Localization of Pedestrian Lights on Mobile Devices

Kai Rothaus,*, Jan Roters*, and Xiaoyi Jiang*
* Department of Mathematics and Computer Science, University of Münster, Einsteinstrasse 62, 48149 Münster, Germany
E-mail: {rothaus, j_rote01, xjiang}@math.uni-muenster.de Tel: +49-251-83-33759

***Abstract*—In this work a framework is presented to localize and classify pedestrian lights using mobile devices. Our method can be applied as interactive assistance for visually handicapped people to help them passing a pedestrian crossing. Since the computation power and the storage resources of mobile devices are limited the main objective on the localization task is the efficiency of the used computer vision algorithms. The requirement on the traffic light classification is not to miss the crucial red light (reliability). We have designed a prototype for German traffic lights and realized it on a Nokia N95. The presented results demonstrate the efficiency as well as reliability of our method.**

I. Introduction

Our research is motivated by two aspects: (1) the demand of assistance systems to help amaurotic people participating in the all day life and (2) the possibilities of mobile vision, which are offered by modern mobile computing devices equipped with cameras (e.g. smart phones or PDAs with camera).

In this work we present a framework, which are capable for mobile vision devices, to localize and classify traffic lights. These two steps (localization and classification) can be considered as a filter and refinement operation. The main contribution of this paper is the localization task. Hereby, candidate regions (possibly containing traffic lights) are filtered out of a given input (color) image. These regions have to be refined to the one crucial traffic light. Thereby, we take care of two main objectives, namely interactivity and reliability:

- Firstly, the algorithm should perform fast, so that within a split second, the user gets the information if it is safe to pass a pedestrian crossing (with traffic lights) or not.
- Secondly, a false positive feedback of a green light should be avoided in any circumstances.

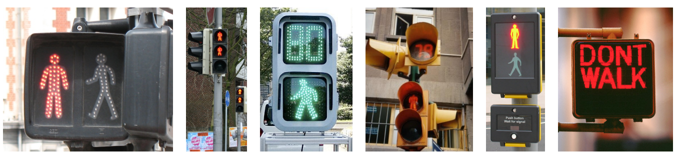
Mobile phones are becoming ubiquitous [1]. According to an UN study in 2007 mobile phone use reaches 50% of the worldwide population. In the Western world there are even more mobile phones than inhabitants. Almost every recent mobile device features an on-board camera. Recently, these devices have attracted notice in the computer vision community and became an active research field. At the first 'International Workshop on Mobile Multimedia Processing' and on ICPR 2008 some interesting works are presented in the field of mobile vision. Liu et al. [2] presented their MobileEye software suite of assistance tools for people with visual disabilities. Wachenfeld et al. [3] used a mobile phone to interpret 1-D bar codes.

At the 'International Conference on Computers Helping People with Special Needs' there have been published some works helping people with visual disabilities. Aranda and Mares [4] presented a system that detects traffic lights. The mobile system 'Crosswatch' [5] helps pedestrians at traffic intersections with zebra crossings to orientate themselves in the correct direction. Other approaches help sightless people in indoor environments (e.g. [6]). Traffic light detection is not only helpful for pedestrians but also an important task for driver assistance systems [7].

The segmentation of the crucial pedestrian lights with a mobile phone is challenging by reason of several aspects:

(a) The **image quality** and the resolution of the capture device are relatively low.
(b) **Computation power** and memory resource are restricted.
(c) Pedestrian lights have **different appearances** in different countries and even for different manufactures (see Fig. 1).
(d) The distance to the pedestrian light could vary between about 4 and 24 meters. Therefore, the **scale of the traffic light** could differ (see Fig. 2(a) and (b)).
(e) The image could has been captured with a non-neglected **rotation** (see Fig. 2(c)).
(f) There could be **many traffic lights** in the image but only one is crucial (see Fig. 2(d)).
(g) Traffic lights can be **temporarily occluded** by vehicles (see Fig. 2(e)).
(h) The **illumination condition** varies between night and sunny weather. Thus, the captured colors of one traffic light depends on the capture time (see Fig. 10).

The remainder of this paper is organized as follows. In the next section, we concretize the external restrictions: (a) low image quality and (b) reduced computational resources. Furthermore, we describe the appearance of pedestrian lights in germany (c). With this knowledge we present (Sec. III) a localization approach, which is robust against scale (d) and rotation (e). The selection of the crucial traffic light (f) will be content of Section IV. In Section V we demonstrate the ability of our approach. The open problems namely occlusion (g) and unknown illumination conditions (h) are part of Section VI. We end up with by discussing the possibilities and restrictions of our approach in Section VII.



(a) (b) (c) (d) (e) (f)

Fig. 1. Pedestrian lights in different countries: (a) France, (b) Germany, (c) Turkey, (d) Japan, (e) UK, and (f) USA

(a) (b) (c) (d) (e) (f)

Fig. 2. Challenges: (a) minimal distance, (b) maximal distance, (c) rotation, (d+e) two traffic lights, and (f) occlusion

II. Problem Specification

The computer vision methods discussed in this paper depend on the features of the mobile vision device. The main idea of the work is to explore good features and to design a framework for detecting pedestrian lights with a mobile vision device, but not to design the best algorithm for all configurations.

