

Visual Road Following Using Intrinsic Images

Tomáš Krajník¹

Jan Blažíček²

João M. Santos¹

Abstract—We present a real-time visual-based road following method for mobile robots in outdoor environments. The approach combines an image processing method, that allows to retrieve illumination invariant images, with an efficient path following algorithm. The method allows a mobile robot to autonomously navigate along pathways of different types in adverse lighting conditions using monocular vision.

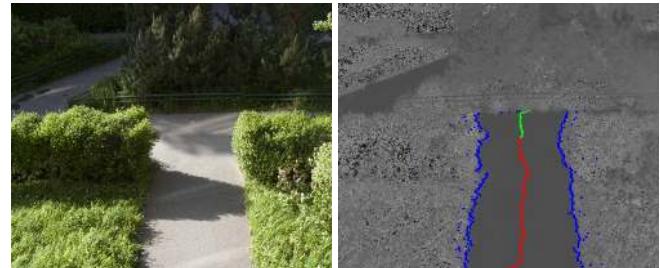
To validate the proposed method, we have evaluated its ability to correctly determine boundaries of pathways in a challenging outdoor dataset. Moreover, the method’s performance was tested on a mobile robotic platform that autonomously navigated long paths in urban parks. The experiments demonstrated that the mobile robot was able to identify outdoor pathways of different types and navigate through them despite the presence of shadows that significantly influenced the paths’ appearance.

Index Terms—visual navigation, mobile robotics

I. INTRODUCTION

One of the most desired features of robotic systems is autonomy – the capability to act without the need of human intervention. In the case of mobile robots, the key skill is autonomous navigation – the ability to reach a desired destination on its own. While the ability of reliable autonomous navigation in outdoor environments has been already demonstrated during the DARPA Grand [1] and DARPA Urban [2] Challenges, the vehicles that achieved autonomous navigation along the desired paths were equipped with expensive sensory and computational equipment and relied on precise RTK-GPS systems. Their dependence on expensive sensors and high computational demands makes wider, real-world deployment of the aforementioned navigation methods in small robots rather difficult.

Among other sensoric equipment, small and affordable digital cameras are a staple in mobile robotics along with laptops and mobile devices, which have arguably gotten powerful enough to run most machine vision techniques in real time. The main disadvantage of standard cameras stems from their passive nature – the performance of visual navigation and localization systems is heavily influenced by illumination factors. This is a problem especially in outdoor environments, there the (unstable and somewhat unpredictable) illumination determines the appearance of the scene perceived by the robot. One of the particular problems



(a) Original image

(b) Intrinsic image + detected path

Fig. 1: Shadow removal and path detection example.

is the presence of shadows that move as the main illuminant of outdoor scenes, the Sun, changes its position. The presence of shadows causes problems especially to visual based scene segmentation, because even a single narrow shadow cast over a surface splits it into two disjoint segments. This is especially problematic when the mobile robot is using monocular camera to identify traversable terrain, such as pathways in vegetation.

We present a computationally efficient visual navigation system that allows a mobile robot to identify and traverse distinct pathways in outdoor environments under difficult lighting conditions. The proposed approach combines a photometric method, which can infer material reflectance properties under varying illumination, with a path-following algorithm, that can guide a mobile robot along arbitrary pathways. The image preprocessing method exploits the fact that the Sun acts as a black body radiator and the distribution of its wavelengths, which is determined by its temperature, follows Plack’s law. Along with a self-calibration step, which determines the sensitivity of the cameras color sensors to particular parts of the Sun’s spectrum, the method can calculate reflectances of the perceived objects in a pixel-wise manner. This effectively removes any shadows cast over the scene perceived by the robot’s on-board camera. After the shadows are removed, we employ a fast image processing method that identifies boundaries of the pathway which the robot moves on. The positions of the path boundaries are directly transformed to velocity and steering commands for the robot, which is guided along the path.

While the use of intrinsic images for shadow removal in outdoor environments is not novel, the technique has been so far used to improve robustness of vision-based localization [3], [4], [5], [6]. However, our method does not rely on a priori known maps, but simply steers the robot along pathways which are detected in intrinsic images.

¹Lincoln Centre for Autonomous Systems, University of Lincoln, UK {tkrajnik, santos}@lincoln.ac.uk

²Faculty of Information Technology, Czech Technical University in Prague, Czech Republic {blazicekj}@gmail.com

The work was funded by EU project 600623 ‘STRANDS’. Kudos to those who helped with experiments.

