further, as is shown in figure 7(a). The scattered blue light reaches the observer making the sky appear blue. At sunset the distance from the sun to the observer is bigger. The blue color is (almost entirely) scattered away before light reaches the observer, as is shown in figure 7(b). The remaining red light is scattered in the atmosphere from where it reaches the observer making sky appear redder.

Rayleigh scattering also causes the color hue across the sky area, which creates the obstacle for entropy minimization algorithms. In figure 6(a) the minimal entropy corresponds to the grayscale image (b) with the lowest sky color hue. Contrary, if the sky area was found and removed from computations, the invariant angle was found correctly, as in figure 6(c).

In this thesis an algorithm [24] based on Otsu's thresholding method [25] is used for horizon and sky detection.

2.5.1 Otsu's threshold method

The Otsu's Global thresholding method for gray-level histograms is based on the comparison of how each gray level performs as a threshold. The algorithm successively chooses each gray-level to divide the histogram into two classes: background and foreground. The gray-level minimizing the intra-class variance is then chosen as an optimal threshold. Since sky is generally brighter than the rest of an image, this segmentation method may be safely used to its identification.

Assume a grayscale image with N intensity levels. Otsu's threshold method uses its normalized gray-level histogram. In such histogram each bin represents the probability

p_i of the corresponding gray-level, $i = \{0, 1, \ldots, N-1\}$. Each possible threshold T divides the histogram into two classes $C_1 = \{0, 1, \ldots, T\}$ and $C_2 = \{T+1, T+2, \ldots, N-1\}$.

The class probabilities are

$$q_1(T) = \sum_{i=0}^{T} p_i \quad \text{and} \quad q_2(T) = \sum_{i=T+1}^{N-1} p_i. \tag{30}$$

The class means are calculated as

$$\begin{aligned} \mu_1(T) &= \sum_{i=0}^{T} iP(i|C_1) = \sum_{i=0}^{T} i\frac{p_i}{q_1(T)} \quad \text{and} \\ \mu_2(T) &= \sum_{i=T+1}^{N-1} iP(i|C_2) = \sum_{i=T+1}^{N-1} i\frac{p_i}{q_2(T)}, \end{aligned} \tag{31}$$

where $P(i|C_j)$ is the conditional probability of gray-level i given class C_j.

The class variance is defined as

$$\begin{aligned} \sigma_1^2(T) &= \sum_{i=0}^{T} [i-\mu_1]^2 P(i|C_1) = \sum_{i=0}^{T} [i-\mu_1]^2 \frac{p_i}{q_1(T)} \quad \text{and} \\ \sigma_2^2(T) &= \sum_{i=T+1}^{N-1} [i-\mu_2]^2 P(i|C_2) = \sum_{i=T+1}^{N-1} [i-\mu_2]^2 \frac{p_i}{q_2(T)}. \end{aligned} \tag{32}$$

The optimal threshold T_o is found according to [24] by minimizing the intra-class (within-class) variance:

$$\sigma_W^2(T_o) = \min_{0 \le T < N} \sigma_W^2(T), \tag{33}$$

where

$$\sigma_W^2(T) = q_1(T)\sigma_1^2(T) + q_2(T)\sigma_2^2(T). \tag{34}$$

Performing all of these operations for every possible threshold T would be ineffective, therefore I implemented the method using algorithm 2. Here, the number of computations is minimized by taking variables out of cycle in order to get rid of inner sums. In the following calculations parameter T is omitted even though variables are still dependent on it. Assume, $k = 1, 2$ stands for each class. Variables q_k, μ_k and σ_k aren't calculated from the beginning in every iteration for different T. Instead, new variables μ'_k and σ'_k are introduced. These variables, as well as q_k, are set to zero for $k = 1$ and to a maximum value for $k = 2$. During iterations through gray-levels, variables with $k = 1$ are increased and variables with $k = 2$ are decreased according to T. The intra-class variance σ_W^2 calculation is changed with respect to new variables.

My calculations start on equations (34) and (32). Since $q_k(T)$ is canceled out, new variable $\sigma''_2 = q_k(T)\sigma_k^2(T)$ is introduced so that $\sigma_W^2 = \sigma''_1 + \sigma''_2$. It's equal to

$$\sigma''_k = \sum_{i \in C_k} [i-\mu_k]^2 p_i = \sum_{i \in C_k} \left[i^2 p_i - 2i\mu_k p_i + \mu_k^2 p_i\right]. \tag{35}$$

Some parameters in this equation aren't dependent on i, therefore it can be further modified:

$$\sigma''_k = \sum_{i \in C_k} \left[i^2 p_i\right] - 2\mu_k \sum_{i \in C_k} [ip_i] + \mu_k^2 \sum_{i \in C_k} [p_i]. \tag{36}$$

Algorithm 2: The algorithm finding Otsu's threshold

```
Input  : Normalized histogram $\mathcal{H}_\theta$ with $N$ gray-levels
Output: Otsu's threshold $T_o$
$\sigma^2_W(T_o) = \infty$
$T_o = 0$
$q_1 = 0.0$, $q_2 = 1.0$
$\mu'_1 = \mu'_2 = 0.0$
$\sigma'_1 = \sigma'_2 = 0.0$
for $i = 0 \dots N-1$ do
    $\mu'_2 = \mu'_2 + i \cdot p_i$
    $\sigma'_2 = \sigma'_2 + i \cdot i \cdot p_i$
for $i = 0 \dots N-1$ do
    $q_1 = q_1 + p_i$
    $q_2 = q_2 - p_i$
    $\mu'_1 = \mu'_1 + i \cdot p_i$
    $\mu'_2 = \mu'_2 - i \cdot p_i$
    $\sigma'_1 = \sigma'_1 + i \cdot i \cdot p_i$
    $\sigma'_2 = \sigma'_2 - i \cdot i \cdot p_i$
    $\sigma''_1 = \sigma'_1 - \frac{(\mu'_1)^2}{q_1}$
    $\sigma''_2 = \sigma'_1 - \frac{(\mu'_2)^2}{q_2}$
    $\sigma^2_W = \sigma''_1 + \sigma''_2$
    if $\sigma^2_W < \sigma^2_W(T_o)$ then
        $\sigma^2_W(T_o) = \sigma^2_W$
        $T_o = i$
```

According to equation (30), $\sum_{i \in C_k} [p_i] = q_k$. In order to get rid of the two remaining sums, I define

$$\mu'_k = \sum_{i \in C_k} [ip_i] \quad \text{and} \tag{37}$$

$$\sigma'_k = \sum_{i \in C_k} \left[i^2 p_i \right]. \tag{38}$$

Equation (31) can be then rewritten as $\mu_k = \frac{\mu'_k}{q_k}$. Therefore, equation (36), can be further modified:

$$\sigma''_k = \sigma'_k - 2\frac{\mu'_k}{q_k}\mu'_k + \left(\frac{\mu'_k}{q_k}\right)^2 q_k = \sigma'_k - \frac{(\mu'_k)^2}{q_k}. \tag{39}$$

All of the described above calculations were made to get rid of the sums inside of iteration through gray-levels. All of the variables present in equation (39) can be declared outside of the cycle and modified at the beginning of every iteration. Number of gray-levels is increasing by one in C_1 and is decreasing by one in C_2 in the iteration i. Therefore, variables are changed accordingly by their increment: $q_1 = q_1 + p_i$, $q_2 = q_2 - p_i$, $\mu'_1 = \mu'_1 + i \cdot p_i$, $\mu'_2 = \mu'_2 - i \cdot p_i$, $\sigma'_1 = \sigma'_1 + i^2 \cdot p_i$ and $\sigma'_2 = \sigma'_2 - i^2 \cdot p_i$.

2.5.2 Horizon detection algorithm

Otsu's method is one of the most used segmentation techniques in computer vision [24]. One of its advantages is its universality—it can be used in histograms with any shapes: unimodal, bimodal and multimodal. Because of this feature it is a good candidate to be the main segmentation method for sky detection. Its weaknesses are dependency on the size of an object in the image and sensitivity to noise, as was demonstrated in [26]. To overcome these obstacles and to speed up the algorithm, certain preprocessing is used.

The noise reduction in the image is often performed by smoothing. In this thesis I use Gaussian blur, which is achieved by convolving the image with a Gaussian function.

It's safe to assume, that sky area in images intended for road detection is in an upper part of the image. Therefore, only approximately upper 50% of the image were considered in the implemented algorithm.

Otsu's threshold method was designed to work on gray-level histograms, therefore transformation is required in color images. In this thesis the grayscale image is formed by the most dominant color channel, according to formula [24]:

$$C_{ch} = \underset{\{R,G,B\}}{\arg\max}(N_R, N_G, N_B), \tag{40}$$

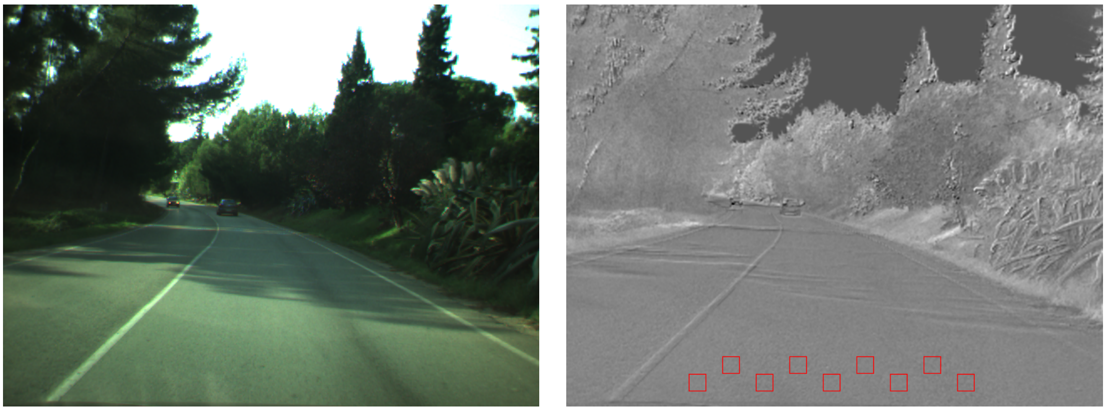
where C_{ch} is the chosen color channel and N_i, $i = R, G, B$ means the number of pixels in which the corresponding channel was dominant.

