Robust Detection of Shady and Highlighted Roads for Monocular Camera Based Navigation of UGV

Ondrej Miksik, Petr Petyovsky, Ludek Zalud and Pavel Jura

Laboratory of Telepresence and Robotics & Computer Vision Group, Faculty of Electrical Engineering and Communication, Brno University of Technology, Brno, Czech Republic

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

Introduction

- Our robotic research group works on Orpheus-AC military reconnaissance mobile robot (e.g. Czech Army)
  - The robot is a part of an armored vehicle for chemical, nuclear and biological contamination measurement
  - Primary task, is to make the measurement and identification in areas with the highest risk of massive contamination
  - Although the robot is primarily teleoperated, we are working on autonomous functions, that will help the user to achieve higher universality and reliability in missions
- Various missions
  - Goods delivery in dangerous regions (e.g. IEDs, ...)
  - Automatic return from teleoperated mission in case of signal loss
  - Autonomous navigation of military robots
- System demands
  - Diverse light conditions (direct sunlight, strong shadows, ...)
  - Structured and unstructured roads (gravel, tarmac, ...)
  - No additional sensors (easy-to-decontaminate)
- Vision system design
  - A fusion of the vanishing point estimation and texture segmentation
  - Vanishing point determines the training area
  - Road color models are constructed from sample pixels by self-supervised learning algorithm and adaptively updated
  - A few simple rules define properties of the color segmentation system (adaptivity speed, selectivity, robustness or behavior in shady and/or overexposed highlighted road segments)

Figure 1: Output of our system

Vanishing Point Estimation

1. Texture flow estimation
   - A bank of self-similar Gabor wavelets

$$\hat{g}_{odd}(x, y, \theta, \lambda) = \exp\left(-\frac{1}{8\sigma^2}(4a^2 + b^2)\right)\sin\left(\frac{2\pi a}{\lambda}\right), \qquad (1)$$

where $x = y = 0$ is the kernel center, $\sigma = \frac{k}{9}$, $k = \frac{10\lambda}{\pi}$, $\lambda = 2^{\log_2(l_w)-5}$, $a = x\cos(\theta) + y\sin(\theta)$, $b = -x\sin(\theta) + y\cos(\theta)$ and "sin" changes to "cos" to obtain even kernel

   - Subtract $\hat{g}$'s DC component and normalize kernel's coefficients, so that norm $L^2 = 1$
   - Compute a square norm of so-called Gabor energy of each of n (e.g. $n = 36$) evenly spaced Gabor filters between 0° and 180°

$$\mathbf{E}(\theta, \lambda) = [(\hat{g}_{odd}(x, y, \theta, \lambda) * I(x, y)]^2 + [(\hat{g}_{even}(x, y, \theta, \lambda) * I(x, y)]^2 \qquad (2)$$

where $*$ denotes convolution (efficient computation – FFTW)

2. Vanishing point voting
   - The dominant orientation at pixel $\mathbf{p}(x, y)$ is chosen as the filter orientation which elicits the maximum Gabor energy at that location

$$\theta_{max} = \arg\max_{\theta} \mathbf{E}(\theta, \lambda). \qquad (3)$$

   - Let the angle of the line joining an image pixel $\mathbf{p}$ and a vanishing point candidate $\mathbf{v}$ is $\alpha(\mathbf{p}, \mathbf{v})$, then $\mathbf{p}$ votes for $\mathbf{v}$ if the difference between $\alpha(\mathbf{p}, \mathbf{v})$ and $\theta_{max}(\mathbf{p})$ is within the dominant orientation estimator's angular resolution

$$vote(\mathbf{p}, \mathbf{v}) = \begin{cases} 1 & \text{if} |\alpha(\mathbf{p}, \mathbf{v}) - \theta_{max}(\mathbf{p})| \leq \frac{\gamma\pi}{n}, \\ 0 & \text{otherwise.} \end{cases} \qquad (4)$$

   - Objective function for each vanishing point candidate $\mathbf{v}$ is

$$votes(\mathbf{v}) = \sum_{\mathbf{p} \in R(\mathbf{v})} vote(\mathbf{p}, \mathbf{v}), \qquad (5)$$

where $R(\mathbf{v})$ is a voting region

   - Smoothing filter CONDENSATION throughout the whole sequence is employed to reduce the influence of noise and to avoid the jumpy characteristics of output