II. RELATED WORK

Path detection algorithms, which allow for safe traversal of artificial roads and natural pathways in outdoor environments without using detailed, a priori known maps, are an important component of mobile robot navigation systems. The problem of path identification is typically tackled either by using laser-based range-finding methods, or by vision-based stereo or monocular sensing. However, using vision in outdoor environments is a challenging problem because of the inherently unstable lighting conditions. Thus, the problem of reliable path detection in outdoor environments has attracted attention of the robotics community for decades.

Cristóforis et al. [7] proposed a novel real-time image-based monocular path detection that allows to detect delimited or semi-structured outdoor paths. In their method, the images are segmented into super-pixels and each super-pixel is classified in order to detect the navigable space and thereby to calculate the path contour, which is used to guide the robot. The image processing algorithm was computationally intensive and it had to be implemented on a low-power GPU to achieve real-time performance.

A fast ‘line-scan’ method is presented in [8], where the robot uses a single camera to build a topological map of pathways in outdoor environment. To increase the robustness to illumination, the method uses only hue and saturation channels of the HSV color space to classify the individual image pixels. Moreover, it assumes that the hue and saturation of the path pixels are provided by the robot’s operator.

Another approach to the path following problem, that also uses a RGB-D monocular camera, is presented by Huang et al. [9]. They combine the range-limited depth information with (long-range) RGB imagery to create cost maps which are subsequently used to plan the robot path. Their approach allows to combine the reliability of the depth-based terrain classification with the ability to detect and avoid cul-de-sacs identified in the RGB data.

The authors of [10] not only combine range data with RGB imagery to create maps that allow elaborate planning of the vehicle path across the navigable terrain, but also use the range data to provide the RGB-based terrain classification method with samples of navigable and untraversable pixels. This allows the RGB-based classification method to adapt itself to different environments on-the-fly.

The paper [11], which is based on monocular sensing as well, proposes to achieve path following by combination of vanishing point detection and color-based probabilistic segmentation. Their vanishing point detection closely follows the method presented in [12], which fuses the responses of Gabor filters applied on the image by means of particle filtering, and the ‘roadness’ of the image pixels is classified by means of Gaussian Mixtures. The authors of [11] tested their algorithm only on datasets due to its computational complexity and report that its real-time implementation requires specialized DSP- or FPGA- based hardware solutions. Most of the aforementioned approaches either assume that the color model of the shadowed and illuminated path is

known in advance or use additional, expensive sensors that provide their autonomous navigation systems with range measurements.

One of the first visual navigation approaches that was explicitly dealing with the problems of shadows was implemented for an automated harvester [13]. The system’s purpose was to track the boundary between the cut and uncut crop, guide the harvester along it and detect the ends of the crop lane. Apart from being able to adapt to slowly changing illumination due to weather changes, it had to deal with situations where the harvester would cast a shadow on the scene. The shadow removal was based on a photometric method that relied on known parameters of the crops reflectance and Sun spectrum.

The authors of [14] and [15] proposed a method of obtaining gray-scale lighting invariant images by taking an RGB color image and combining its channels in a way that factors out its dependency on illumination intensity and color. The approach was based on assumption that the camera RGB sensors have a narrow band, the scene is illuminated by a black body irradiant (the Sun) and that the surface reflectance is Lambertian. The article [16] goes a step further and presents a method of restoring the original color information by comparing the edges found in the shadow-less invariant image to the edges found in the original. Furthermore [17] presents an automated self-calibration method that allows to obtain the lighting-invariant images without the explicit knowledge of the camera parameters.

The aforementioned Finlayson-Marchant color constancy method (coined the ‘intrinsic image’) recently found its application in the field of mobile robotics. In the papers [4], [3], [5], [6] the method was used to improve long-term localization of mobile robots in outdoor environments that are subject to significant appearance changes due to varying illumination. The paper [3] presents a system that utilizes traditional feature-based visual localization on both standard and intrinsic images and demonstrates the improvement of localization robustness on off-line datasets. The article [4] further elaborates on the method by showing that it improves segmentation and interpretation of outdoor urban scenes. Finally, the works [5], [6] demonstrate that use of the color constancy improves performance of teach-and-repeat methods in difficult lighting conditions.