In Section II-A we define the conditions for our research. Since we need specific criterion to detect pedestrian lights, we have to restrict ourselves to specific traffic lights. In Section II-B we declare the appearance of German pedestrian lights. For other countries it should be possible to adjust the criterion and to modify our approach. In Section II-C we present our ground-truth database with pedestrian lights of Münster.

A. Prototype Environment

To prove that the detection is possible with a mobile device, a fast and robust system is implemented on a Nokia N95 mobile phone in Symbian C++. It is equipped with a 330 MHz ARM processor, 18 Mb of available RAM and a 5 MP autofocus camera. To get a fast automated approach we use the low resolution video stream (320×240) to compute the semantics. The authors are aware, that with a higher resolution or faster processors the recognition rate could be increased or the computation time could be decreased, respectively. As said, the focus of this work is on presenting a framework and proving the feasibility by a prototype implementation.

B. Appearance of Pedestrian Lights in Münster, Germany

Pedestrian lights could have different appearances in different countries or even cities. To prove our concept, we have trained our image filter on pedestrian lights of our hometown 'Münster' in Germany. It should be possible to develop similar tools for other cities and choose the correct pedestrian light recognition system according to the GPS signal of the mobile phone. Thereby, the presented pipeline can be retained and only some details have to be adjusted.

For the remainder of this paper the following features of a pedestrian light are assumed to be valid preconditions:

1) **Shape:** rectangular with aspect ratio of $1/2$, $1/3$, or $1/4$.
2) **Color arrangement:** at the bottom there is one green light, at the top/middle there are one or two red lights.
3) **Circuitry:** either red or green light is switched on.
4) **Background:** the majority of the traffic light is dark.
5) **Design:** possible shapes of the green or red lights are limited (see Fig. 6)
6) **Installation:** mounted at a vertical pole at a height of approximately 2.15 meter.

C. Database of Pedestrian Lights

We have built up a database holing 501 images at pedestrian crossings with traffic lights.[1] A ground-truth segmentation is given, storing all visible pedestrian lights and also the crucial one. In 309 images the crucial light is red and in 184 images green. In the remaining 8 images there is no crucial light. Therefrom 3 images are without any traffic light.

In 165 images more than one traffic light is visible. Therefrom 7 situation are ambiguous (the crucial light has another visible light in the neighborhood) and in 9 cases a dangerous constellation is present (the crucial light is red, but the next light in the background is green). Overall, in the ground-truth 424 traffic lights are labeled red and 244 are labeled green.

We have divided our database in two disjoint sets. The first set (300 images) is used for training. With the remaining 201 images we verify the performance of our approach.

III. Localization of Pedestrian Lights

In this section we present an approach to localize possible traffic lights in a low resolution image. This approach is robust against the scale and also against rotation (up to some degree).

As mentioned in the last section, traffic lights have specific features (i.e. shape, arrangement, circuitry, design, background, installation). All these features could be used in a special filter algorithm to find traffic light candidates.

Using a parallel combination scheme one can achieve a high accurate recognition rate. Since we want to realize an interactive approach with restricted computational power, we need a smart combination scheme. Note, that it is much faster to verify if a feature is valid for a specific candidate than to inspect all possible image regions according to the special feature. Therefore, we decided to combine the filter algorithm in a sequential architecture (see Fig. 3).

To reduce the search region we have implemented a detector for vertical lines to find the pale, where the traffic light is mounted (Sec. III-A). Our localization procedure uses a red and a green color filter within the search region (Sec. III-B).