In the implemented algorithm I use Otsu's global threshold method on upper half of an image to find the optimal threshold T separating sky and non-sky areas. Pixel is classified as sky if its value is bigger than threshold and it's located in upper 5% of an image. Such pixels are called seeds. Alternatively, pixel is classified as sky if its value is bigger than threshold and it can be reached from seeds by continuous curve of sky pixels.

Depending on camera position and orientation sky area can have various sizes in different images. Its size can be much smaller in comparison to ground, but also much bigger, especially after cropping image to upper 50%. According to [24], Otsu's segmentation algorithm works better when sizes of both background and foreground are similar. It's stated in [26], that error rate increases rapidly if object size is smaller than approximately 30%. To address this issue authors of [24] use the modification of global Otsu's threshold method. They divide the potential sky area into 10 parts $i = \{1, \ldots, 10\}$. The optimal threshold T_i is calculated for each part using Otsu's method. The percentage of foreground pixels is then calculated in every image i for each threshold T_i, and the sum $\sum_i$ of percentages is computed for every image i. The horizon is obtained by finding the biggest percentage difference between sums $\sum_{i-1}$ and $\sum_i$ of two consecutive parts $i-1$ and i. This estimation of horizon is combined with the Hough transform, and the final result is based on the weighted average of both algorithms.

In my experiments, the Otsu's global threshold method has shown good results in sky detection even without usage of the described modification. In rural areas with forest cover the bright sky was accurately determined even in images where it occupied only approximately 0.33% of potential sky area (at the time upper 60% of an image).

Urban environment was a greater challenge for the algorithm. Some light colored as well as glass covered buildings were occasionally classified as sky in addition to actual sky area. In my opinion, despite the fact that results aren't entirely accurate, such behavior may be rather beneficial. Light colored buildings would have been likely



(a) (b)

Figure 8 **(a)** An example of the image with road. **(b)** An example of the shadow free image.

deleted with outliers in succeeding algorithms. Therefore the stage at which the building is deleted makes a little difference for outlier detectors with variable threshold, but can be crucial for algorithms based on fixed parameters. Glass buildings can be dangerous for entropy minimization algorithms, as these aren't Lambertian surfaces and they mostly reflect non desirable sky. If glass buildings are detected and deleted on this stage, they won't create an additional disruption for entropy minimization algorithms.

The other problem typical but not exclusive to urban environments appears, when mobile robot is turned towards a tall object nearby, such as building or dense forest. In such composition, image visible by robot can entirely lack sky area. The proposed sky detection algorithm would still classify the most bright part of an image as sky. This issue wasn't addressed in the thesis, since this situation isn't very common. Moreover, in some cases it may be beneficial for above mentioned reasons. In worst case scenario, algorithm would classify as sky less than 50% of image pixels, all located in upper half. This way the most critical part of the road can never be wrongly classified as sky and road detection can still be safely performed.

2.6 Road detection on illuminant invariant image

The shadow-free image obtained using the theory of illuminant invariance is grayscale. An example of such image is shown in figure 8(b) next to the original colorful image in figure 8(a). It can be seen in figure 8(b), that thanks to the illuminant invariance the road region has pixels of the similar values in both shadowed and illuminated areas. This is the property on which this type of road detection algorithms is based.

Since the whole road is of the same color it's expected of its values histogram to be unimodal with low dispersion and skewness [7]. Under the assumption, that the bottom part of the image is always road, it can be used to build the road model. In this thesis I use 9 regions with the size of 21 × 21 pixels, placed as depicted in figure 8(b). From these pixels' values $\mathcal{I}_{road}$ I create a normalized histogram $\mathcal{H}\left(\mathcal{I}_{road}\right)$ and use it to classify the remaining pixels in the image.

Authors of [7] take the normalized histogram $\mathcal{H}\left(\mathcal{I}_{road}\right)$ as a probability distribution $\mathrm{p}\left(\mathcal{I}\left(x\right)|road\right)$. For each unclassified pixel value $\mathcal{I}(x)$ they determine its probability of being road and compare it to the fixed threshold λ. Their classification rule looks like

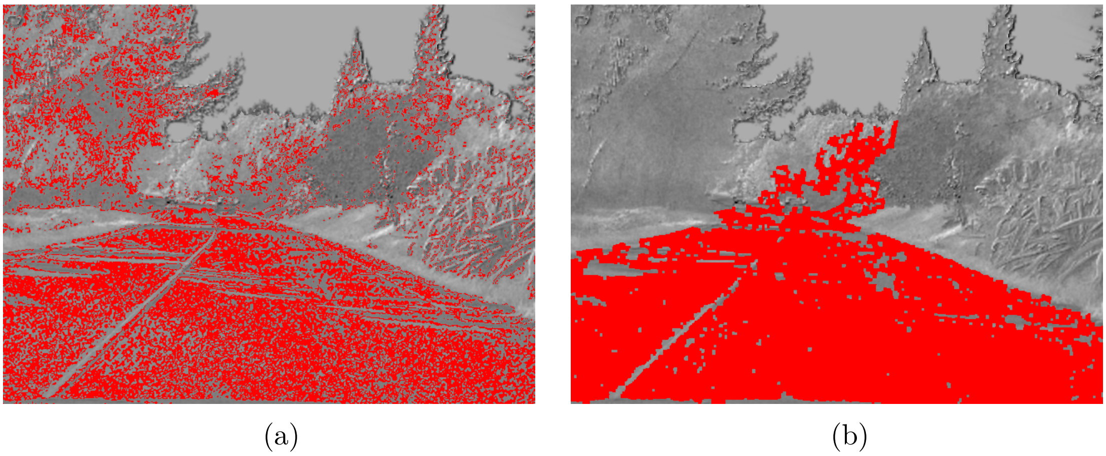


Figure 9 **(a)** An example of the classification rule result performed on figure 8(b). **(b)** An example of the combination of morphological operations used on (a).

following:

$$\begin{cases} x \text{ is road,} & \text{if } \mathrm{p}\left(\mathcal{I}\left(x\right)|road\right) \geq \lambda \\ x \text{ is non road,} & \text{otherwise.} \end{cases} \tag{41}$$

The parameter λ is set using the training dataset, the optimal value is the one which maximizes the average effectiveness.

Authors of [2] use confidence intervals to find the range of values considered road. They calculate two thresholds $\lambda_1 = \mu - 1.86\frac{\sigma}{\sqrt{n}}$ and $\lambda_2 = \mu + 1.86\frac{\sigma}{\sqrt{n}}$, where n is the number of pixels in histogram. Every value $\mathcal{I}\left(x\right)$ which lies within the interval $[\lambda_1, \lambda_2]$ is considered to be road. Authors state, that this method has no need in the training dataset. However, in my experiments, this thresholds seemed to be too strict. According to my calculations, they would classify as road only values lying at maximum distance of $1.86\frac{\sigma}{\sqrt{9 \cdot 10 \cdot 10}} = 0.062\sigma$ from the mean value μ. In my case, where the number of the pixels in $\mathcal{H}\left(\mathcal{I}_{road}\right)$ is bigger, the distance would be 0.030σ.

My classification algorithm is in principle the same as in [7], but visually it resembles [2]. I calculate two thresholds

$$\begin{aligned} \lambda_1 &= \mu - k\sigma, \qquad \text{and} \\ \lambda_2 &= \mu + k\sigma \end{aligned} \tag{42}$$

where k is a parameter estimated using the training dataset. The pixel is classified as road if its value lies within the interval $[\lambda_1, \lambda_2]$, according to the classification rule:

$$\begin{cases} x \text{ is road,} & \text{if } \lambda_1 \leq \mathcal{I}\left(x\right) \leq \lambda_2 \\ x \text{ is non road,} & \text{otherwise.} \end{cases} \tag{43}$$

The proposed algorithms may result in some pixels misclassified [2], as is depicted in figure 9(a). The false-negatives classifications, i.e. road pixels classified as background, may be the result of the road color variations or noise. This type of misclassification is mostly resolved by morphological operations filling the holes between positively classified pixels.

The false-positives are the pixels from the background, which were classified as road. This can be caused by objects in the background, which have the color similar to road. The other possible reason is the illuminant invariant projection to 1D itself. In RGB image with 24-bit—8 bits for each RGB channel—color representation the amount of different values which individual pixel can have is equal to 2^{24}, while in 8-bit grayscale it's only $2^8 = 256$ different gray levels. Therefore, the projection of such RGB pixel to grayscale results in losses in color diversity. The example of such behavior can be seen in figure 2 (a) and (c), where most of green colors is projected to the same values as yellow, approximately 150–152. The false-positives which are far away from road region may be filtered out by taking only positively classified connected neighboring pixels. The starting point may be set to patches $\mathcal{I}_{road}$ from which the road model was created. The result of such algorithm with the combination of several morphological operations performed to fill most of the holes is shown in figure 9(a). Here, many false-positively classified pixels far from road were removed. However, a big part of bushes near the road in the center of the image wasn't removed. Due to this limitation the algorithm works best in terrains, where road and non-road regions have distinguishable colors. As an example, parks with different types of road coverage and grass in background, or asphalt roads in rural areas. The solution to this problem may be the combination of this classifier with other methods. Authors of [2] use stereo vision to address this issue. I propose the combination with color model based classifier described in the next chapter.

2.7 Discussion

In this chapter the road classifier based on the illuminant invariance was presented. The method uses the physical theory of the RGB image formation in order to create a shadow free image.

The illuminant invariance image is obtained from an RGB image by firstly transforming it into the special 2D chromaticity space, where pixels from the same surfaces form straight parallel lines independently on the illuminant. The projection of these values onto the perpendicular line results in the illuminant invariant space. Since, the invariant direction is expected to have the smallest diversity of gray levels, entropy minimization algorithms are used to find the angle characterizing this direction. The shadow free image is grayscale. Different surfaces in this image, e.g. road, grass, buildings, gain different gray levels, although variations in illumination on these surfaces are gone. The road is therefore expected to be of a uniform gray level. The road detection is performed by classifying pixels according to the road model created from patches in the bottom part of the image, considered to be a safe road area.