The aforementioned methods used the intrinsic image for the problem of robotic self-localization, where the robot has to use its sensory input and a map to determine its true position. The method presented in this paper does not perform localization of the robot in a priori known maps. Rather than that, we apply the intrinsic image for the problem of reactive visual-based road following, where the robot’s steering and velocity is determined solely from its on-board camera image. Contrary to [7] and [8], which deal with the problem of shadows cast over the pathways by using separate color models for shadowed and illuminated path, our method uses the intrinsic image method to suppress the effect of shadows prior to the path detection / pixel classification step.

III. INTRINSIC IMAGE

The term ‘intrinsic image’ was introduced in [18], which is concerned with decomposition of images into different layers, representing intrinsic properties like shading, reflectance or shape. One of the promising approaches that allows to retrieve material reflectance from a single image, was introduced by [14] and [15], who proposed addressing the problem from a purely photometric point of view. The aforementioned approaches are based on an assumption that the camera RGB sensors have an infinitely narrow band, the illumination is provided by a black body and all surfaces in the scene are Lambertian. While it is clear that none of these assumptions are completely satisfied, the results presented in [16] have demonstrated feasibility of the approach for shadow removal. Moreover, authors of [4] and [3] have shown that the use of the aforementioned method improves visual-based mobile robot localization.

Let us briefly review the idea of the intrinsic image calculation as presented in [14]. Let us denote the response of a single RGB element of the camera sensor as ρ_k , where $k = r, g, b$ corresponds to the red, green and blue channels respectively. Then

$$\rho_k = \sigma \int E(\lambda) S(\lambda) Q_k(\lambda) d\lambda, \quad (1)$$

where σ is a constant factor denoting the Lambertian shading term contributing to the overall light reflected at the sensor, $E(\lambda)$ is the illumination of the surface element perceived, $S(\lambda)$ denotes the surface reflectance function, $Q_k(\lambda)$, is the spectral sensitivity of RGB sensors and λ denotes wavelength. Assuming Planckian illuminant and infinitely narrow-band color sensitivities given by q_k , Finlayson argues that the equation (1) can be simplified to

$$\rho_k = \sigma E(\lambda_k) S(\lambda_k) q_k. \quad (2)$$

Assuming black body irradiant and using Wien’s approximation to Planck’s law allows to parametrise the spectral power distribution solely by its color temperature T as

$$E(\lambda, T) = I c_1 \lambda^{-5} e^{-\frac{c_2}{T\lambda}}, \quad (3)$$

where c_1, c_2 are constants and I characterizes the overall light intensity. Thus, the Equation (1) can be rewritten as

$$\rho_k = \sigma I c_1 \lambda_k^{-5} e^{-\frac{c_2}{T\lambda_k}} S(\lambda_k) q_k. \quad (4)$$

Calculating a logarithm of the red and green sensor response relatively to the response of the blue sensor leads to the term

$$\chi_{r,g} = \log \frac{\rho_{r,g}}{\rho_b}. \quad (5)$$

Substituting Equation (4) to (5) leads to

$$\chi_{r,g} = \log \frac{s_{r,g}}{s_b} + \frac{\lambda_{r,g}^{-1} - \lambda_b^{-1}}{T}, \quad (6)$$

where $s_k = \lambda_k^{-5} S(\lambda_k) q_k$. Note that the first term is determined only by the camera sensitivity parameters λ_k and q_k and surface reflectance $S(\lambda_k)$, while the second term depends only on the camera parameters and temperature of

the illuminant. Rewriting the Equation (6) in vector form results in

$$\chi' = s + \frac{1}{T} \mathbf{e}, \quad (7)$$

where χ' is a vector representing log-chromaticity of the image obtained, s is a vector independent of the illuminant and \mathbf{e} is a vector independent of the surface. A graphical representation of 2D log-chromaticity space modeled by Equation (7) is shown on Figure 2. Here, the log-chromaticity of a particular pixel (blue) is given by the surface and the illumination properties. As the T parameter varies, the log-chromaticity χ' moves along a straight line given by the vector \mathbf{e} . Correct estimation of the illuminant variation direction

