Rapid Vanishing Point Estimation for General Road Detection

Ondrej Miksik

Abstract— **This paper deals with fast vanishing point estimation for autonomous robot navigation. Preceding approaches showed suitable results and vanishing point estimation was used in many robotics tasks, especially in the detection of ill-structured roads. The main drawback of such approaches is the computational complexity – the possibilities of hardware accelerations are mentioned in many papers, however, we believe, that the biggest benefit of a vanishing point estimation algorithm is for primarily tele-operated robots in the case of signal loss, etc., that cannot use specialized hardware just for this feature. In this paper, we investigate possibilities of an efficient implementation by the expansion of Gabor wavelets into a linear combination of Haar-like box functions to perform fast filtering via integral image trick and discuss the utilization of superpixels in the voting scheme to provide a significant speed-up (more than 40 times), while we loose only 3-5% in precision.**

I. INTRODUCTION

During the past few decades, the robotics community has made great efforts in developing autonomous robots. One of the biggest challenges are definitively reliable perception. Many papers have been published about vision-based navigation. By contrast with other sensors like laser range finders [1], [2] or radars [3], vision-based navigation is cheap, does not require any additional sensor, since many robots are already equipped with a camera (for telepresence), and suitable for long-term sensing, which is usually necessary for high-speed driving.

Early systems usually deal with structured roads. The well-known and mature projects were developed in Carnegie Mellon University (CMU)'s Navlab [4], [5], that use a number of Gaussian color models to represent road and non-road surfaces (*UNSCARF*, *SCARF*) and deal with both, intersections and shadows, however it require some overlapping between the frames. Another projects like *ARGO* from the Università di Parma [6], [7] or a more recent system proposed by Dong-Si et al. [2] use stereo vision. Approaches, that use optical flow estimation [8], [9] do not work well on chaotic roads, when the camera is unstable and the optical flow estimation is not robust enough. One of the most powerful systems was developed by Stanford AI Lab – vehicle *Stanley* uses self-supervised learning based on a combination of laser range finders and camera [1], however this combination is not useful for robots of a smaller size.

Another branch of research is represented by approaches that estimate the so-called *Vanishing Point* (VP) of the road. Original paper by Rasmussen [10] investigates grouping of dominant orientations of a texture flow, that is suitable for unstructured, or ill-structured roads with no significant borders. The algorithm consists of two stages: (1) estimation of dominant orientations by a bank of Gabor wavelets, and (2) a voting scheme, which is used to determine the most likely coordinates of the road's vanishing point.

The Rasmussen's approach works well in an outdoor environment with ill-structured roads which are barely visible even for humans. The above mentioned algorithm does not require any a priori knowledge about the road surface, difficult classifier training, etc. It provides information about the correct course for robot navigation, however the main drawback is, the lack of information about the free space ahead of robot. The later refinement employs laser range finders to deal with obstacles [11].

Another paper by Kong et al. [12], [13] proposed the idea of a locally adaptive soft voting scheme to prevent tending to favour points that are high in the image, which sometimes leads to large errors in the estimation of the vanishing point. The second important part of these papers discuss road segmentation by an Orientation Consistency Ratio and the two most dominant edges.

More recently, Miksik et al. [14] proposed a fusion of the frequency based estimation of a vanishing point and probabilistically based texture segmentation. The texture segmentation utilizes the continuously re-learned Gaussian Mixture Model, which is used to determine the free road ahead of the robot. The main advantage of this approach is, that the road can be detected even if the vanishing point estimation fails and does not require lidars.

And finally, an approach published by Qi et al. [15] is similar - an example-based global image matching method is used to get an approximate idea of clear path candidate regions, and a Gaussian Mixture Model, models local image patches to further improve the clear path detection.