In ideal conditions entropy minimization works well for the invariant angle estimation. The incorrect angle calculation may occur mostly in images with big illuminated/shadowed area ratio. In such images shadows can be not present, can be very small in comparison to the illuminated area or occupy most of the image. In these cases shadow area isn't sufficient to make a reasonable difference in entropy. The algorithm may find incorrect invariant angle minimizing another image quality. Such errors bring the risk, that the road area and background may come out to be of similar gray levels, resulting in big amount of false-positives. Moreover, if the shadow is small enough to be ignored by entropy minimization, but big enough to cover the critical part of the road, false-negatives may occur. If such shadow is located where the road model is created, the resulting road model may not fit the rest of the road. If it is a long line across the road, the algorithm taking continuous road may not pass it and the road above the shadow would be misclassified as a background. To address this issue, authors of [7] consider the results of few consecutive frames to find the most appropriate value. I, to speed up the algorithm, performed entropy minimization computation of the invariant angle offline on the training dataset containing only images with appropriate-sized shadows. The mean value of the invariant angles was then set as a fixed value in other experiments.

Chapter 3

Gaussian Mixture Models

Alternatively to illuminant-invariant road detectors, some authors base their approach on creating color models of the road and then using this information in the road detection. Authors of [3] and [4] propose to use Gaussian Mixture Models (GMMs) for this purpose. In the scene shown in figure 10 it's obvious to a human, that the road has gray color. But in computer vision this gray color is composed of a big number of different but at the same time similar RGB triples $\mathbf{R}$. The human-perceived gray color can be learned by a computer as a three dimensional multivariate Gaussian distribution. Normally, the road may consist of different colors, as well as have shadows represented by other colors. Each of them can be defined by a different Gaussian distribution. Therefore, the Gaussian Mixture Model is introduced as a weighted combination of several Gaussian distributions.

3.1 The creation of a GMM

The road detection method based on Gaussian Mixture Models works with offline learned models. For this purpose, a training dataset—some number of sample images—is need to be collected and prepared in advance. In these images, road and non-road regions should be manually selected. Pixels from the training dataset are used to create GMMs. In some literature only GMM for road regions is created. I, same as Zhou et al. in [4], create two GMMs: GMM_0 for road and GMM_1 for non-road, with K_0 and K_1 components respectively.

The RGB color of a pixel is a random vector $\boldsymbol{x} = \mathbf{R} = [R_1, R_2, R_3]^{\mathrm{T}}$. The multivariate Gaussian (normal) distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ of this vector can be defined by two of its parameters: mean value $\boldsymbol{\mu}$ and 3×3 covariance matrix $\boldsymbol{\Sigma} = \mathrm{E}\left[(\mathbf{X} - \boldsymbol{\mu})(\mathbf{X} - \boldsymbol{\mu})^{\mathrm{T}}\right]$.

The model of the road or non-road region is represented by a GMM with K triples $\mathrm{GMM} = \{(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, w_1), (\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2, w_2), \ldots, (\boldsymbol{\mu}_K, \boldsymbol{\Sigma}_K, w_K)\}$, where w_i is the weight of the Gaussian component i. The weight is calculated as a ratio of pixel count N_i in the component i to a pixel count in the model GMM as $w_i = N_i / \sum_{j \in \{1,\ldots,K\}} N_j$.

The training pixels are divided into different GMM components by data clustering algorithm. In this thesis I implemented Orchard and Bouman binary splitting algorithm [27], which is described in algorithm 3. The basic principle of the method lies in dividing of the cluster with the biggest variance into two, until the desired number of clusters is reached. Authors claim, that the direction of the biggest variation is determined by the eigenvector $\boldsymbol{v}_k$ corresponding to the largest eigenvalue λ_k of cluster's k covariance matrix $\boldsymbol{\Sigma}_k$. The decision about which cluster to split next is made by comparison of λ_k, $k = \{1, \ldots, K\}$. The cluster C_m with the largest eigenvalue λ_m is chosen as a best current candidate. The split is made along the plane perpendicular to the direction $\boldsymbol{v}_m$ of the biggest variation and passes through the mean value $\boldsymbol{\mu}_m$ of the cluster.

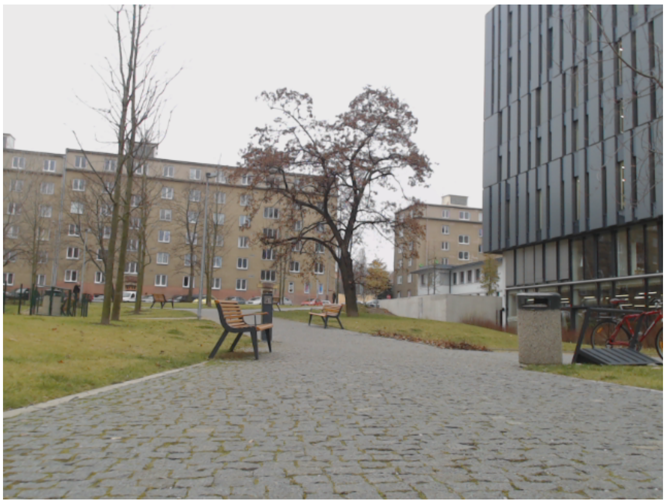


Figure 10 An example of the image with road.

Algorithm 3: The creation of a GMM from the set of pixels

```
Input  : The number of componets $K$, set of pixels $Px$ with $Nx$ pixels
Output: GMM with $K$ components
Create the first cluster from all the pixels: $C_m = Px$, $m = 1$
Calculate $\boldsymbol{\mu}_m$, $\boldsymbol{\Sigma}_m$
$N_m$ =: number of pixels in the component
$\lambda_m$ =: the largest eigenvalue of $\boldsymbol{\Sigma}_m$
$\boldsymbol{v}_m$ =: the eigenvector corresponding to $\lambda_m$
Create GMM component $G_m = (\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m, N_m/Nx)$
for $k = 2, \ldots, K$ do
    for $x \in C_m$ do
        if $\boldsymbol{v}_m x \leq \boldsymbol{v}_m \boldsymbol{\mu}_m$ then
            Add $x$ to $C_k$
            Remove $x$ from $C_m$
    Calculate new $\boldsymbol{\mu}_m$, $\boldsymbol{\Sigma}_m$, $N_m$, $\lambda_m$, $\boldsymbol{v}_m$
    Calculate $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$, $N_k$, $\lambda_k$, $\boldsymbol{v}_k$
    Update $G_m = (\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m, N_m/Nx)$
    Create $G_k = (\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, N_k/Nx)$
    $m = \arg\max_{\{1,\ldots,k\}} \{\lambda_1, \ldots, \lambda_k\}$
Return $GMM = \{G_1, \ldots, G_K\}$
```

According to [28], the probability of an individual pixel value $\boldsymbol{x}$ fitting the model described by a 3 dimensional multivariate Gaussian distribution G_k is

$$\mathcal{P}_{Gk}(\boldsymbol{x}) = \frac{1}{\left(\sqrt{2\pi}\right)^3 \sqrt{|\boldsymbol{\Sigma}_k|}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k)}. \tag{44}$$

In my algorithm I, same as Zhou et al. in [4], omit the constant $\left(\sqrt{2\pi}\right)^3$ from the equation to simplify the computations.

The probability of an individual pixel value $\boldsymbol{x}$ fitting the model GMM with K components is

$$\mathcal{P}_{GMM}(\boldsymbol{x}) = \sum_{k=1}^{K} \left[w_k \mathcal{P}_{Gk} \right]. \tag{45}$$

3.2 Road detection

It's possible to directly employ GMMs for road detection. Authors of [3] propose to compute Mahalanobis distance [28]

$$D_M(\boldsymbol{x}, k) = \sqrt{(\boldsymbol{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_k)} \tag{46}$$

between the pixel value $\boldsymbol{x}$ and the mean value $\boldsymbol{\mu}_k$ of each GMM component G_k. The minimal distance $D_M(\boldsymbol{x}, l)$ is then taken as a score of how likely $\boldsymbol{x}$ is fitting the road model. Authors use only road model GMM_0 and compute distances for the most promising components according to their weight.

Authors of [4] propose to combine GMMs representing the color information in image with its structural information. They construct a weighted graph out of both of these terms. They use a min-cut/max-flow algorithm to split the graph into two parts: road and background.

The min-cut/max-flow algorithm is based on the work of Boykov and Jolly [29]. They assume an image with a size $N \times M$ pixels. From this image they compose a graph $\mathcal{G}$ with $N \times M + 2$ nodes. While, most of the nodes correspond to pixels, the graph also contains two special nodes: source—object terminal, in my case road—and sink—background terminal. Nodes are connected with each other by weighted edges. Authors define two types of nodes: n-links and t-links. N-links are the type of edges connecting neighboring pixels with each other. T-links connect pixel nodes with terminal nodes. The weights of t-links are defined by color term, while weights of n-links are defined by a structural term.

The initial t-link costs $w_{p,t}$, where p is the pixel node and $t \in \{0, 1\}$ is a terminal node, are assigned according to a negative log-likelihood:

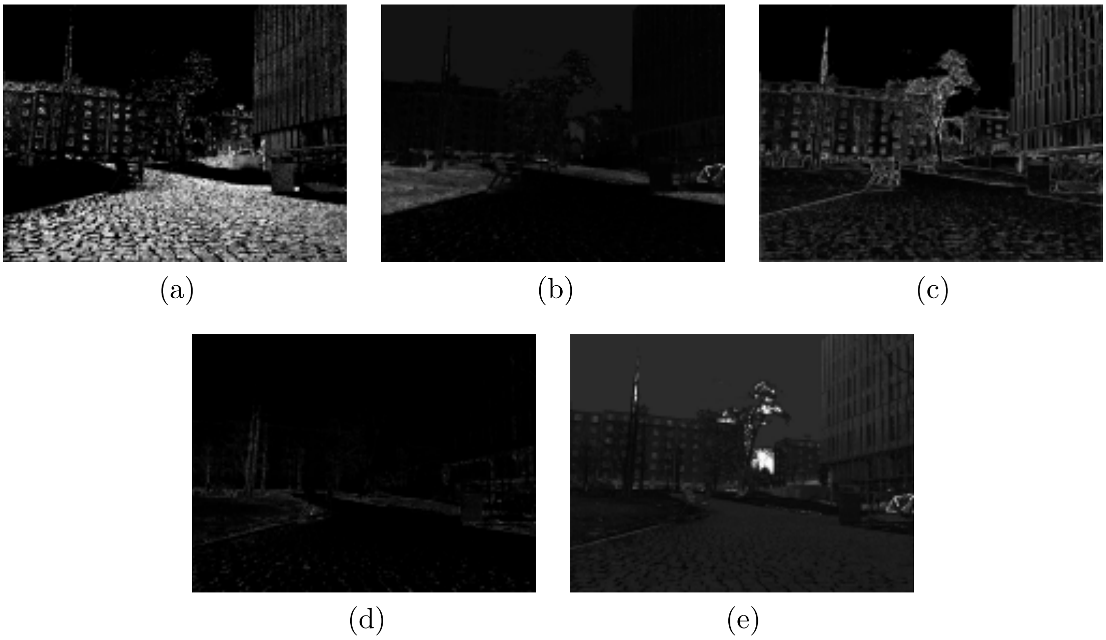
$$\begin{aligned} w_{p,0} &= -\ln \mathcal{P}_{GMM0} \\ w_{p,1} &= -\ln \mathcal{P}_{GMM1}. \end{aligned} \tag{47}$$