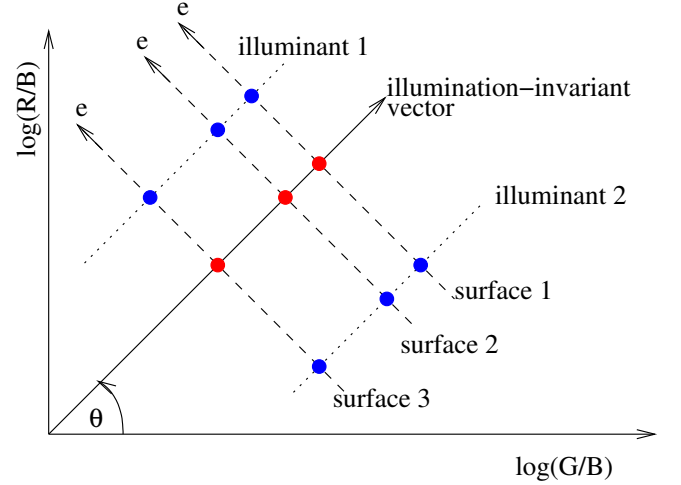


Fig. 2: Log-chromaticity space explained: the log-chromaticity of the perceived pixels (blue) is given by the surface reflectivity properties and color temperature of the illuminants. Knowing the angle θ allows to project the perceived pixel into a position on the illumination-invariant vector (red) that characterizes the surface reflectivity. The projected positions of the pixels constitute the grayscale values of an intrinsic image.

\mathbf{e} (which is determined by the camera parameters), allows to project the log-chromaticity vector χ' onto the vector orthogonal to \mathbf{e} . Theoretically, the angle θ of this orthogonal vector can be calculated from the camera sensor sensitivities. However, these sensitivities are not always known, because typical camera can operate in a wide variety of settings that significantly affect its parameters. From a practical point of view, one can rather try to find a suitable θ by self-calibration technique described in [17], which proposes to select the angle θ so that the set of projected chromaticity values has the lowest entropy.

To avoid numerical stability issues caused by potentially small values of ρ_b , we follow the paper [17] and calculate a 3D log-chromaticity space, where the responses of the individual channels are not relative to the blue channel as in Equation (5), but to a geometric mean of the channels, i.e. $\sqrt[3]{\rho_b \rho_r \rho_g}$. Otherwise, our implementation closely follows the aforementioned description.

IV. IMPLEMENTATION

The path following method consists of two parallel threads. The first, ‘control’ thread is responsible for generating the steering commands that guide the robot along a given path. The second, ‘adaptation’ thread calculates the value of the angle θ by means of entropy minimization.

The control thread uses the latest camera image and latest estimation of the parameter θ to calculate an intrinsic image. The intrinsic image is then searched for a uniform segment that is likely to represent a traversable path. Then, the boundaries of the path are used to generate the steering commands that keep the robot in the middle of the path while avoiding obstacles. The ‘adaptation’ thread, that runs with a lower priority, is used to adapt the value of the θ according to the perceived scene.

Both algorithms are implemented in C/C++ language, use the dc1394 library to communicate with the camera and the Aria library to interface with the control board of the P3AT robot, which was used for the experimental evaluation.

A. Intrinsic image calculation

After receiving the image from the camera, the invariant image is calculated according to the description in Section III, see Algorithm 1.

Algorithm 1 Invariant image calculation

```

1: function GETINVARIANTIMAGE(I,  $\theta$ )
2:   for each row of I do
3:     for each column of I do
4:        $\rho_{r,g,b} \leftarrow \mathbf{I}(r, g, b)$  remap to  $[0, 1]$ 
5:        $\rho_M \leftarrow \sqrt[3]{\rho_r \cdot \rho_g \cdot \rho_b}$   $\triangleright$  normalization
6:        $\chi_{r,g,b} \leftarrow \log(\rho_{r,g,b}/\rho_M)$   $\triangleright$  3D log-chrom.
7:        $\psi_{1,2} \leftarrow \mathbf{U} \times \chi_{r,g,b}$   $\triangleright$  project to 2D
8:        $\mathbf{I}_i \leftarrow (\psi_1 \cos \theta + \psi_2 \sin \theta)$   $\triangleright$  intrinsic pixel
9:     end for
10:  end for
11:  return  $\mathbf{I}_i$   $\triangleright$  illumination-invariant if  $\theta$  is correct
12: end function

```

Following the Algorithm 1 line by line, we can see that all operations are performed pixel-wise, which is computationally efficient and parallelizable. The matrix \mathbf{U} used for projection from the 3D log-chromaticity to a 2D log-chromaticity space is set to the value recommended in [17].