It is obvious, that there are a bunch of papers and promising work, that utilizes reliable Vanishing Point estimation. On the other hand, none of the above mentioned papers discuss the crucial issue for mobile robotics: "How to estimate the VP in real-time?", because the original algorithm is quite close to real-time, but not close enough[1]. It is possible to use massive computational power of specialized DSP, FPGA or GPGPU, however, we strongly believe, that the main domains of such a guide-path following algorithms are for e.g. primarily teleoperated robots that are able to return in the case of signal loss, a swarm of cheap robots with assisted

The author is with the Department of Control and Instrumentation, Faculty of Electrical Engineering and Communication, Brno University of Technology, Kolejni 4, Brno, Czech Republic
ondra.miksik@gmail.com
http://www.miksik.co.uk

[1] Rasmussen uses Nvidia GeForce 6800 to accelerates voting.

autonomy, … In these cases, it is usually inconvenient (price) or even impossible (weight) to use additional hardware. To our best knowledge, we are the first in this paper, who aim at computational efficiency, instead of just a precision of the VP estimation algorithm. In this paper, we propose a method, which achieves results comparable to the above mentioned, however our method is significantly faster without any dependency on a specific hardware platform.

The remainder of this paper is organized as follows. The texture flow estimation process is discussed in section II, section III describes the refined voting scheme, section IV provides performance evaluation and the paper is concluded in section V.

II. TEXTURE FLOW ESTIMATION

The first step of a vanishing point estimation algorithm is the estimation of a *texture flow*. The dominant orientation $\theta(\mathbf{p})$ of an image at pixel $\mathbf{p}(x,y)$ describes the strongest local parallel structure or texture flow. There exist various techniques, which can be used for estimation of dominant orientation, involving usage of Gaussian pyramids with principle component analysis, steerable filters, etc. We follow the line of research that investigates grouping of dominant orientations, that are estimated by a bank of 2D Gabor wavelet filters, since they are known to be accurate [10], [16], [17]. The kernels of Gabor wavelet filters are quite similar to the 2D receptive field profiles of the mammalian cortical simple cells and show suitable characteristics of spatial locality and orientation selectivity [18].

Gabor wavelets consist of a product of an elliptical Gaussian and a complex plane wave, which determines the frequency and the phase content of the local parts of a signal as it changes over time. All kernels are constructed from one mother wavelet by its dilation and/or rotation, which means, that the wavelets are self-similar.

The set of $k \times k$ Gabor kernels for an orientation θ, radial frequency in radians per unit length ω and odd or even phase are defined by

$$\psi(x,y,\theta,\omega) = \frac{\omega}{\sqrt{2\pi}c} e^{-\frac{\omega}{8c^2}(4a^2+b^2)} \left(e^{ia\omega} - e^{\frac{c^2}{2}} \right), \tag{1}$$

where $x = y = 0$ is the kernel center. Next, a and b are defined by

$$\begin{aligned} a &= x\cos(\theta) + y\sin(\theta), \\ b &= -x\sin(\theta) + y\cos(\theta), \end{aligned} \tag{2}$$

and $c = 2.2$, $\omega = \omega_0 \times 2^s$, $\omega_0 = 2.1$ and $s = \{0,1,2,3,4\}$. More details about 2D Gabor wavelets are given in [17].

Although it is often mentioned, that the response of the bank of filters might be efficiently computed by FFT, it was observed, that such computations are not fast enough due to the large number of filters in the bank (consider commonly used 36 orientations and 5 scales) [19]. Thus, we need to optimize this computational routine by a widely used trick with integral images, which is well-known especially in the face recognition domain [20].

Integral images represent convenient data structure widely used in computer vision, since computational complexity of Haar-like box filters with integral images is independent on the size of the kernel. Integral image can be computed efficiently with a recursive implementation in a single pass over the image as

$$I(x,y) = i(x,y) + A + I(x,y-1) \tag{3}$$

where I is the integral image, i is the input image and A is the accumulate of pixels in the current row. Once the integral image is precomputed, the sum of a rectangular area of any possible size can be computed in a constant time by only three additions

$$\sum_{x'=(left-1)}^{right} \sum_{y'=(top-1)}^{bottom} i(x',y') = I_D + I_A - I_B - I_C, \tag{4}$$

where A is top-left, B is top-right, C is bottom-left and D is the bottom-right corner.