The probability itself wasn't suitable, as edge cost is suppose to be smaller at more desirable pixels. Since negative log-likelihood in my implementation may result in big values if $\mathcal{P}_{GMM}$ was small, I normalized the result and introduced the trade-off factor:

$$\begin{aligned} w'_{p,0} &= \xi_0 w_{p,0,\text{norm}} \\ w'_{p,1} &= \xi_1 w_{p,1,\text{norm}} + \zeta. \end{aligned} \tag{48}$$

Constants ξ_0 and ξ_1 are used to prioritize one set of data over the other. If $\xi_0 > \xi_1$, the min-cut algorithm is less likely to choose low probability values from GMM_0 as non-road. Parameter $\zeta \geq 0$ was introduced, when I discovered in my experiments, that algorithm performed better if it was present.

The example of how the probabilities are distributed in the image is shown in figure 11. The probability of pixel value $\boldsymbol{x}$ fitting the road model GMM_0 is displayed in (a) and its normalized log-likelihood in (b). Similarly, the probability of pixel value $\boldsymbol{x}$ fitting the non-road model GMM_1 is displayed in (d) and its normalized log-likelihood in (e). Greater values are represented by lighter colors, while lower values by darker colors.



(a) (b) (c)

(d) (e)

Figure 11 Probability maps of figure 10. **(a)** Probability of $\boldsymbol{x}$ matching a road model $\mathcal{P}_{GMM0}$. **(b)** Negative log-likelihood of (a) $w_{p,0,\mathrm{norm}}$. **(c)** The illustration of a structural term. **(d)** Probability of $\boldsymbol{x}$ matching a non-road model $\mathcal{P}_{GMM1}$. **(e)** Negative log-likelihood of (d) $w_{p,1,\mathrm{norm}}$.

To define n-link costs $w_{p,q}$, where p and q are neighboring pixels, Zhou et al. propose to use the following formula:

$$\begin{aligned} w_{p,q} &= e^{-\dfrac{\|\boldsymbol{x}_p - \boldsymbol{x}_q\|}{2\mathrm{E}\left[\|\boldsymbol{x}_p - \boldsymbol{x}_q\|^2\right]}}, \\ w'_{p,q} &= \gamma w_{p,q} \end{aligned} \tag{49}$$

where $\|\ldots\|$ denotes L^2 norm, $\mathrm{E}\left[\ldots\right]$ denotes the expected value over the image and γ is the trade-off between color and structure components. They also propose to use local geometric structure to adjust $w_{p,q}$ to different regions varying contrasts, but I don't use it in my work. The illustration of how the structural term is spread through the image is shown in figure 11(c). Since $w_{p,q}$ defines 8 neighbors for most of the pixels, it isn't represented in the image directly. Instead, each pixel is representing the mean value of L^2 norms $\|\boldsymbol{x}_p - \boldsymbol{x}_q\|$ between pixel value $\boldsymbol{x}_p$ and its neighbors' values $\boldsymbol{x}_q$ in the original RGB image.

The transformation of image features into graph changes road detection problem into the problem of finding the appropriate graph cut. Authors propose to use min-cut/max-flow method described in [30] in order to find the cut with the minimum cost. This algorithm is time consuming [4]. Therefore in images with large resolution downsampling is advised to speed up the computations, especially in real-time applications.

3.3 Combination of illuminant invariance theory and GMMs

The presented graph cut method for road detection has a potential to provide good results by combining color models with structure information. One of its disadvantages can be that it's strongly dependent on the quality of the training dataset. The illuminant invariance methods on the other hand don't need a training dataset to provide results. As was stated in the previous chapter, illuminant invariant algorithms may have problems with false positives, if background near the road area happen to have similar gray levels. I propose to combine both of the methods in order to verify if it brings any improvements.

In my implementation, I firstly perform road detection based on the illuminant invariance, including morphological operations and road continuity. The result is incorporated into GMM based graph cut algorithm by simply setting parameter ζ in equation (48) to a positive value in pixels classified as road. This action should increase the cost of t-link from these pixels to sink, making it less probable to them being connected.

Chapter 4

Experimental results

In my first experiments I used publicly available dataset DS1 used by Álvarez et al. in [7]. The dataset contains RGB images from the camera installed on a windshield of a car. Photos were taken at noon of a sunny day. The dataset contains images of an asphalt road in rural area. In my experiments I used only first 279 images, because the rest was corrupted by lens flare.

The second dataset DS2 I used was created by me and my thesis supervisor during the first experiment with the robot. This dataset was created as a training dataset for GMM based methods. The images containing shadows were used for invariant angle estimation. During this experiment illuminant invariant method was firstly tested and PID controller parameters were estimated. The experiment took place before noon in a small park on Evropská street in Prague. The park is surrounded by buildings, therefore I consider it an urban environment.

The second and the last experiment with the robot was intended to test all of the methods of road detection. It resulted in dataset DS3. It was conducted in the same area as DS2.

The main evaluations in this thesis were conducted by comparing results of road detection algorithms with ground truth. From both of these data true positives TP, false positives FP, true negatives TN and false negatives FN were computed. The evaluations were made using two measurements [4]: precision and error rate.

Precision

$$PR = \frac{TP}{TP + FP} \tag{50}$$

is the ratio of correctly determined road pixels to the total number of pixels classified as road. Precision reacts to the number of FP in the detected road, decreasing with bigger number of misclassified background. However, it can't differentiate a good example of road detection with big number of TP from the one with low number of TP if $FP = 0$— in both cases $PR = 1$. The range of values of this measurement is from 0 to 1, since $TP \leq TP + FP$.

Error rate

$$ER = \frac{FP + FN}{TP + FN} \tag{51}$$

gives the ratio of false detections to the total ground truth road area. This measurement solves the problem with precision's lack of sensitivity to $FP = 0$ by taking the total false detection number.

By combining both of the measurements it's possible to identify good road detection results with low error number. The results with low ER and high PR are preferred.

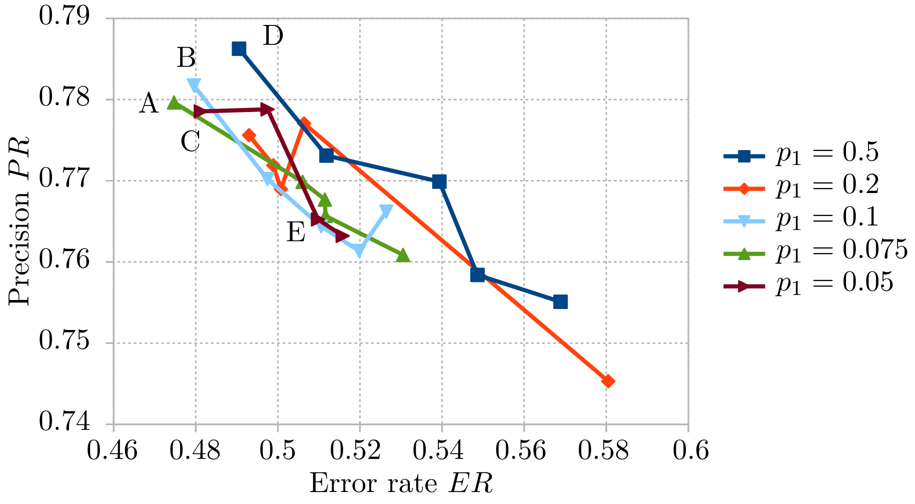


Figure 12 Road detection (based on the illuminant invariance) quality depending on parameters p_1 and p_2 in entropy minimization algorithm 1.

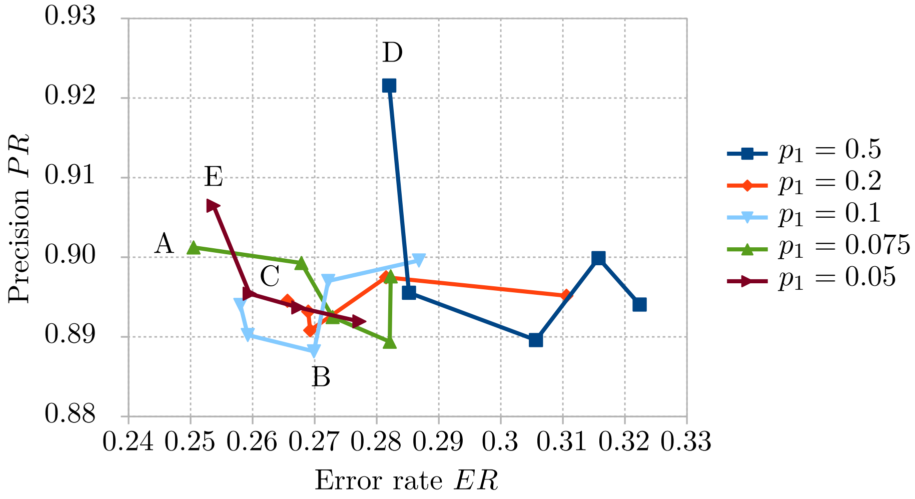


Figure 13 Continuous road detection (based on the illuminant invariance) quality depending on parameters p_1 and p_2 in entropy minimization algorithm 1.