[1] Database available under cvpr.uni-muenster.de\research\pedestrianlights

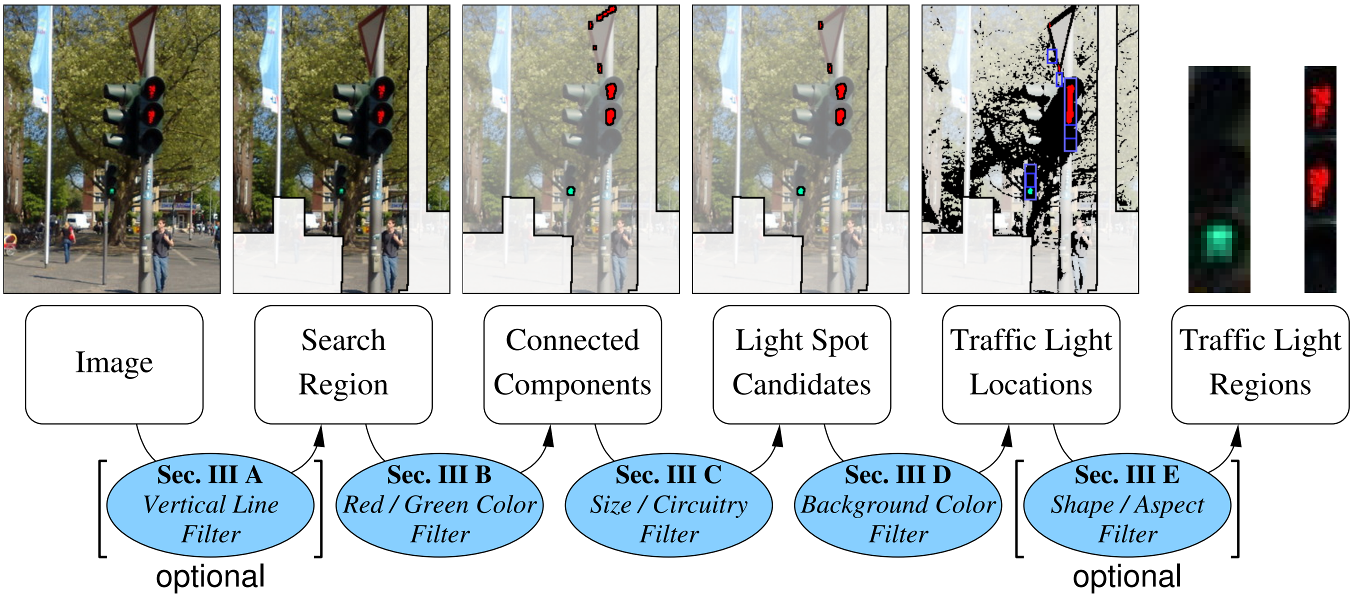


Fig. 3. Sequential combination scheme for localization: (a) input color image, (b) region of no interest brightened, (c) color filter response in green or red, resp., (d) color regions after pruning, (e) dark filter response in black, search region in blue, initial bounding boxes in light blue, (f) localized traffic lights.

After a connected component analysis we compute the size and the circuitry to reduce false positives (Sec. III-C). In Section III-D we explain the next step: the examination of the background color. The optional last step is a shape-based segmentation of the pedestrian light (Sec. III-E).

A. Vertical Line Filter

Given an input image (in which we like to find a pedestrian light) we want to restrict the search region. Since we assume that the pedestrian light is mounted on a pale, we have implemented a Hough-Transformation to detect vertical lines. As parameter space we use the intersection point of the line at the bottom image line and the slope of the line. After a conversion to a gray-scale image and a slight smoothing the course procedure of the line detection is as follows:

1) Detect vertical edges (e.g. with Canny Operator).
2) Apply Hough-Transformation [8].
3) Find local maxima in accumulator.
4) Detect corresponding line segments in the image.
5) Define regions of interest around the line segments.

The quality of this step is reasonable good, but the computation time is a drawback. Even after limiting the number of vertical lines and reducing the number of image operations, the approach performs in general faster without restricting the search area. However, we keep this idea as optional first step in our pipeline, since future mobile computer vision devices might be equipped with faster processors and more memory. In this case a restriction of the search area could be beneficial.

B. Red and Green Color Filter

The most significant feature of traffic lights is the bright color of the lamps. In this step we search for such colors in the region of interest. Therefore, the color of each pixel is checked to fulfill some filter rules. We use the RGB color space, since this is the default color space on mobile vision devices and a conversion to another color space is very time-consuming. Figure 4 shows a plot of green (a) and red (b) traffic light colors, which are extracted from the ground-truth.

In the following we explain how to establish a filter for red traffic lights in three steps: (1) analyze the color distribution of ground-truth, (2) design fast and valuable parameterized filter rules, (3) optimize the parameter.

(1) Analyze the data: One portion of the color samples in Figure 4(b) is distributed along the gray axis of the RGB-cube (one cluster near black and one cluster along the axis itself). Another is located along the red color and the rest of the samples a introduced by noise. So we estimate a Gaussian mixture model in 3D with 4 contributions: black cluster, gray cluster, red cluster, and noise cluster (see Fig. 4(c)). Since the most significant colors to detect red lights should be the red color, we only keep the Gaussian distribution of the red cluster.

(2) Design the filter rules: The Gaussian distribution of the red cluster is defined by its mean color $\mu = (0.48, 0.06, 0.07)$ and the three Eigenvectors v_1, v_2, and v_3 corresponding to the Eigenvalues $\lambda_1 = 0.0590, \lambda_2 = 0.0032, \lambda_3 = 0.0005$.

A color $c = (r, g, b)$ is considered as traffic light color if and only if the following three rules are fulfilled:

$$I_{\text{red}}(c) := c \cdot v_1 \geq th_{\text{red},1} \quad (1)$$
$$(c - \mu) \cdot v_2 \leq th_{\text{red},2} \cdot I_{\text{red}}(c) \quad (2)$$
$$|(c - \mu) \cdot v_3| \leq th_{\text{red},3} \quad (3)$$

That means the red intensity I_{red}, which is the distribution along the dominant axis, should be lower bounded (see (1)). Furthermore, the distance to the red intensity axis along v_2 should be limited toward the gray diagonal (see (2)). The third rule is motivated by the observation that the distribution along v_3 is very tight. More precisely, the distance of c along this direction is thresholded (see (3)).