It is necessary to decompose the Gabor filters into a linear combination of Haar-like box filters to perform filtration in an integral image domain. First, we define the dictionary $\mathbf{D} = \{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_N\}$ of Haar-like box filters, where each $\mathbf{b}_i$ is a column vector formed by reshaping the rectangular box function (defined below). The dictionary consists of single Haar-like box filters which can be formally written as binary functions

$$h_{single}(u,v) = \begin{cases} 1 & u_0 \le u \le u_0 + w' - 1 \\ & v_0 \le v \le v_0 + h' - 1 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where $[u_0, v_0]$ are the coordinates of the top left corner and w' and h' are the size of the white box. Gabor wavelets are (anti)symmetric, therefore we also use a vertically symmetric box function

$$h_{symmetric}(u,v) = \begin{cases} 1 & u_0 \le u \le u_0 + w' - 1 \\ & v_0 \le v \le v_0 + h' - 1 \\ 1 & w - u_0 - w' + 1 \le u \le w - u_0 \\ & h - v_0 - h' + 1 \le v \le h - v_0 \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

and horizontally symmetric box function are defined in a similar manner. It is obvious, that dictionary $\mathbf{D}$ of the basis functions (atoms) is over-complete, redundant and non-orthogonal. Next, we need to approximate the Gabor wavelets ψ as a linear combination of atoms $\mathbf{b}$ from dictionary $\mathbf{D}$.

$$\psi \approx \widehat{\psi} = \sum_{i \in \Lambda} c_i \mathbf{b}_i \tag{7}$$

It is known, that the problem of finding a global optimum of the approximation is considered as NP-hard, since the dictionary consists of $H(H+1)W(W+1)/4$ single and $2H(H+1)W(\frac{W}{2}-1)/8$ symmetric atoms for a $W \times H$ large kernels.

The authors of [21], [22] proposed the use of a greedy algorithm called Optimized Orthogonal matching Pursuit

(OOMP)[2] [23], [24] that finds a sub-optimal solution and best approximates the original function. OOMP efficiently selects the most representative atoms from an arbitrary redundant nonorthogonal base vector dictionary in a Hilbert space. The OOMP iteratively selects the given number of base vectors $\mathbf{B}_\Lambda = \{\mathbf{b}_{l_1}, \mathbf{b}_{l_1}, \ldots, \mathbf{b}_{l_{|\Lambda|}}\}$ from a dictionary according to the following procedure: suppose that after iteration $\kappa-1$, the already selected $\kappa-1$ atoms are defined by the index set $\Lambda_{\kappa-1} = (l_i)_{i=1}^{\kappa-1}$. At iteration κ, the OOMP selects the index $l_\kappa = i$, that minimizes the new residual, which is equivalent to maximizing

$$\frac{|<\gamma_i, \varepsilon_{\kappa-1}>|}{\|\gamma_i\|}, \quad \|\gamma_i\| \neq 0, \quad i \in \overline{\Lambda}_\kappa, \tag{8}$$

where $\varepsilon_{\kappa-1} = \mathbf{x} - \xi_{\mathbf{B}_{\Lambda_{\kappa-1}}}(\mathbf{x})$ is the reconstruction residual using $\mathbf{B}_{\Lambda_{\kappa-1}}$, $\gamma_i = \mathbf{b}_i - \xi_{\mathbf{B}_{\Lambda_{\kappa-1}}}(\mathbf{b}_i)$ is the component of $\mathbf{b}_i$ that is orthogonal to the subspace spanned by $\mathbf{B}_\Lambda$. $\xi_{\mathbf{B}_{\Lambda_{\kappa-1}}}(\mathbf{x}) = \mathbf{B}_\Lambda(\mathbf{B}_\Lambda^{\mathrm{T}}\mathbf{B}_\Lambda)^{-1}\mathbf{B}_\Lambda^{\mathrm{T}}\mathbf{x}$ is the reconstruction of the signal $\mathbf{x}$ using the nonorthogonal bases indexed by $\Lambda_{\kappa-1}$. $\overline{\Lambda}_{\kappa-1}$ are the indices, that are not selected in the previous $\kappa-1$ iterations.