4.1 Parameter estimation

4.1.1 Entropy minimization p_1 and p_2

Parameters p_1 and p_2 from equations (25) and (27) were estimated on the shorter version of DS1, containing 52 manually selected images with shadows. I compared the results of the road classification rule described by equation (43) computed with different combinations of 5 p_1 and 5 p_2 values. I also made a test of how the same values would perform in the final road detection result after performing morphological operations and finding the continuous road area from the seeds. In these experiments I calculated precision and error rate for values $p_1 \in \{0.5, 0.2, 0.1, 0.075, 0.05\}$ and $p_2 \in \{0.05, 0.02, 0.01, 0.005, 0.001\}$. Precisions of the classification rule results are listed in table 1, error rates are in table 2. The illustration of how these values are distributed in comparison with each other is shown in figure 12. Similarly, the results of the continuous road detection are listed in tables 3 and 4. The illustration is in figure 13. The value for $p_1 = 0.05$ and $p_2 = 0.05$ wasn't calculated, since parameters should be $p_1 > p_2$.

Table 1 Precision of road classification rule based on illuminant invariance, depending on parameters p_1 and p_2.

p_2 p_1	0.05	0.02	0.01	0.005	0.001
0.5	0.755	0.758	0.773	0.770	0.786
0.2	0.745	0.777	0.769	0.776	0.772
0.1	0.766	0.761	0.770	0.764	0.782
0.075	0.761	0.770	0.780	0.766	0.768
0.05		0.765	0.779	0.763	0.779

Table 2 Error rate of road classification rule based on illuminant invariance, depending on parameters p_1 and p_2.

p_2 p_1	0.05	0.02	0.01	0.005	0.001
0.5	0.569	0.549	0.512	0.539	0.491
0.2	0.580	0.506	0.501	0.493	0.499
0.1	0.526	0.520	0.497	0.511	0.480
0.075	0.530	0.506	0.475	0.512	0.511
0.05		0.510	0.481	0.516	0.498

Table 3 Precision of road detection (continuous) based on illuminant invariance, depending on parameters p_1 and p_2.

p_2 p_1	0.05	0.02	0.01	0.005	0.001
0.5	0.894	0.890	0.896	0.900	0.922
0.2	0.895	0.897	0.893	0.891	0.895
0.1	0.900	0.897	0.888	0.894	0.890
0.075	0.898	0.899	0.901	0.892	0.889
0.05		0.906	0.895	0.894	0.892

Table 4 Error rate of road detection (continuous) based on illuminant invariance, depending on parameters p_1 and p_2.

p_2 p_1	0.05	0.02	0.01	0.005	0.001
0.5	0.322	0.306	0.285	0.316	0.282
0.2	0.311	0.281	0.269	0.269	0.266
0.1	0.287	0.272	0.270	0.258	0.259
0.075	0.282	0.268	0.250	0.273	0.282
0.05		0.254	0.260	0.267	0.277

In figure 12 and figure 13 I labeled the most promising results of low ER and high PR with letters. Labels correspond to tuples $[p_1, p_2]$ and are following:

A $[0.075, 0.01]$,
B $[0.1, 0.001]$,
C $[0.05, 0.01]$,
D $[0.5, 0.001]$,
E $[0.05, 0.02]$.

Table 5 The precision and the error rate of road classification rule in illuminant invariance method, depending on the parameter k.

k	0.1	0.2	0.3	0.4	0.5	0.6	0.7
ER	0.890	0.837	0.786	0.738	0.694	0.653	0.616
PR	0.813	0.812	0.811	0.809	0.806	0.803	0.800

k	0.8	0.9	1	1.1	1.2	1.3	1.4
ER	0.583	0.554	0.528	0.507	0.488	0.473	0.461
PR	0.796	0.793	0.788	0.784	0.780	0.775	0.770

During these experiments the decision on parameters λ_1, λ_2 and final order of morphological operations was not yet made, therefore the results may differ from the final implementation. Due to this fact, the choice of $[p_1, p_2]$ was based mostly on the results of the classification rule with λ_1, λ_2 empirically estimated. Although, the continuous road results were also considered, as they give a rough idea of how parameters would perform in future tests.

In classification rule test, values A, B, C and D have shown similar results with low ER from 0.475 to 0.491 and high PR from 0.779 to 0.786. The difference between individual values was less than 2% for ER and less than 1% for PR. In continuous road test shown in figure 13, A and B have shown similar results with about 0.5% difference in PR and 1% difference in ER. The point C had roughly 2% worse error rate than A, and more than 1% worse precision than A. The point D has shown approximately 2% better PR but almost 3.5% worse ER. In this experiment point E had good PR and ER, but in figure 12 with classification rule results it performed worse, than A, B, C and D in both of measurements. Parameters A $[0.075, 0.01]$ were chosen for following experiments, because they've shown the lowest error rate and acceptably high precision in both classification rule and continuous road tests.

Invariant angle θ

This shorter version of DS1 was used to estimate the invariant angle θ_{DS1} of the used camera. The entropy minimization algorithm was used on each image to estimate the angle. From the results I've manually chosen only ones where the angle wasn't incorrect, which resulted in 32 values. From these values I calculated the mean value $\theta_{DS1} = 153$, which was used in some of further experiments as a fixed value.

4.1.2 Illuminant invariance road detection λ_1 and λ_2

For this test I used every 10th image from dataset DS1 with fixed invariant angle $\theta_{DS1} = 153$. The thresholds λ_1 and λ_2 are calculated using the equation (42). Therefore the experiment was focused on finding the parameter k_{DS1}. The precisions and error rates of classification rule (43) are shown in table 5 and illustrated in figure 14. This figure illustrates how with increasing k both ER and PR are decreasing. The difference in error rate between the best and the worst result is approximately 43%, the difference in precision is smaller: approximately 4%. Even the smallest $ER = 0.461$ corresponding to $k = 1.4$ mean that a lot of pixels are being misclassified. Precision values mean, that on average more than 76.5% of pixels classified as road are truly road.

Figure 15 and table 6 illustrate the precision and error rate of continuous road detection depending on the parameter k. In this figure error rate is naturally decreasing with

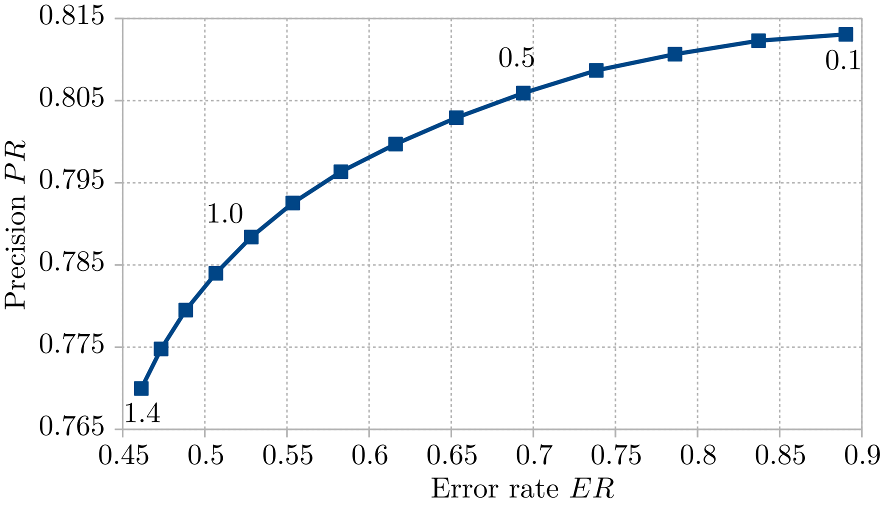


Figure 14 The precision and the error rate of road classification rule in illuminant invariance method, depending on the parameter k.

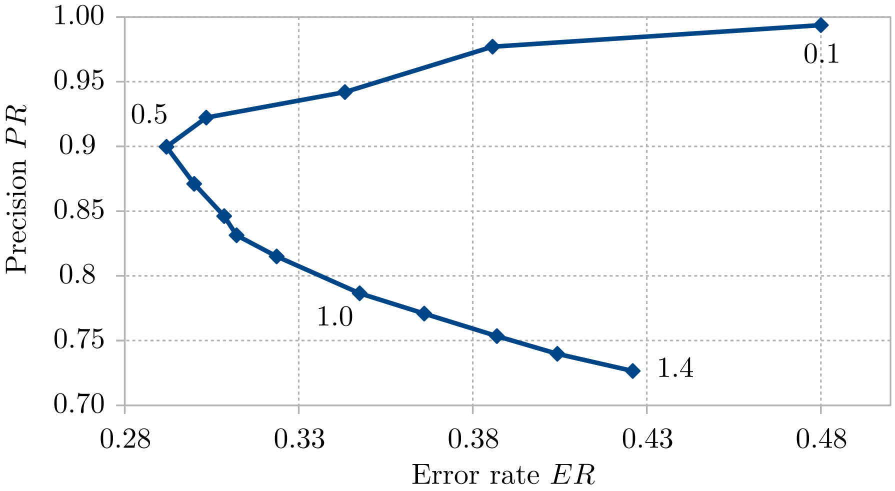


Figure 15 The precision and the error rate of continuous road found using illuminant invariance method, depending on the parameter k.

the increasing k, and at some point starts to increase back. The initial decrease in ER can be explained by the increasing number of true positives, as bigger interval between threshold λ_1 and λ_2 means more pixels are classified as road. The successive increase in error rate may be due to the excessive enlargement of the interval causing more pixels to be false positively classified. The precision is decreasing, which can indicate the increasing number of false positives.

The minimal average error rate with value $ER = 0.292$ was corresponding to $k = 0.5$, the respective precision $PR = 0.900$ was acceptable. Therefore, the parameter $k_{DS1} = 0.5$ was chosen in following experiments to calculate thresholds.

The still high error rate can be caused by a large number of false negatives present in the found road. The reason is that used morphological operations doesn't fill all the holes in the road area, as can be seen in figure 9(b).

The precision and error rate on the whole dataset DS1 using parameter $k_{DS1} = 0.5$ were $PR = 0.803$ and $ER = 0.697$ for classification rule. The results of the road detection were $PR = 0.891$ and $ER = 0.308$.