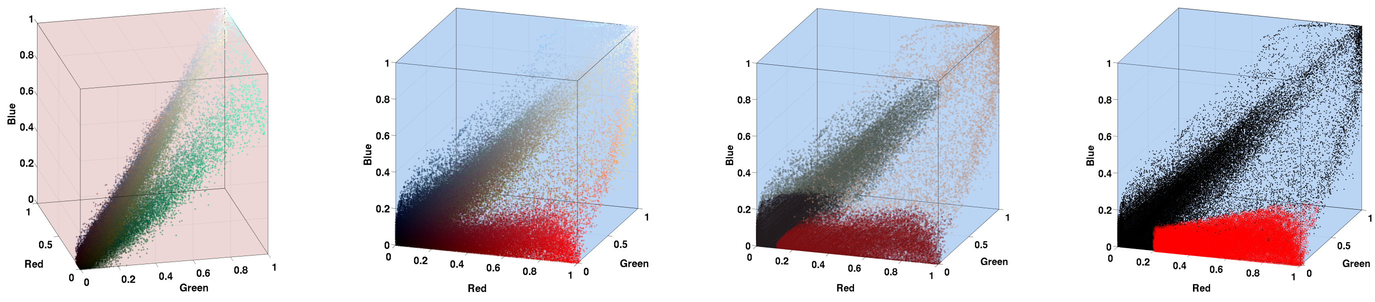


Fig. 4. Green (a) and red (b) traffic light colors from ground-truth. (c) clustering of the red samples, (d) filter for red colors

The resulting red traffic light region in the RGB-cube is wedge-shaped with missing apex. In Fig. 4(d) an example is shown with thresholds $th_1 = 0.20$, $th_2 = 0.25$, and $th_3 = 0.07$.

(3) Optimize parameter: To optimize the parameter we apply the whole process on the training data with different parameter settings and take the best (see Sec. V for details).

The process of designing a color filter for green traffic lights is similar, but instead of using 4 Gaussian Models it is sufficient to use 3. The green parameter are denoted as $th_{\text{green},1}, th_{\text{green},2}, th_{\text{green},3}$.

The responds of the color filters are represented by a binary image. As a post-procession step, we apply a morphological closing and compute the neighbored components.

C. Segmentation using Size and Circuitry

During the last step we have identified pixels, which have the desired color to be part of a traffic light lamp. These pixels are already grouped to connected components.

We assume that the crucial traffic light is between 4 and 24 meters away. In our setting this range corresponds to a width of the traffic light between 2.5 and 15 pixel. These parameters can be utilized to filter out regions which are too small or too huge, by thresholding the size of the connected components.

Furthermore, we know that exclusively the red or the green light is switched on. Connected components featuring red and green pixels as well as vertical neighbored connected components representing a green and a red signal cannot be part of a valid traffic light. All such candidates are refused.

As a post-processing step we melt two red connected components, which are vertically neighbored, since a red light could consist of two lamps.

D. Background Color Filter

The result of the last step are connected components of adequate sizes and colors. We know that the green lamp under a red light is switched off and vice versa. This fact enables us to implement a background filter, which inspects the image region under a red light candidate and above a green light. One can define a search region, where we expect the switched off light. If there are no dark pixels within this appropriate search region, it allows us to refuse this candidate.

In our implementation this filter is simply defined as

$$\mathrm{I}(p) \leq th_{\text{red, dark}} \quad \text{or resp.} \quad \mathrm{I}(p) \leq th_{\text{green, dark}} \tag{4}$$

where $\mathrm{I}(p) = (\mathrm{R}(p) + \mathrm{G}(p) + \mathrm{B}(p))/3$ is the intensity of the pixel p. Furthermore, $th_{\text{red, dark}}$ and $th_{\text{green, dark}}$ are darkness thresholds. The result of this step is a so-called initial bounding box around all traffic light candidates plus search region.

E. Shape-Based Segmentation

We have already localized possible traffic light candidates, by their lamp color, their size, arrangement and background color. In this last step we aim to segment the traffic lights according to their rectangular shapes. Firstly, we assume that the rotation angle of the capture is fairly low (about $\pm 10^o$). A traffic light region should fulfill the following constraints:

1) Traffic light and background are contained.
2) Aspect ratio is between $1/4$ and $1/2$.
3) Many pixels (e.g. 80%) are either light or background.
4) Width of the region lies between 2 to 15 pixel.

To ease the computation we consider axis-parallel rectangular regions only. The task can be modeled by an optimization, like: Find the region of maximal size, which fulfills all constraints.

Even using a suboptimal but fast optimization strategy, this last step decreases the performance so that an interactive application is impossible on our hardware. Furthermore, the computation of the borders is somehow non-robust. Since the profit of this segmentation is negligible compared to the computational costs, we abandon the segmentation step. In future settings the segmentation might be profitable. For instance we need a segmented region for a model-based verification (see Sec. IV-A). Therefore, we keep the segmentation as optional step in our localization pipeline.