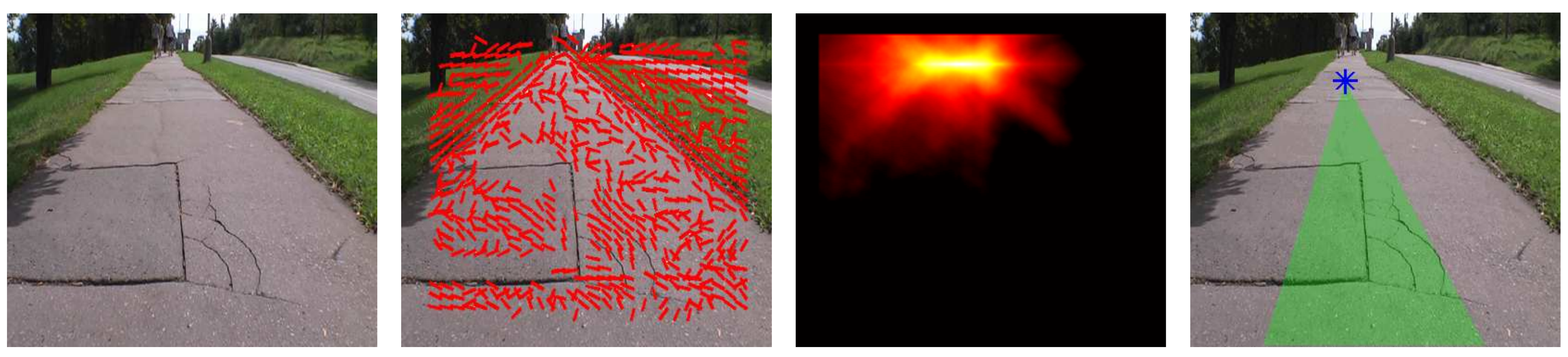

Figure 2: Vanishing point estimation – input (a), texture flow (b), voting (c), output (d)

Results - Adaptivity and Robustness

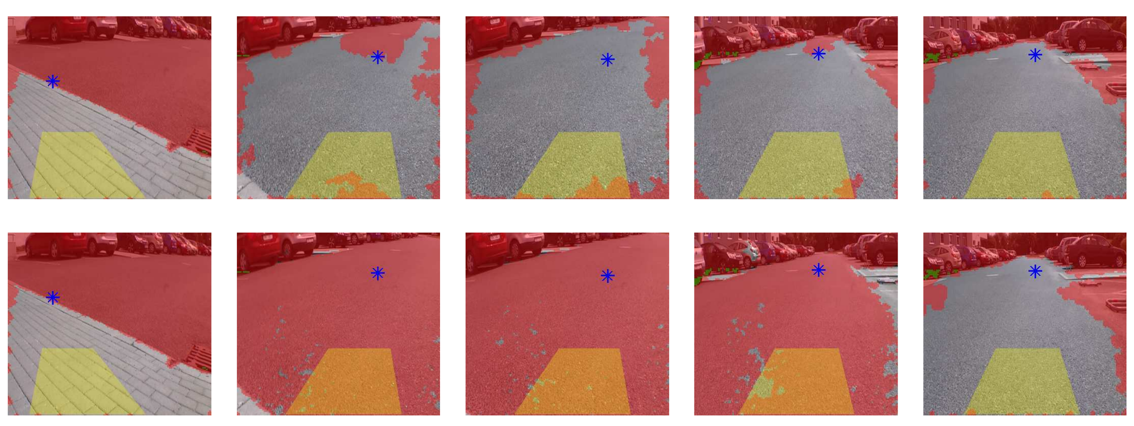

Figure 3: AWU and DF (top) and without (bottom)

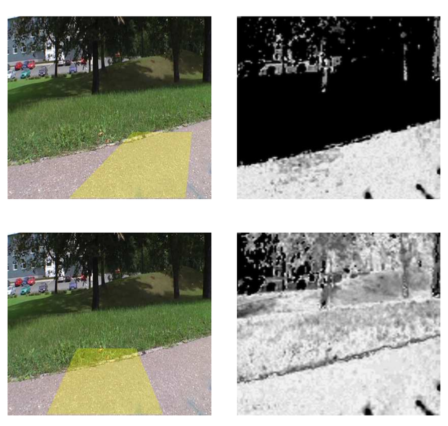

Figure 4: Training area

Results

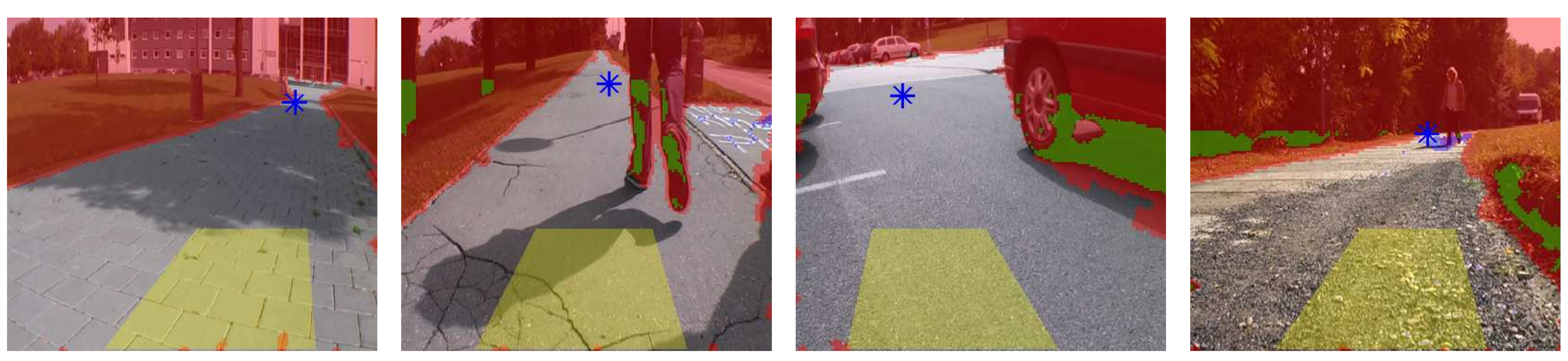

Figure 5: Output of the proposed algorithm

Road Extraction

1. Training area
   - By comparison with previously published algorithms, the training area is determined by the estimated vanishing point
   - Non-static training area prevents to learning of outliers if the robot is close to the road borders (cf. Fig. 4)