Table 6 The precision and the error rate of continuous road found using illuminant invariance method, depending on the parameter k.

k	0.1	0.2	0.3	0.4	0.5	0.6	0.7
ER	0.480	0.386	0.343	0.303	0.292	0.300	0.309
PR	0.994	0.977	0.942	0.922	0.900	0.871	0.846

k	0.8	0.9	1	1.1	1.2	1.3	1.4
ER	0.312	0.324	0.347	0.366	0.387	0.404	0.426
PR	0.831	0.815	0.787	0.771	0.753	0.740	0.726

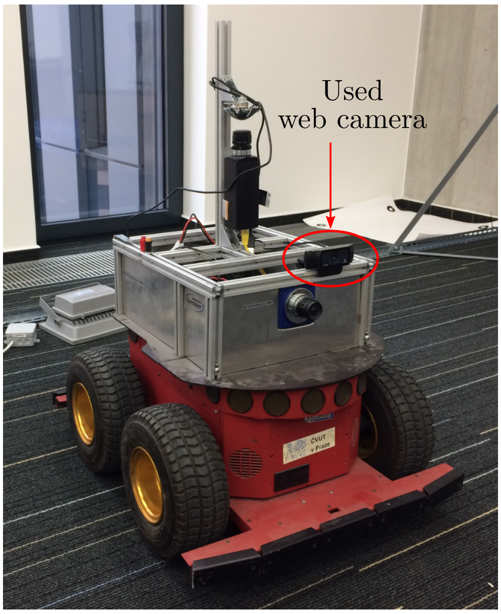


Figure 16 The mobile robot used in my experiments.

4.2 Experiments with robot

The experiments were conducted on Intel® Core™ i7-4700MQ CPU with 2.40 GHz frequency. The implementation of algorithms was in C++ language with optimization and without parallelization. The mobile robot Pioneer used in my experiments is depicted in figure 16. The USB web camera Logitech® c920 was mounted on top of the robot and connected to the laptop. The communication between the computing laptop and steering onboard computer was performed over Wi-Fi using ROS—Robot Operating System.

During experiments I encountered an issue with two web cameras I used: Microsoft® LifeCam HD-3000 and Logitech® c920. Both of them had produced images with the ring of the distorted colors in the middle of the image. In HD-3000 it was noticeable even in RGB image, as can be seen in figure 17, therefore it wasn't suitable for experiments. In c920 the ring wasn't visible to human eye on an RGB image, but it had considerable effect on the entropy minimization algorithms. It was present in grayscale illuminant invariant images. The problem seems to be isolated to web cameras, as I didn't encounter it in images from other digital cameras, mobile phones nor in datasets available on the Internet.

Due to compatibility issues I had to use the second web camera in my experiments. Due to its defect I had to use entropy minimization algorithms with big caution. I used

Figure 17 A photo of a white sheet of paper made by HD-3000 web camera.

images with shadows and empirically evaluated the shadow removal quality. I used the entropy plot over the angles to estimate the most appropriate value of the invariant angle.

Unfortunately, the experiments took place in late December, when the weather wasn't sunny. Therefore, I was unable to test how illuminant invariance part of the algorithm performed in experiments with robot. But I believe, that experiments I made on datasets available on the Internet, were sufficient to test the validity of the theory in my implementation.

4.2.1 Invariant angle estimation

The estimation of $\theta_{DS2,DS3}$ was complicated by weather and used camera. Unfortunately due to weather, I managed to obtain only 3 photos containing shadows during creation of dataset $DS2$. These images are img_1, img_2 and img_3, they are depicted in the first row of figure 18. To compensate the lack of images for the angle estimation I've made several photos at home. Two of them are also depicted in figure 18 as img_4 and img_5. These images were made in a dark room with candle being the only source of light. Candle flame is a Planckian light, therefore the theory of illuminant invariance should stand. On the other hand, candle produce light of low intensity, therefore img_4 and img_5 are very noisy, which interferes with entropy minimization algorithm.

Running entropy minimization algorithms on these 5 images resulted in several different values of θ listed in table 7. According to entropy plots depicted in the second row of figure 18, each of these angles is located on the only considerable peak and therefore is the only candidate for being invariant angle in corresponding image. Nevertheless the third row of figure 18, where the grayscale images corresponding to the minimal entropy are depicted, illustrates that 4 of 5 angles are evidently incorrect. Only angle $\theta = 152$ estimated in image img_3 had value that resulted in shadow free image. I chose it to be a fixed invariant angle for datasets DS2 and DS3 $\theta_{DS2,DS3} = 152$. The value isn't perfectly accurate, as shadows can still be visible in fourth row of figure 18, however they are vague relatively to the original image. Ideally the angle would be estimated on a bigger set of images and the value would be calculated as mean. Due to the small number of images it was my only estimate.

Table 7 The calculated invariant angle θ_{DS2} found in different images.

Image	img_1	img_2	img_3	img_4	img_5
θ	113	112	152	12	105

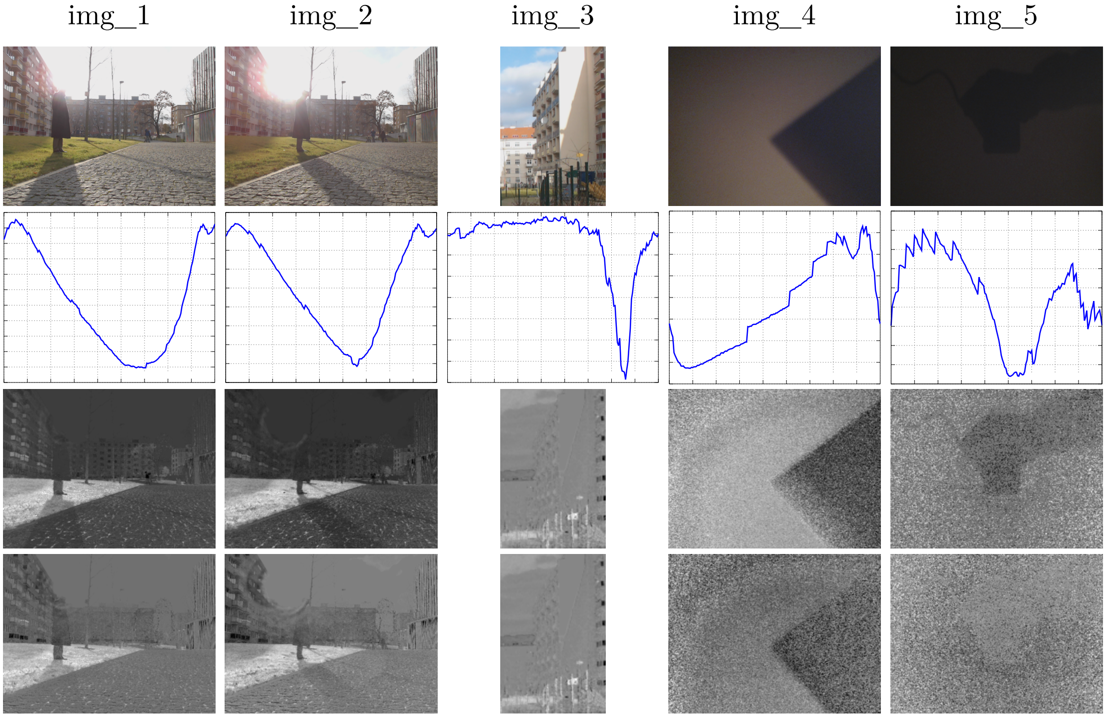


Figure 18 The illustration of the invariant angle estimation on web camera c920. **First row:** original images. **Second row:** entropy plots. **Third row:** grayscale images corresponding to the minimal entropy. **Fourth row:** grayscale images corresponding to alngle $\theta = 152$.

4.3 Road Following Algorithm

In order to test road detection algorithms on the mobile robot, I needed to implement a path following algorithm. Since, its main purpose was to be a mean in testing, I decided to keep it simple. The input to the algorithm is a black and white mask image distinguishing road and background. The forward speed is set to one of two fixed values, while the angular velocity is controlled by a PID controller.

In this algorithm I consider the coordinate system of an image as depicted in figure 19. The origin is placed in the center of a bottom row, x axes is pointing to the right and y axes is pointing up. The reference vector $\boldsymbol{v}_r$ is placed alongside y axes, its length was set so that all the patches used for creation of a road model had y coordinate smaller than $|\boldsymbol{v}_r|$. The approximate direction $\boldsymbol{v}_\mu$ where the robot is suppose to move is calculated as a mean value of road pixels. This way if there's a turn, most of the road pixels would be located in its direction and their mean value would be shifted along the x axes.

The forward speed is set to 0.3 ms^{-1}. If y coordinate of mean vector $\boldsymbol{v}_\mu$ is lower than the reference, robot is stopped. In this case, the angular velocity is set to 0.1 ms^{-1}, so robot starts to turn right, until $\boldsymbol{v}_\mu$ is appropriate to start moving forward.

When forward speed is set to 0.3 ms^{-1} the angular speed is controlled by a PID controller

$$u(t) = K_P e(t) + K_I \int_0^t e(\tau)\, d\tau + K_D \frac{\mathrm{d}e(t)}{\mathrm{d}t}, \tag{52}$$

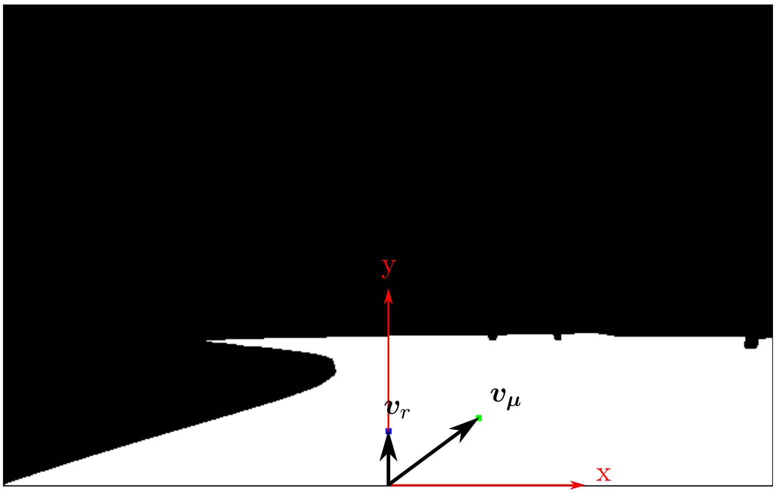


Figure 19 Vectors used to calculate the error value for PID controller.