To satisfy one of the design constraints for filters measuring phase disparities and to ensure optimal phase behavior of all filters contained in the bank, it is necessary to remove the DC component of the filter. This is quite straightforward, if we can easily remove the mean from the filter. The problem is, that the decomposition of the filter into the linear combination consists of coefficients and associated basis functions (Haar-like box filters), that can not be changed (binary functions). Hence, the constraint is obvious: it is necessary to remove the DC component just by tuning the coefficients associated with selected atoms. Let χ_{E_i} be the characteristic function and $m(E_i)$ is the measure (e.g. square of L^2) of the box $\mathbf{b}_i$. Then, the approximated Gabor wavelets with the removed DC component are computed as

$$\widehat{\psi}_{DC} = \sum_{i \in \Lambda} \left(c_i - \frac{\delta}{m}\right)\mathbf{b}_i, \tag{9}$$

where $\delta = \sum_i c_i m(E_i)$ and $m = \sum_i m(E_i)$. The final kernels $\widehat{\psi}_{L^2}$ are obtained by normalization, so that $< \widehat{\psi}_{L^2}, \widehat{\psi}_{L^2} >= 1$, i.e. normalized by L^2 norm.

Consequently, the standard convolution can be approximated with N Haar-like box filters selected by OOMP as

$$\varsigma_{\theta,\omega} = \psi_{L^2} * \mathbf{\Omega} \approx \widehat{\psi}_{L^2} * \mathbf{\Omega} = \sum_{i=1}^{N} \alpha_i(\mathbf{b}_i\mathbf{\Omega}), \tag{10}$$

where $*$ denotes the convolution operator, $\mathbf{\Omega}$ is the image patch and α_i are the DC corrected and L^2 normalized coefficients.

The advantage of computing with an integral image trick is, that the filtering of Gabor wavelets approximated by Haar-like binary box functions with an image patch significantly reduces the number of floating point multiplications and additions. Computing of an integral image (only once per image) requires only $width \times height \times 2$ integer additions with recursive implementation. Each of the N selected atoms consist of one or two boxes. Let n_{sin} be the number of single and n_{sym} be the number of selected symmetric Haar-like box functions (naturally, $N = n_{sin} + n_{sym}$). Then, the approximated convolution needs only $3n_{sin}$ and $7n_{sym}$ integer additions, N floating point multiplications and additions[3]. It is obvious, that the most important parameter, that influence filtering speed is the number of selected atoms N.

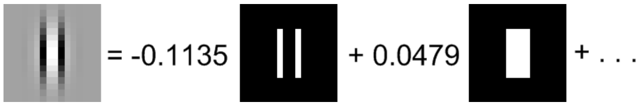


Fig. 1. Decomposition of a Gabor wavelet into a linear combination of Haar-like box functions.

Gabor wavelets have two parts: a real and an imaginary component. Thus, an average over the scales of a square norm of the so-called Gabor energy (complex response) is computed to get the best characteristics of a local texture jet

$$\mathbf{E}_\theta(x,y) = \mathrm{Avg}_\omega \left[\Re(\varsigma_{\theta,\omega}(x,y))^2 + \Im(\varsigma_{\theta,\omega}(x,y))^2\right] \tag{11}$$

Rasmussen [10] defines the dominant orientation of a texture flow at pixel $\mathbf{p}(x,y)$ as the filter orientation which elicits the maximum complex response at that location, however, it was observed [21], [22], that the estimated dominant orientation is not reliable at all pixels, especially at those, that are not related with road. Kong et al. [12], [13] propose to use a confidence score, which measures how peaky the function $\theta \mapsto \mathbf{E}_{\theta,\omega}(x,y)$ is near the optimum angle $\theta(x,y)$, however their confidence score does not take into account, how many maxima the function have. Thus, we slightly refined the confidence score evaluation, since the peaky function near the optimum angle is not enough - to get the reliable dominant orientation estimation, all point with multiple maxima must be rejected as well.