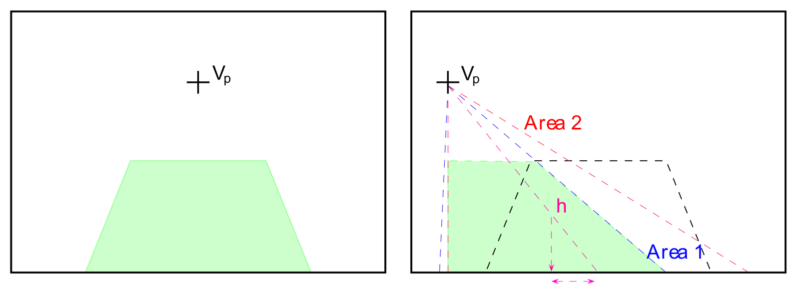

Figure 6: Training area in its default position (left) and shifted training area (right)

2. Gaussian Mixture Model – construction
   - Hierarchical agglomerative (bottom-up) k-means clustering instead of EM (lower computational complexity and sufficient accuracy)
   - Number of models c are not fixed to some value, but is adaptable with the different types of road surface
   - Each cluster c is represented by its mean vector $\boldsymbol{\mu}$, covariance matrix $\boldsymbol{\Sigma}$ and a mass m (number of pixels associated to the cluster)

$$\boldsymbol{\mu}_c = \frac{1}{n_c}\sum_{i=1}^{i=n_c} \mathbf{p}_{c,i}, \qquad \boldsymbol{\Sigma}_c = \frac{1}{n_c}\sum_{i=1}^{i=n_c} \mathbf{p}_{c,i}\mathbf{p}_{c,i}^{\mathrm{T}} - \boldsymbol{\mu}_c\boldsymbol{\mu}_c^{\mathrm{T}}, \qquad m_c = n_c \qquad (6)$$

3. Gaussian Mixture Model – update
   - In addition to c training models, n_l learned models represent "history of the road"

$$(\boldsymbol{\mu}_L - \boldsymbol{\mu}_T)^{\mathrm{T}}(\boldsymbol{\Sigma}_L + \boldsymbol{\Sigma}_T)^{-1}(\boldsymbol{\mu}_L - \boldsymbol{\mu}_T) \leq d_{similar}, \qquad (7)$$

   - Exponential forgetting – if the training model overlaps any learned model, the learned model is updated according to formulas

$$\boldsymbol{\mu}_{updated} = \frac{m_L\boldsymbol{\mu}_L + m_T\boldsymbol{\mu}_T}{m_L + m_T}, \quad \boldsymbol{\Sigma}_{updated} = \frac{m_L\boldsymbol{\Sigma}_L + m_T\boldsymbol{\Sigma}_T}{m_L + m_T}, \quad m_{updated} = m_L + m_T \qquad (8)$$

4. Shadows and overexposed highlights
   - Among a shady and/or overexposed highlighted road segments, models with the same original color could be easily discarded after a few frames
   - Thus, the models with high mass are compared with those with low mass
   - If the mean colors of those models are similar, the mass of small models is adjusted to above some value
5. Preprocessors
   - Important in situations when a robot is not yet among the shadows/highlights
   - Without preprocessors, a huge dark shadows/overexposed highlights will be labeled as a non-road
6. Adaptivity and robustness
   - Integral character of a mass updating formula: similar problem in feedback control theory – anti-windup (AWU)
   - A decay factor (DF) is taken off from each model to remove models which were not updated for many frames (cf. Fig. 3)
7. Measurement and thresholding
   - A "roadness" score is measured as a minimum of Mahalanobis square norm between each pixel and learned models

$$D(\mathbf{p}, \boldsymbol{\mu}_i) = \min_i((\mathbf{p} - \boldsymbol{\mu}_i)^{\mathrm{T}}\boldsymbol{\Sigma}_i^{-1}(\mathbf{p} - \boldsymbol{\mu}_i)) \qquad (9)$$

   - Next, it is possible to use these values as an input of probabilities to some higher AI
   - Thresholding, postprocessing (morphology), only blob connected with the training area by flood fill, is preserve as a road region, others are discarded as non-road

Conclusions

- No additional sensor or difficult calibration
- The fusion of two different approaches leads to better robustness
- Tested on a number of different sequences (> 10 000 images)
- Various environments and robustness to difficult illumination conditions
- AWU and Decay Factor influence adaptivity speed

Appeared in *Proceedings of the IEEE ICRA 2011* ondra.miksik@gmail.com http://www.miksik.co.uk