IV. Classification of Pedestrian Lights

The localization procedure (Sec. III) results in a set of traffic light candidates $\mathbb{TLC}_1, \ldots, \mathbb{TLC}_k$. In this section we discuss how to refine this candidate set by classification. The features we could use are the position of the traffic light candidate in the image and the pixel color within the corresponding image region. If the segmentation step of the localization pipeline is left out, we use the initial bounding box as segmentation.

We are faced with two problems: (1) Some of the candidates might be false positives, i.e. other objects than traffic lights. (2) Under all traffic lights in the image we have to find the crucial one for the pedestrian. These problems should be solved during two different steps: verification and selection (see Fig. 5).

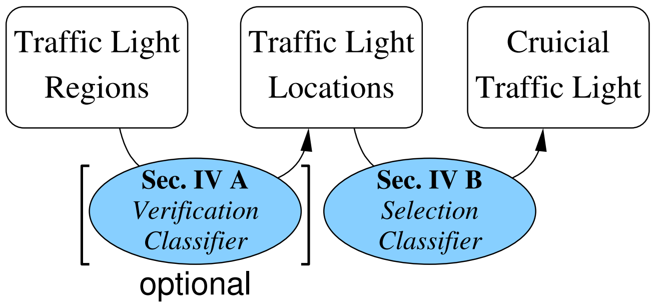


Fig. 5. Sequential combination scheme for classification

Fig. 6. Templates for the graphic design of our pedestrian lights

In Section IV-A we present the idea of a model-based verification to classify the candidates in correct traffic lights and false positives. Thereafter, we describe how to select the traffic light, which is crucial for the pedestrian (Sec. IV-B).

A. *Model-based Verification*

We have assumed that the design of the traffic lights is restricted. In Fig. 6 all possible designs for a pedestrian sign in Münster are presented. The obvious way to verify, if a traffic light candidate shows a correct traffic light can be realized by a template matching approach. We suggest the following steps:

1) Normalization of the image region width.
2) Transformation to a gray-scale image.
3) Matching of the design templates in the modified region.

We are planning to investigate the usefulness of such an approach. According to computational restrictions the problems are (1) the low resolution and (2) that the traffic light borders are not detected. In this situation a template matching seems not to be promising. However, if the computational power of smart phones increases, it could be profitable to reduce the false-positive error by a model-based verification approach.

B. *Selection of the Crucial Light*

By reason of the perspective, the important traffic light should be the biggest and highest of all traffic lights in the image. These two simple criterion are used to select the crucial traffic light. More precisely, we report a traffic light candidate $\mathbb{TLC}_i$ as crucial if all of the following constraints are true:

- $\mathbb{TLC}_i$ is the broadest traffic light
- $\mathbb{TLC}_i$ is the highest traffic light
- No other traffic light has a similar height than $\mathbb{TLC}_i$

The color of such a traffic light $\mathbb{TLC}_i$ is obvious since the region contains exactly one type of traffic light color, either red or green. There could be different failures. The catastrophic error is, that a green light is reported during a red phase. Reporting no traffic light or a false red report are errors which abridge the convenience but not affect the user's security.

V. Experiments and Results

Our algorithm depends only on 8 main parameter, 4 in each case (red or green light, resp.). These two parameter groups are optimized separately. In our experiments we subsample each parameter space and test 10.000 parameter settings. With our ground-truth, we measure the quality of the setting by counting the number of correctly detected traffic lights (TP), falsely detected traffic lights (FP) and missed traffic lights (FN).

In the following we optimize the parameter groups for red (Sec. V-A) and subsequently for green (Sec. V-B) traffic lights. Using the optimized parameter we present the performance of the classification step in Section V-C. Thereafter, we validate the results on our validation set (Sec. V-D) and discuss some selected results (Sec. V-E). Finally, a brief investigation of the rotational robustness (Sec. V-F) is presented.

A. *Optimize Parameter for Red Traffic Lights*

The missing of a red sign could cause serious problems. So our optimization criterion is to maximize the precision with a bounded miss rate. Fig. 7 (a) shows the performance of the investigated red parameter settings. We claim a recall

$$R = TP/(TP + FN) \tag{5}$$

of at least 75% and choose the setting with the best precision

$$P = TP/(TP + FP). \tag{6}$$

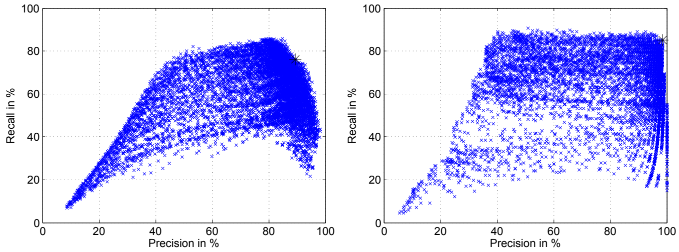
The result of our optimization are the parameter $th_{\text{red},1} = 0.3$, $th_{\text{red},2} = 0.15$, $th_{\text{red},3} = 0.028$, $th_{\text{red, dark}} = 0.19$, With a recall of 76.0% a precision of 89.5% is achieved. This optimized performance is visualized as a black asterisk in the Fig. 7 (a).