B. Path following

Having obtained the invariant image, we can proceed to find the path for the robot to move across. The algorithm first performs histogram equalization of the intrinsic image, which allows for a wider separation of the path from the background. The equalized image is then thresholded and the image’s pixels are classified as either ‘path’ or ‘background’. The algorithm then scans the image line by line from the bottom up, identifying the edges and obstacles on the path ahead of the robot in simple pixel-wise operations. If the width of the path detected falls below a certain threshold, the

algorithm terminates the image analysis and calculates the robot’s steering from the average deviation of the detected path center from the centre of the image. The forward speed is determined from the row at which the algorithm was terminated – the higher the row, the higher the robot’s forward speed. To slow down the robot when taking sharp turns, the forward speed is decreased by the absolute value of the steering speed. The main advantage of the method, which is summarized as Algorithm 2, is its computational efficiency and robustness to clutter.

Algorithm 2 Path following method

```

1: function FINDPATH(I,  $\tau$ ,  $gap_{max}$ ,  $width_{min}$ )
2:    $\omega \leftarrow 0$   $\triangleright$  initialize steering value
3:    $\mathbf{I} \leftarrow equalize(\mathbf{I})$   $\triangleright$  histogram equalization
4:    $\mathbf{I} \leftarrow \mathbf{I} > \tau$   $\triangleright$  image thresholding
5:    $path\_center \leftarrow \mathbf{I}.width/2$   $\triangleright$  search position init
6:   for Each  $y$  of I starting from  $(\mathbf{I}.height - 1)$  do
7:      $gap \leftarrow 0$   $\triangleright$  reset boundary counter
8:      $right\_edge \leftarrow \mathbf{I}.width-1$   $\triangleright$  right boundary init
9:     for Each  $x$  from  $path\_center$  to  $\mathbf{I}.width$  do
10:      if  $\mathbf{I}[x, y] == 0$  then  $\triangleright$  non-path pixel?
11:         $gap \leftarrow gap + 1$   $\triangleright$  increase counter
12:        if  $gap \geq gap_{max}$  then  $\triangleright$  boundary found
13:           $right\_edge \leftarrow (x - gap_{max})$ 
14:          break out of FOR loop
15:        end if
16:      else  $\triangleright$  path pixel?
17:         $gap \leftarrow 0$   $\triangleright$  reset boundary counter
18:      end if
19:    end for
20:     $gap \leftarrow 0$   $\triangleright$  reset boundary counter
21:     $left\_edge \leftarrow 0$   $\triangleright$  left path boundary init
22:    for Each  $x$  from  $path\_center$  to 0 do
23:      if  $\mathbf{I}[x, y] == 0$  then  $\triangleright$  non-path pixel?
24:         $gap \leftarrow gap + 1$   $\triangleright$  increase counter
25:        if  $gap \geq gap_{max}$  then  $\triangleright$  boundary found
26:           $left\_edge \leftarrow (x + gap_{max})$ 
27:          break out of FOR loop
28:        end if
29:      else  $\triangleright$  path pixel?
30:         $gap \leftarrow 0$   $\triangleright$  reset boundary counter
31:      end if
32:    end for
33:     $width \leftarrow right\_edge - left\_edge$ 
34:    if  $width \leq width_{min}$  then  $\triangleright$  path end found
35:       $path\_length \leftarrow (\mathbf{I}.height - y)$ 
36:      Break out of FOR loop
37:    end if
38:     $path\_center \leftarrow (right\_edge + left\_edge)/2$ 
39:     $\omega \leftarrow \omega + path\_center - \mathbf{I}.width/2$ 
40:  end for
41:   $\omega \leftarrow c_\omega \omega$   $\triangleright$  calculate the steering speed
42:   $v \leftarrow c_v(path\_length) - |\omega|$   $\triangleright$  and forward speed
43:  return  $v, \omega$   $\triangleright$  send  $v$  and  $\omega$  to the robot’s controller
44: end function