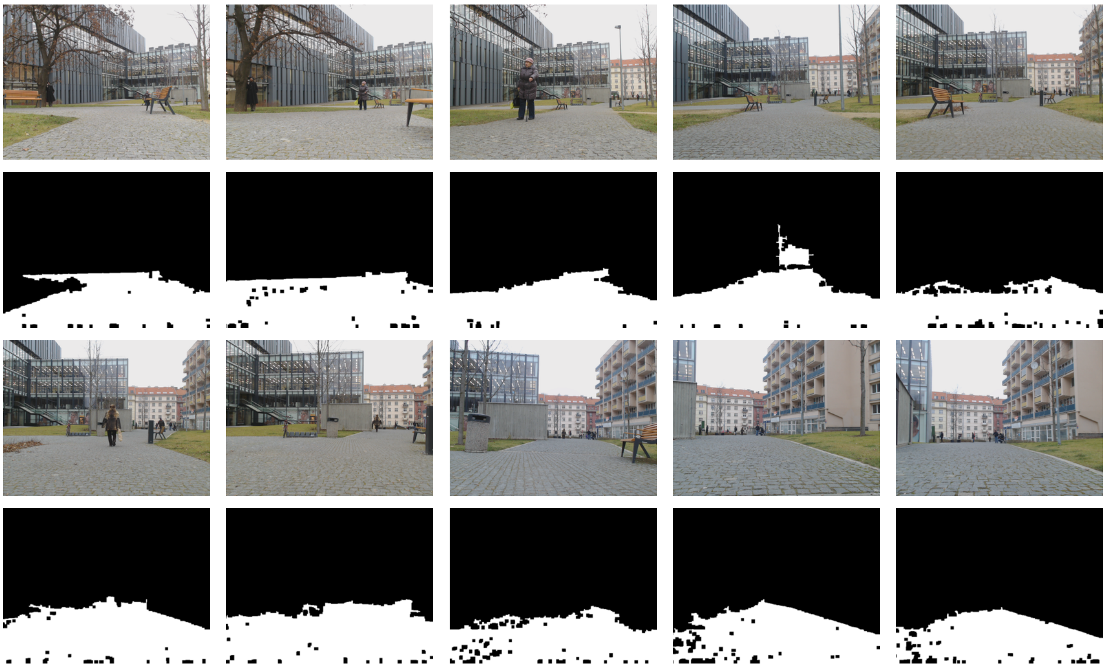


Figure 20 The results of the illuminant invariance based road detection in experiments with robot. Figure contains 10 evenly distributed images from DS3 and corresponding found paths.

where $u(t)$ is a control value and $e(t)$ is an error value. In my case, integral K_I and derivative K_D terms were set to zero, because proportional K_P was sufficient. The error value is computed as a sinus of an angle α between $\boldsymbol{v}_r$ and $\boldsymbol{v}_\mu$. Angles $-\frac{\pi}{2} \leq \alpha \leq \frac{\pi}{2}$ are expected in my application. In this interval both α and $\sin\alpha$ are monotonically increasing, therefore both functions could be used to produce a signed control value.

4.3.1 Experiments with robot and evaluation

The aim of the final experiment was to test if presented road detection techniques are sufficient for robot path following. The dataset DS3 is comprised of this experiment results.

The illustration of the illuminant invariance based road detection testing is shown

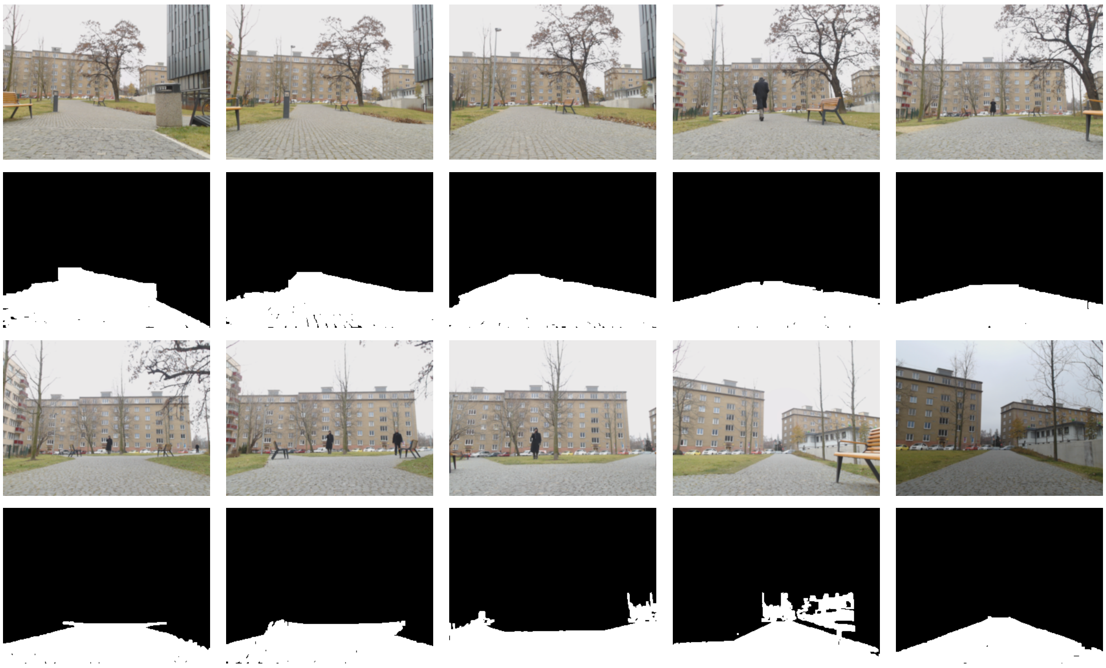


Figure 21 The results of the combined method of road detection in experiments with robot. Figure contains 10 evenly distributed images from DS3 and corresponding found paths.

in figure 20. In this experiment parameters were $\theta_{DS3} = 153$ and $k = 0.6$. During experiments with dataset DS2 I've noticed that cobblestone coverage was a difficulty to morphological operations and continuous road extraction. The amount of dilations suitable for distant part of the road wasn't enough to deal with holes created by gaps between stones in the nearer areas. Moreover the number of dilations suitable to fill these gaps created many false positives in more sensitive parts. Gaps are more visible in the bottom part of an image. I estimated that the border line can be drawn at approximately 0.79 of the height of the image. Therefore in this experiment I used different number of morphological operations on the top and on the bottom parts of the image.

Images in figure 20 illustrate that illuminant invariant road detection can successfully distinguish the road from the background. In these images the border between the road and the grass is clearly seen in the classification results. The algorithm may misclassify some background objects if they have similar to road color and no strong border between them, as is seen in the 4th image in the first row.

The images taken from camera had resolution 640 × 480. The algorithm managed to process one frame, e.g. find the road, calculate error value for PID controller and send the command to steering computer, on average in 0.45 s. In images downsampled to resolution 320 × 240 it processed one image for an average of 0.25 s.

The illustration of the combination of illuminant invariance and GMM graph cut based road detections testing is shown in figure 21. In this experiment parameters were $\theta_{DS3} = 153$, $k = 0.6$, $\xi_0 = 200$, $\xi_1 = 50$, $\gamma = 50$, numbers of GMM components were $K_0 = 5$, $K_1 = 5$. Parameter $\zeta = 25$ was set only in pixels that were classified as road by illuminant invariance road detector. Due to the time consumption of the graph cut algorithm this experiment used images downsampled to resolution 320 × 240. One frame was processed on average in 0.92 s.

The results of the road detection look better than ones from illuminant invariance based method. The road border near grass is more precise, detected road area has less

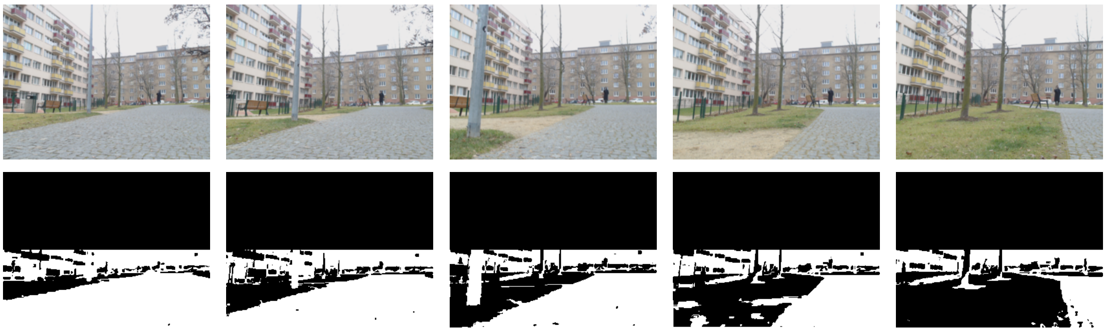


Figure 22 The results of the GMM based road detection in experiments with robot. Figure contains 5 evenly distributed images from DS3 and corresponding found paths.

holes. But the problem with false positives still remains.

In my opinion, the combination doesn't bring much improvement to the illuminant invariance method. False negative holes may be reduced by simpler algorithms. False positives are still present, although can sometimes be reduced. Overall result doesn't seem to compensate for the overcomplexity and time consumption.

The results of GMM based graph cut road detection in experiments with robot are shown in figure 22. This experiment ran with the same parameters as combined road detection method, with the exception of ζ, which was set to $\zeta = 8$ in all pixels. The algorithm also ran on downsampled images. In this experiment the crop of 50% of the image was used instead of sky detection, which was an attempt to reduce the number of false positives. The method processed one image on average for 0.6 s.

During this short experiment the robot failed to navigate along the road, due to incorrect road detection. It can be seen in figure 22, that algorithm classifies the road and the grass properly. The border between each of them is precise. However background buildings and gray colored street lighting poles were also classified as road. This can be caused by the variety of reasons.

Firstly, due to badly determined parameters. The number of different parameters in this algorithm is large. All of 6 parameters—ξ_0, ξ_1, γ, K_0, K_1 and ζ—have to be tuned in order to algorithm to work properly. I managed to try only a small number of possible combinations and could just not find the appropriate one.

Secondly, colors of road and buildings may be too similar. The training dataset was created in the same park few days earlier. I chose several images evenly distributed through bigger set of data, but also manually picked ones with unique color distribution. It's possible, that in some images the road color was similar to the building color in other images. For example, in the first row of figure 21, the road and the building may seem to have similar colors. The probability of them having the same color is increased if compared in different images. In this case GMM_0 and GMM_1 may contain similar components, resulting in incorrect classifications.