B. *Optimize Parameter for Green Traffic Lights*

The optimization of the green parameter set depends on a bounded precision. The precision equals 100% if and only if we have detected no false green light. We allow at most 1.5% FP (i.e. $P \geq 98.5\%$) and choose the parameter vector yielding the best recall. Fig. 7 (b) shows the performance of the investigated green parameter settings. The best thresholds of the green filter are: $th_{\text{green},1} = 0.2$, $th_{\text{green},2} = 0.15$, $th_{\text{green},3} = 0.05$, $th_{\text{green,dark}} = 0.19$. With these parameter we achieve a recall of about 85.0% (see black asterisk in Fig. 7 (b)).

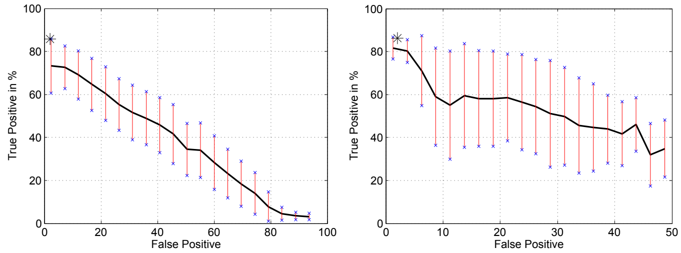
C. *Performance of Classification*

The optimization depends on all visible traffic lights in the scene. The performance for detecting the crucial traffic light is presented in Fig. 8 using ROC-curves. Here, the true positive rate is plotted against the number of false positives. Furthermore, the standard deviation is visualized by the vertical lines. Our optimized parameter (the black asterisk) lead to a stable recognition of the crucial traffic light. As desired the number of false positives are very small in the case of green light



(a) (b)

Fig. 7. Recall and Precision for (a) red and (b) green traffic lights



(a) (b)

Fig. 8. ROC-curve for detecting the crucial (a) red and (b) green traffic light

detection. We report in 2 cases a wrong crucial green light (precision of 98.1%) and keep a recall (i.e. true positive rate) of 86.3%. The performance of the red traffic light detection is similar: we classify in 4 cases false red traffic lights (precision of 97.4%) and achieve a recall of 86.3%.

D. *Validating the Results*

As mentioned we have a validation set of 201 images, which are not used during the parameter optimization. We fixed the parameter and applied the approach on this validation set. For red traffic lights we yield a precision of 96.5% and a recall of 83.3%. The precision for green traffic lights is 98.3% and the recall is 90.8%. We report 5 wrong crucial traffic lights and falsely report no traffic light in 28 of the verification images. This corresponds to an overall miss rate of 16.4%.

E. *Example Results of our Approach*

Fig. 9 depicts some results produced with our approach. Thereby, we put a white frame around all traffic light candidates and an additional blue frame around the reported crucial one. In the first row (a-d) perfect recognitions are presented, even in dark illumination conditions (a), bright traffic light color (b), objects in the front (c) or rainy weather (d).

Sometimes noisy objects are detected as traffic light candidates (Fig. 9(e-h)). Objects on small vehicles normally cause no problem (e), since they are almost not green and much below the traffic light. Some objects on trees (f) or buildings (g) could be identified as traffic light candidates, too. This situation is much more difficult, since the objects may be placed above the traffic light. A template matching could decrease such false positives. Currently, template matching is not integrated in our system. Another situation in which an additional template matching step could be helpful are transversely mounted street traffic lights (see Fig. 9(h)).

However, there are some limitation, which we present in the third row of Fig. 9(i-l). If traffic lamps are capture with low saturation (see (i) and (j)) the traffic light could be missed. Sometimes big vehicles occlude the traffic light (k), or the scene is contradictory (l).

Some problems (e.g. (h) and (l)) are introduced by a poor perspective angle and can be corrected by changing the viewpoint. This is shown in the last row of Fig. 9. In the next Section we discuss an extension on the video stream, which reduce the effect of poor perspective.

F. *Investigating the Rotational Robustness*

Currently, we are investigating the rotational robustness of our approach. First experiments show that a rotational angle of $\pm 10^o$ only slightly affects the performance of our approach. For these tests we have rotate the images in both directions and report the angular range in which the result keeps stable. Including all images in this test scenario, we can identify 328 (i.e. 73.1%) of the red and 206 (i.e. 84.4%) of the green traffic lights with an optimal rotation. If the images are rotated by maximal $\pm 10^o$, we recognize 254 red and 180 green traffic lights. This means, the localization keeps stable for 77.4% red and 87.4% green lights in comparison to the optimal rotation.

VI. EXTENSIONS

The major drawback of our approach is the high miss rate of 16.4%. This is influenced by temporary occlusions and inappropriate illumination conditions. The reason of the last may be versatile, but the effect is that the captured colors are falsified by an over- or an underexpose of the lights.