```

C. Calculating the angle θ

To calculate a the invariant image, we have to decide on the best angle θ . To estimate a suitable θ , we used a method based on minimization of Shannon’s entropy of the intrinsic image. Shannon’s entropy computation is implemented as an intrinsic pixel probability distribution estimation obtained from a histogram. The calculation of the histogram requires to outlying values and optimal choice of the histogram bin width. We address the problem of outlying sample removal simply by discarding 5% of the darkest and brightest pixels and the bin width is determined using the Scott’s normal reference rule [19]. The algorithm for choosing the suitable parameter θ is described in Algorithm 3. Because of the

Algorithm 3 Entropy-based selection of θ

```

1: function GETBESTANGLE(I)
2:    $i_{max} \leftarrow 0.95$            ▷ maximal pixel intensity
3:    $i_{min} \leftarrow 0.05$         ▷ minimal pixel intensity
4:    $H_{min} \leftarrow \infty$        ▷ initialize entropy value
5:    $\theta_{best} \leftarrow -1$      ▷ and the corresponding angle
6:    $s \leftarrow \mathbf{I}_{height} \cdot \mathbf{I}_{width}$    ▷ number of pixels
7:   for each  $\theta$  from 0 to  $180^\circ$  do
8:      $o \leftarrow 0$              ▷ number of outliers
9:      $\mu \leftarrow 0$            ▷ initialize mean
10:     $\mathbf{I}' \leftarrow getInvariantImage(\mathbf{I}, \theta)$ 
11:    for each pixel value  $i$  in  $\mathbf{I}'$  do
12:      if  $(i \geq i_{max}) \vee (i \leq i_{min})$  then
13:         $o \leftarrow o + 1$      ▷ update number of outliers
14:      else
15:         $\mu \leftarrow \mu + i$    ▷ update the mean of inliers
16:      end if
17:    end for
18:     $\mu \leftarrow \mu / (s - o)$    ▷ calculate the mean
19:     $\sigma \leftarrow 0$          ▷ initialize std. deviation
20:    for each  $i$  in  $\mathbf{I}'$  do
21:      if  $i_{min} < i < i_{max}$  then
22:         $\sigma \leftarrow \sigma + (i - \mu)^2$ 
23:      end if
24:    end for
25:     $\sigma \leftarrow \sqrt{\sigma / (s - o)}$  ▷ calculate the std. deviation
26:     $m \leftarrow s - o$          ▷ calculate number of inliers
27:     $h \leftarrow 3.49 \cdot \sigma \cdot m^{-1/3}$  ▷ Scott’s rule for bin width
28:     $H \leftarrow H(\mathbf{I}', h, i_{min}, i_{max})$  ▷ Shannon entropy
29:    if  $(H < H_{min})$  then     ▷ if entropy H minimal
30:       $H_{min} \leftarrow H$      ▷ store the value of H
31:       $\theta_{best} \leftarrow \theta$  ▷ and store best angle  $\theta$ 
32:    end if
33:  end for
34:  return  $\theta_{best}$            ▷ return the best angle  $\theta$ 
35: end function

```

algorithm’s computational demands, it cannot process the images at high rates and thus it runs as a separate thread with low priority. If required, our system allows to initiate the Algorithm 3 manually as well.

V. EXPERIMENTS

To evaluate the algorithm’s ability to deal with uneven illumination, we have performed a serie of experiments on the datasets and with a real mobile robot.

A. Dataset-based evaluation

For offline evaluation, we have used a dataset of over 1000 images collected in the ‘Stromovka’ urban park in Prague, Czech Republic. These were collected by a P3AT mobile robot equipped with a Fire-I 400 digital camera that was tilted towards the ground in order to get a good overview of the terrain on front of the robot. The dataset images contain a large variety of pathways composed of cobblestone, concrete and sand lined by grass, fallen leaves, ornamental flowers and trees. The algorithm’s efficiency was evaluated by means of its ability to correctly recognize that the area in front of the robot was navigable while the surrounding areas do not represent the path. The algorithm did well in reducing the difference between sunlit and shadowed areas of the pathways, which resulted in correct road segmentation in 99.4% of the dataset’s images. Processing of one 640×480 image on a 2GHz Core 2 Duo CPU took approximately 140 ms. A typical example of such shadow removal and segmentation result can be found on Figures 1 and 3. To document on the results of this evaluation, we also provide the dataset’s raw and segmented images as a video sequence [20].