Despite the fact that GMM based graph cut didn't perform well in my experiments, it looks like a promising method. Unlike illuminant invariance based method it doesn't use seeds from the bottom part of the image. Therefore it's capable of correct road classification even if the robot is positioned outside of drivable area. The indication can be seen in the last image in figure 22.

Chapter 5

Conclusion

I presented in this thesis three methods of road detection. I mainly focused on the method based on the illuminant invariance theory, but also described Gaussian Mixture Models based graph cut technique and proposed the combination of both methods.

Firstly, I described the principle of illuminant invariance theory. Assuming Planckian light sources, Lambertian surfaces and narrow-band cameras, it allows to create a simple model of an RGB pixel. Using this model it's possible to project the pixel onto chromaticity space, i.e. color specification independent on luminance. Pixels from same surfaces form parallel lines in the log-chromaticity space, independently on their illumination conditions. The illuminant invariant (or simply shadow free) grayscale image can be produced by projecting pixels onto the perpendicular line.

The slope of this line can be found using entropy minimization algorithm. It's principle lies in searching through all possible angles. For each angle the grayscale image is computed and entropy is calculated. The image with the minimal entropy is shadow free. The found invariant angle is dependent only on camera parameters. The advantage of this method is that it can be used on an unknown camera. On the other hand, it may fail if ratio of shadow to non-shadow area isn't adequate. The method is also sensitive to noise, outliers and non-Lambertian surfaces such as sky. I describe methods used in literature to overcome all of these problems. The implemented outlier detector is based on the Chebyshev's inequality, which should allow it to adapt to the unique distribution in each image. The sky detection algorithm is based on Otsu's thresholding method. The principle lies in finding the minimum intra-class variance among all possible threshold values.

The road detection algorithm based on the illuminant invariance method assumes, that in grayscale image the road area have a uniform gray level. The road model is created from the pixels located in the bottom part of the image. Afterwards, the remaining pixels are classified as a road or as a background according to it. The set of morphological operations is applied to close the gaps between true positives and cut out false positives.

The second implemented method of road detection uses Gaussian Mixture Models to create color models of road and non-road areas from training dataset. The road detection is performed by finding the minimum cut in a weighted graph created from the image. In this graph pixels correspond to nodes. Two additional terminal nodes corresponding to road and background are present. Edges' weights are defined using trained color models and structural term.

The combination of both methods firstly detects road using illuminant invariant method. The results of this detection are then used in creation of weighted graph for graph cut method.

Finally, I describe my parameter estimation, conducted experiments and the results. In my experiments it seems that illuminant invariant method have outperformed the graph cut methods. It was much faster: 0.25 s in comparison to 0.6 s and 0.92 s. The road detection was more accurate than in GMM based graph cut.

The future work may be focused on combining the illuminant invariant road detector with other techniques in order to reduce the number of false positives near the road area. Good gap closing algorithms increasing the number of true positives can also improve results. While I don't see the combination of illuminant invariant and graph cut methods as perspective, standalone graph cut may be enriched in variety of ways. For example, by finding better parameters, by using adaptive structure term and by updating trained color models.

Appendix A
CD content

The content of the attached CD is described in table 8.

Table 8 The content of the attached CD.

File or directory	Description
text/	Source files of the thesis text
tracking/	The ROS node for road detection and tracking
diploma.pdf	This thesis in PDF format
DS2.zip	Dataset DS2
DS3.zip	Dataset DS3 with results
training.zip	Training dataset created from DS2

Bibliography

[1] V.H. Mistry and R. Makwana. “Survey: Vision based road detection techniques”. In: *Int. J. Comput. Sci. Inf. Technol* 5 (2014), pp. 4741–4747.

[2] B. Wang and V. Frémont. “Fast road detection from color images”. In: *Intelligent Vehicles Symposium (IV), 2013 IEEE*. June 2013, pp. 1209–1214. DOI: 10.1109/IVS.2013.6629631.

[3] H. Dahlkamp et al. “Self-supervised Monocular Road Detection in Desert Terrain”. In: *Robotics: science and systems*. Vol. 38. Philadelphia. 2006.

[4] H. Zhou et al. “Efficient Road Detection and Tracking for Unmanned Aerial Vehicle”. In: *IEEE Transactions on Intelligent Transportation Systems* 16.1 (Feb. 2015), pp. 297–309. ISSN: 1524-9050.

[5] Y. He, H. Wang, and B. Zhang. “Color-based road detection in urban traffic scenes”. In: *Intelligent Transportation Systems, IEEE Transactions on* 5.4 (2004), pp. 309–318.

[6] G.D. Finlayson, M.S. Drew, and C. Lu. “Entropy Minimization for Shadow Removal”. In: *International Journal of Computer Vision* 85.1 (2009), pp. 35–57. ISSN: 1573-1405. DOI: 10.1007/s11263-009-0243-z.

[7] J.M.A. Álvarez and A.M. López. “Road Detection Based on Illuminant Invariance”. In: *Intelligent Transportation Systems, IEEE Transactions on* 12.1 (Mar. 2011), pp. 184–193. ISSN: 1524-9050. DOI: 10.1109/TITS.2010.2076349.

[8] T. Krajnik, J. Blazicek, and J.M. Santos. “Visual road following using intrinsic images”. In: *Mobile Robots (ECMR), 2015 European Conference on*. Sept. 2015, pp. 1–6. DOI: 10.1109/ECMR.2015.7324212.

[9] H. Wang et al. “Image and Graphics: 8th International Conference, ICIG 2015, Tianjin, China, August 13-16, 2015, Proceedings, Part III”. In: ed. by Yu-Jin Zhang. Cham: Springer International Publishing, 2015. Chap. Road Detection Based on Image Boundary Prior, pp. 212–222. ISBN: 978-3-319-21969-1. DOI: 10.1007/978-3-319-21969-1_19.

[10] H. Kong, J. Y. Audibert, and J. Ponce. “General Road Detection From a Single Image”. In: *IEEE Transactions on Image Processing* 19.8 (Aug. 2010), pp. 2211–2220. ISSN: 1057-7149. DOI: 10.1109/TIP.2010.2045715.

[11] J. M. Álvarez, T. Gevers, and A. M. López. “Vision-based road detection using road models”. In: *2009 16th IEEE International Conference on Image Processing (ICIP)*. Nov. 2009, pp. 2073–2076. DOI: 10.1109/ICIP.2009.5414321.

[12] G.D. Finlayson, S.D. Hordley, and P.M. Hubel. “Color by correlation: A simple, unifying framework for color constancy”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 23.11 (2001), pp. 1209–1221.

[13] T. Acharya and A.K. Ray. *Image Processing: Principles and Applications*. Wiley, 2005. ISBN: 9780471719984.

[14] M.R. Peres. *The Focal Encyclopedia of Photography: Digital Imaging, Theory and Applications, History, and Science*. Focal Press. Elsevier/Focal Press, 2007. ISBN: 9780240807409.

[15] G.D. Finlayson, M.S. Drew, and B.V. Funt. "Spectral sharpening: sensor transformations for improved color constancy". In: *J. Opt. Soc. Am. A* 11.5 (May 1994), pp. 1553–1563. DOI: `10.1364/JOSAA.11.001553`.

[16] M.S. Drew et al. "Sensor transforms for invariant image enhancement". In: *Color and Imaging Conference*. Vol. 2002. 1. Society for Imaging Science and Technology. 2002, pp. 325–330.

[17] M.F. Modest. *Radiative Heat Transfer*. Elsevier Science, 2013. ISBN: 9780123869906.

[18] Graham D Finlayson and Steven D Hordley. "Color constancy at a pixel". In: *JOSA A* 18.2 (2001), pp. 253–264.

[19] G.D. Finlayson and M.S. Drew. "4-sensor camera calibration for image representation invariant to shading, shadows, lighting, and specularities". In: *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*. Vol. 2. 2001, 473–480 vol.2. DOI: `10.1109/ICCV.2001.937663`.

[20] G. Strang. *Linear Algebra and Its Applications*. Thomson, Brooks/Cole, 2006. ISBN: 9780030105678.

[21] P. Kumar. "Intrinsic Image Based Moving Object Cast Shadow Removal in Image Sequences". In: *Digital Image Computing Techniques and Applications (DICTA), 2011 International Conference on*. Dec. 2011, pp. 410–415. DOI: `10.1109/DICTA.2011.76`.

[22] B. G. Amidan, T. A. Ferryman, and S. K. Cooley. "Data outlier detection using the Chebyshev theorem". In: *2005 IEEE Aerospace Conference*. Mar. 2005, pp. 3814–3819. DOI: `10.1109/AERO.2005.1559688`.

[23] American Heritage Dictionary. *The American Heritage Science Dictionary*. Houghton Mifflin Company, 2005. ISBN: 9780618455041.

[24] A.M. Neto et al. "Robust horizon finding algorithm for real-time autonomous navigation based on monocular vision". In: *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE. 2011, pp. 532–537.

[25] N. Otsu. "A threshold selection method from gray-level histograms". In: *Automatica* 11.285-296 (1975), pp. 23–27.

[26] S.U. Lee, S.Y. Chung, and R.H. Park. "A comparative performance study of several global thresholding techniques for segmentation". In: *Computer Vision, Graphics, and Image Processing* 52.2 (1990), pp. 171–190.

[27] M. T. Orchard and C. A. Bouman. "Color quantization of images". In: *IEEE Transactions on Signal Processing* 39.12 (Dec. 1991), pp. 2677–2690. ISSN: 1053-587X.

[28] A.C. Rencher. *Methods of Multivariate Analysis*. Wiley Series in Probability and Statistics. Wiley, 2003. ISBN: 9780471461722.

[29] Y. Y. Boykov and M. P. Jolly. "Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images". In: *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*. Vol. 1. 2001, 105–112 vol.1. DOI: `10.1109/ICCV.2001.937505`.

[30] Y. Boykov and V. Kolmogorov. “An Experimental Comparison of Min-cut/Max-flow Algorithms for Energy Minimization in Vision”. In: *Energy Minimization Methods in Computer Vision and Pattern Recognition: Third International Workshop, EMMCVPR 2001 Sophia Antipolis, France, September 3–5, 2001 Proceedings*. Ed. by M. Figueiredo, J. Zerubia, and A. K. Jain. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 359–374. ISBN: 978-3-540-44745-0. DOI: 10.1007/3-540-44745-8_24.