In the following we discuss how to deal with these challenges. The problem of occlusion can be solved by video processing (Sec. VI-A). It increases the recognition rate and makes the approach more robust against slight illumination changes. In Section VI-B we discuss the effect of unknown illumination and present an idea of resolving the problem.

A. *Video Processing*

Yet, we have only presented techniques on images. In the following we present two techniques which utilize the video stream to reduce the high miss rate. The first technique is to repeat the procedure in case of no response. Secondly, tracking of the crucial object could be a useful method to recognize and correct a false response.

1) Repeat the Procedure: Missing the traffic light in an image could have different reasons. In all situations it might be helpful, to try it again with a later image.

Reason 1, "traffic light is occluded by big vehicles": Since in the situation of no detected traffic light our approach would never give the command to pass the crossing this constellation is handled correct.

Reason 2, "traffic light colors falsified by wrong exposure": By moving the camera and repeating the localization and classification the traffic light might be found.

Reason 3, "scene is contradictory": In this situation two traffic lights are located close to each other (see Fig. 2 (e)).

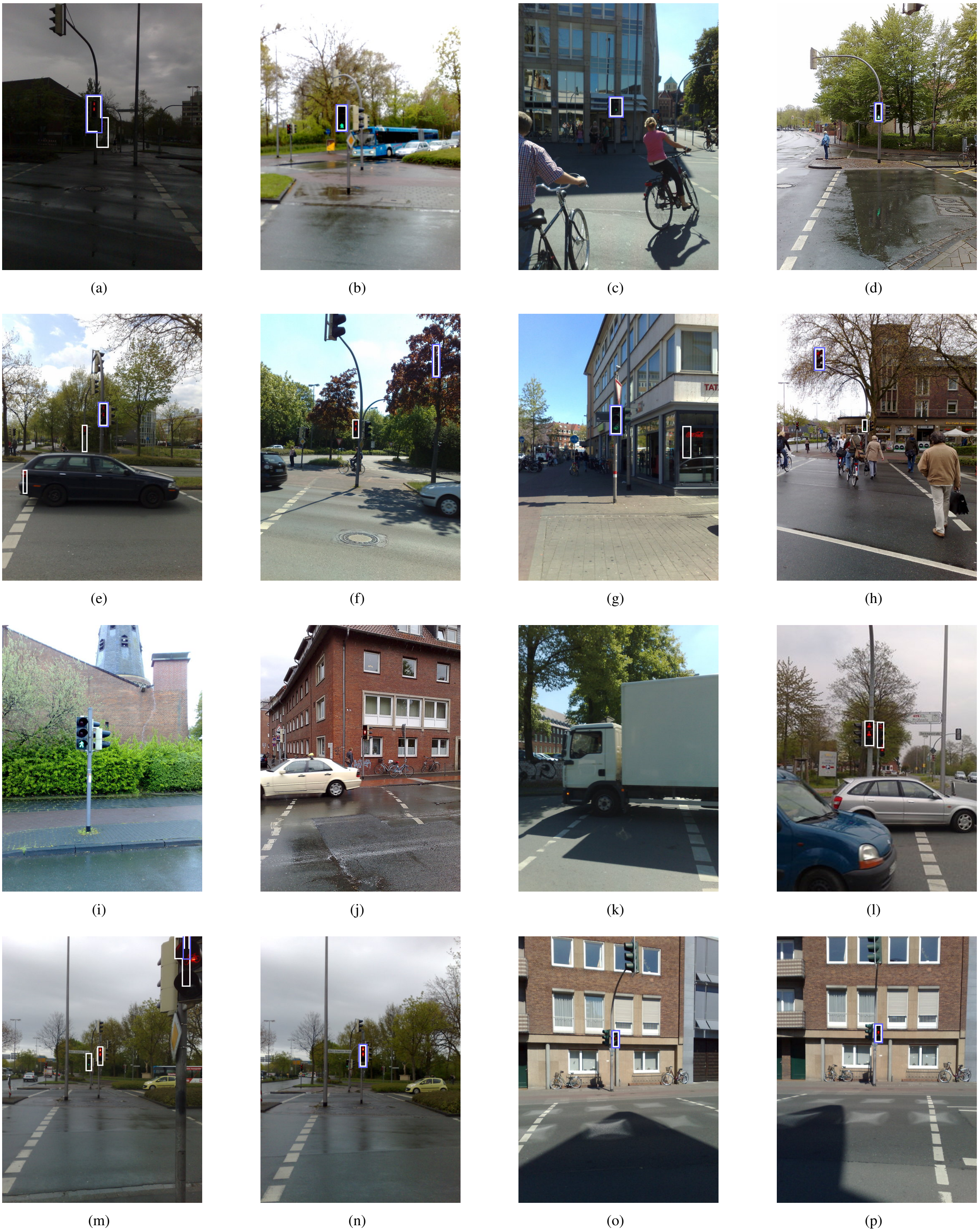


Fig. 9. Results of our approach. (a-d) perfect results, (e-h) noisy objects, (i-l) no traffic light reported, (m-p) change of perspective

(a) (b) (c)

Fig. 10. Difficult illumination: (a) dusk, (b) frontlighting, and (c) night

By changing the perspective the green light for the street lane getting out of sight, so that a decision is possible.