The entropy-based selection of the angle θ , that is needed to calculate the intrinsic image, had an interesting secondary effect. In cases when one side of the path was lined with sand while the other with grass, which would present a problem in color-based path-finding algorithms, the angle θ was set in a way that the resulting intrinsic intensity values for both foliage and sand were similar, which resulted in reliable segmentation of the pathway. However, the entropy-based selection occasionally failed in cases when the intensity of shadows was low and the scenes were cluttered. In these cases, the entropy of the intrinsic images was affected by the clutter more than by the shadows, which caused the Algorithm 3 to generate a wrong value of θ , which typically lead to failure in recognizing the path.

B. Evaluation on a mobile robot

The final test was conducted in the Stromovka park in clear-sky, sunny weather. The park is heavily populated by trees, which cast a lot of shadows on the pathways, see Figure 4. However, the algorithm performed flawlessly and the robot was able to autonomously navigate narrow concrete and cobblestone pathways across a 350 m long path while ignoring the shadows. After the test, we have evaluated the results of the path detection and identified a few images, where the path was not segmented correctly, see Figure 4b. However, the path detection was incorrect only in areas around the horizon, which did not affect the steering commands send to the robot. Moreover, the effect disappeared when the robot moved closer to the incorrectly segmented area. During the experiment, the robot has successfully processed approximately 3000 images taken by its

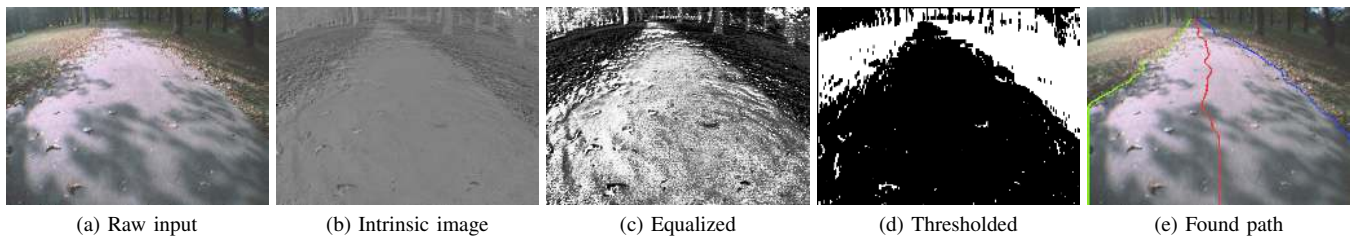


Fig. 3: Path recognition processing stages leading to removal of shadows

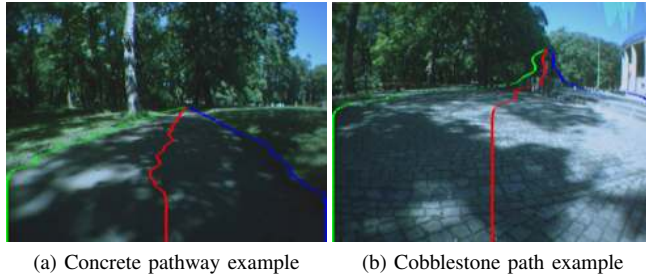


Fig. 4: Example images from the robot's on-board camera during the final experiment.

on-board color camera while moving at speeds about 0.3 m/s. To extend the robot's battery life, the image processing was slowed down artificially to 3 frames per second.

VI. CONCLUSIONS

We have presented a monocular-camera-based system capable of guiding mobile robots along outdoor pathways under adverse lighting conditions. Our method combines the idea of the intrinsic image, which allows for efficient retrieval of material reflectance properties in a wide range of illumination conditions, and a reactive path-following algorithm, which identifies boundaries of the pathway in front of the robot and determines the robot's steering and forward speed. During real-world experiments, the method has proven its ability to eliminate the effect of shadows cast over the traversable terrain, which normally cause the standard visual-based path-identification methods to fail.

Future work will focus on formulation of the intrinsic image calculation for cheap cameras with non-ideal wavelength sensitivity and nonlinear response to illumination. Moreover, we will try to simplify the method to allow its deployment on small robotic platforms with computationally constrained hardware.