Instead of using of an ordered set of the complex responses, that do not take into account the angle θ for which the response is measured, we rather pick just the strongest response $e_{max}(x,y)$ from $\mathbf{E}_{\theta,\omega}(x,y) = \{e_1(x,y), e_2(x,y), \ldots, e_A(x,y)\}$ and measure the confidence score as

$$\mathrm{Conf} = 1 - \frac{\mathrm{Avg}\,\vartheta}{e_{max}(x,y)}, \tag{12}$$

$$\vartheta = \{e_{max-b}(x,y), \ldots, e_{max}(x,y), \ldots, e_{max+b}(x,y)\}, \tag{13}$$

where A is the number of orientations and b is the coefficient, that determines how much weaker the other responses are expected to be (we use $b = \frac{A}{4} - 1$). In addition to that, all other responses from $\overline{\vartheta}$ (complement of the set ϑ) are compared with Avg ϑ and if any of them is higher than Avg ϑ, the confidence score is set to zero to reject the pixel with multiple maxima. Next, the confidence score is normalized to the range $< 0, 1 >$ and threshold, so that

[2] Terminological remark: *Orthogonal* is related with the full backward orthogonality of the residual in each iteration and the fact, that the reconstruction using OOMP-selected base vectors is orthogonal to the residual.

[3] It is expected, that subtraction has the same computational complexity as addition, throughout this paper.

all pixels with a confidence level lower than T = 0.3 are discarded.

III. VOTING SCHEME

The second stage of the vanishing point estimation algorithm is voting. The idea behind the voting scheme assumes, that the set of parallel lines in the 3D space do not look like a parallel under the perspective projection caused by a pinhole camera, however these lines converge to some point on the image plane, the so-called vanishing point. The vanishing point is important in many computer vision applications, e.g. structure from motion [25], robotics, ... One can argue, that a single vanishing point have only straight roads and there might be multiple vanishing points in the case of a curved road. This is not a constraint, since we simply estimate the strongest response. The original voting scheme proposed by Rasmussen [10] is the tightest bottleneck of the whole algorithm. Kong et al. [12], [13] proposed a locally adapted soft-voting scheme, that slightly reduces the computational complexity of the original algorithm, however it is still far away from real-time.

Hence, the first step of our algorithm is a reduction of vanishing point candidates (the number of voting pixels were reduced by the confidence score). The idea is that the vanishing point should lie close to the points with significant dominant orientation, since the locally adapted voting strategy allows voting only to the points, that are in a close supporting subregion (discussed below). Thus, we take into account only those points, that are not rejected in the previous stage (confidence score based thresholding) and perform simple morphological dilation to include pixels, that are close to the huge support region, however they do not have significant dominant orientation themselves. The set of these points C are the vanishing point candidates. It is possible to use more sophisticated algorithms, however we use only such a simple preprocessing filter due to the very low computational complexity.

Next, we introduce superpixels into the voting scheme: a histogram of orientations is computed for each $f \times f$ subregion in the image and if the population of the histogram maxima is higher than some threshold $\tau = 0.5(f \times f)$, the angle which corresponds with the maxima is associated with the current superpixel and the next $l-1$ strongest orientations in histogram are compared with another threshold, equal to some fraction (0.5) of the histogram maxima frequency. If the frequencies of these additional $l-1$ orientations are higher than this threshold, they are associated with the current superpixel as well. To preserve the character of the superpixel, we not only store orientations, but also their histogram frequencies.