2) Tracking: Another powerful tool is the tracking of the crucial traffic light. By tracking the main object some misclassification could be corrected. This helps to avoid the detection of traffic light similar objects on moving objects (e.g. lights of a car, truck painting). By tracking the cataclysmal object over time the catastrophic error of falsely reporting a green sign, can be decreased.

In our implementation we have realized both, repeating the procedure in case of no response and tracking. Currently, we are working on computing the benchmark of these extensions. First tests demonstrate, that the video processing techniques increase the user's convenience. Under appropriate illumination conditions the system reports the correct traffic light in almost all situations so that the miss classification rate in a real world application is much lower than 16.4% (on images). Thereby, a delay of at most 2 seconds is acceptable.

B. Illumination Robustness

Fig. 10 depicts situations, where the illumination condition is enduring inappropriate (dusk, frontlighting, night). Other examples are heavy snow, fog, or rain. The problem in all such situations is that the colors are falsified and some features are not captured in the image anymore. For instance the contrast of the traffic light frame and pale against the background are much lower than in normal conditions. Furthermore, in dark scenes bright lamps could cause a halo or other artificial objects. To reach illumination robustness, one has to account on such phenomenons.

Our current prototype cannot handle such situations of poor visibility in a stable manner. However, the example in Fig. 9 (a) demonstrates that it is possible to detect the crucial light even under challenging conditions. Therefore, it should be possible to achieve a good recognition rate under unknown illumination conditions by the following extension:

1) Classify the illumination conditions. (e.g. in normal light, night, dusk, frontlighting, snow, fog, rain).
2) Select an adequate traffic light recognizer depending on illumination.
3) Recognize traffic light with the selected method.

VII. Discussion and Conclusions

In this paper we have presented a framework for detecting traffic lights on a mobile vision device. Thereby, we have accomplished several challenges: low image quality, poor resolution, restricted computation power and memory resource, scalability, rotational robustness, many traffic lights and last but do not least temporarily occlusion. We have presented different features for detecting pedestrian lights and demonstrate the way they could been used in a prototype implementation.

This prototype runs on a Nokia N95 smart phone and are designed to detect German standard pedestrian lights. The results are very promising in normal illumination condition. The highest influence for failures is the falsification of colors in the case of poor visibility. We have increased robustness by the use of two video stream procession techniques, but in extreme situations our approach fails, yet.

With our work we have shown, that smart phones can be used as powerful tools for visually handicapped people. For a pre-commercial development some further work has to be done. Most essential is to avoid false green light reports. For this, template matching should be applied on the magnified region, which is reported as crucial light.

Furthermore, the robustness of our approach has to be investigated more precisely. The experiments should be extended to the application of our tool in a real world scenario. The open problem is that it is not possible to apply our approach on a mobile phone and simultaneously record this process (for evaluation of the result). Note, that the tool runs on our setting in an interactive mode. The crucial light is initially found in much less than a second and tracking is possible with 5 to 10 frames per second.

More work has to be spent on extending the approach to other devices, cities and states. Hereby, complexer environments (more people, more commercial lights, bad weather) should be taken into account. Additional filter rules should be developed in the case of difficult illumination conditions.

Beside these open works, we have proved that our image analysis pipeline is suitable for localizing pedestrian traffic lights. Our prototype on a mobile vision device is capable to localize and track traffic lights in an interactive mode.

References

[1] R. Ballagas, J. Borchers, M. Rohs, J. G. Sheridan, The smart phone: A ubiquitous input device, IEEE Pervasive Computing 5 (1) (2006) 70–77.

[2] X. Liu, A camera phone based currency reader for the visually impaired, in: Assets '08: Proc. of the 10th Int. SIGACCESS Conf. on Computers and Accessibility, ACM, New York, NY, USA, 2008, pp. 305–306.

[3] S. Wachenfeld, S. Terlunen, X. Jiang, Robust recognition of 1-d barcodes using camera phones, in: ICPR, 2008, pp. 1–4.

[4] J. Aranda, P. Mares, Visual system to help blind people to cross the street, in: ICCHP, 2004, pp. 454–461.

[5] V. Ivanchenko, J. Coughlan, H. Shen, Crosswatch: A camera phone system for orienting visually impaired pedestrians at traffic intersections, in: ICCHP, 2008, pp. 1122–1128.

[6] R. Manduchi, J. Coughlan, V. Ivanchenko, Search strategies of visually impaired persons using a camera phone wayfinding system, in: ICCHP, 2008, pp. 1135–1140.

[7] T.-H. Hwang, I.-H. Joo, S. I. Cho, Detection of traffic lights for vision-based car navigation system, in: PSIVT, 2006, pp. 682–691.

[8] P. Hough, Method and means for recognizing complex patterns, 1962.