REFERENCES

- [1] S. Thrun *et al.*, "Stanley: The robot that won the DARPA Grand Challenge," *Journal of field Robotics*, 2006.
- [2] M. Buehler, K. Iagnemma, and S. Singh, *The DARPA urban challenge: autonomous vehicles in city traffic*. Springer Science & Business Media, 2009, vol. 56.
- [3] C. McManus, W. Churchill, W. Maddern, A. Stewart, and P. Newman, "Shady dealings: Robust, long-term visual localisation using illumination invariance," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 901–906.
- [4] W. Maddern, A. D. Stewart, C. McManus, B. Upcroft, W. Churchill, and P. Newman, "Illumination invariant imaging: Applications in robust vision-based localisation, mapping and classification for autonomous vehicles," in *Proc. of Workshop on Visual Place Recognition in Changing Environments, IEEE International Conference on Robotics and Automation (ICRA)*, 2014.
- [5] K. MacTavish, M. Paton, and T. Barfoot, "Beyond a shadow of a doubt: Place recognition with colour-constant images," in *Proceedings of Field and Service Robotics (FSR)*, June 2015.
- [6] M. Paton, K. MacTavish, C. Ostafew, and T. Barfoot, "It's not easy seeing green: Lighting-resistant stereo visual teach-and-repeat using color-constant images," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2015.
- [7] P. Cristóforis, M. Nitsche, T. Krajník, and M. Mejail, "Real-time monocular image-based path detection," *Journal of Real-Time Image Processing*, pp. 1–14, 2013. [Online]. Available: <http://dx.doi.org/10.1007/s11554-013-0356-z>
- [8] K. Košnar, T. Krajník, and L. Přeučil, "Visual topological mapping," in *European Robotics Symposium 2008*, ser. Springer Tracts in Advanced Robotics, H. Bruyninckx, L. Přeučil, and M. Kulich, Eds. Springer Berlin Heidelberg, 2008, vol. 44, pp. 333–342. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-78317-6_34
- [9] W. H. Huang, M. Ollis, M. Happold, and B. A. Stancil, "Image-based path planning for outdoor mobile robots," *Journal of Field Robotics*, vol. 26, no. 2, pp. 196–211, 2009. [Online]. Available: <http://dx.doi.org/10.1002/rob.20272>
- [10] H. Dahlkamp, A. Kaehler, D. Stavens, S. Thrun, and G. R. Bradski, "Self-supervised monocular road detection in desert terrain," in *Proceedings of Robotics: Science and Systems (RSS)*, 2006.
- [11] O. Miksik, P. Petyovsky, L. Zalud, and P. Jura, "Robust detection of shady and highlighted roads for monocular camera based navigation of UGV," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2011, pp. 64–71.
- [12] C. Rasmussen, "Grouping dominant orientations for ill-structured road following," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2004.
- [13] M. Ollis and A. Stentz, "Vision-based perception for an automated harvester," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 3, Sep 1997.
- [14] G. D. Finlayson and S. D. Hordley, "Color constancy at a pixel," *Journal of the Optical Society of America: Optics, image science, and vision*, vol. 18, no. 2, pp. 253–64, Feb 2001.
- [15] J. A. Marchant and C. M. Onyango, "Shadow-invariant classification for scenes illuminated by daylight," *Journal of the Optical Society of America: Optics, image science, and vision*, vol. 17, no. 11, pp. 1952–1961, Nov 2000. [Online]. Available: <http://josaa.osa.org/abstract.cfm?URI=josaa-17-11-1952>
- [16] G. D. Finlayson, S. D. Hordley, C. Lu, and M. S. Drew, "On the removal of shadows from images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 1, pp. 59–68, Jan 2006.
- [17] G. Finlayson, M. Drew, and C. Lu, "Entropy minimization for shadow removal," *International Journal of Computer Vision*, 2009. [Online]. Available: <http://dx.doi.org/10.1007/s11263-009-0243-z>
- [18] H. G. Barrow and J. M. Tenenbaum, *Recovering Intrinsic Scene Characteristics from Images*. Academic Press, 1978.
- [19] D. W. Scott, "On optimal and data-based histograms," *Biometrika*, vol. 66, no. 3, pp. 605–610, 1979. [Online]. Available: <http://biomet.oxfordjournals.org/content/66/3/605.abstract>
- [20] "Stromovka intrinsic image datasets," [Cit: 2015-06-06]. [Online]. Available: http://purl.org/robotics/intrinsic_road_following