Once we have the set of possible vanishing point candidates C, one can make to vote the pixels with an estimated dominant orientation to obtain the vanishing point. Formally, 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, which has a finite value of $\frac{\pi}{n}$. This hard-voting strategy proposed by Rasmussen [10] has one drawback – voting pixels are all pixels below the vanishing point candidate and voting pixels are not weighted by the distance to the vanishing point candidate. This caused, that the vanishing point candidates higher in the image to have more potential voting pixels and it may lead to false detection. To overcome this limitation, Kong et al. [12], [13] introduced locally adaptive soft-voting, that reduce the region of voting pixels R to an intersection of the Gabor response image with a half-disk below the vanishing point candidate $\mathbf{v}$ centered at $\mathbf{v}$. A radius of the half-disk is $r = 0.35\Gamma$, where Γ is the length of the image diagonal. Each pixel with significant dominant orientation from R can vote for vanishing point candidate $\mathbf{v}$ if the following condition is satisfied

$$vote(\mathbf{p},\mathbf{v}) = \begin{cases} \frac{1}{1+[\gamma d(\mathbf{p},\mathbf{v})]^2} & \text{if} \quad \gamma \leq \frac{5}{1+2d(\mathbf{p},\mathbf{v})}, \\ 0 & \text{otherwise,} \end{cases} \qquad (14)$$

where $\gamma = |\alpha(\mathbf{p},\mathbf{v}) - \theta_{max}(\mathbf{p})|$.

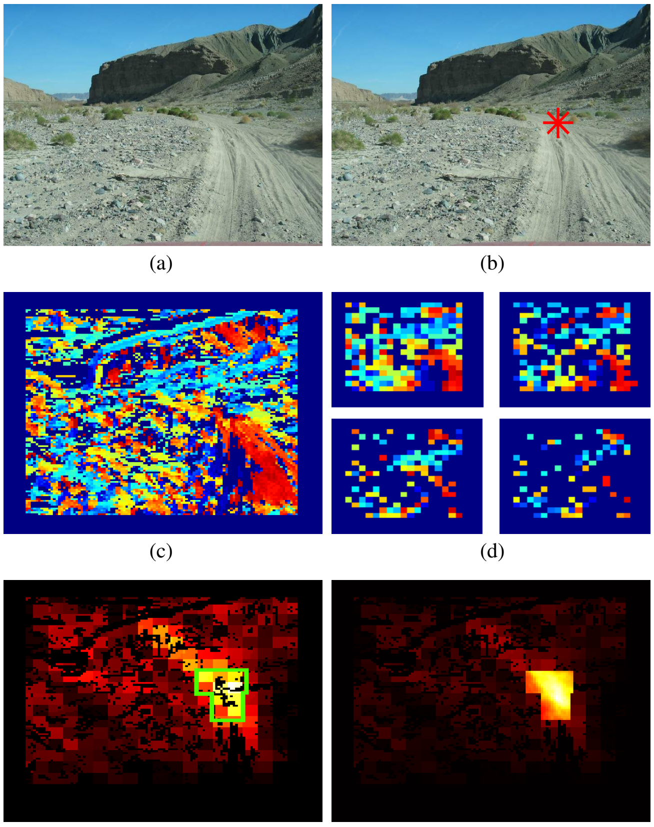

(a) (b) (c) (d) (e) (f)

Fig. 2. Vanishing point estimation: a) input image, b) output, c) dominant orientations, d) superpixels (4 levels), e) rough estimation and an area with highest score (green), f) refinement

Next, the definition of an objective function for each vanishing point candidate $\mathbf{v}$ is straightforward

$$votes(\mathbf{v}) = \sum_{\mathbf{p} \in R(\mathbf{v})} vote(\mathbf{p},\mathbf{v}). \qquad (15)$$

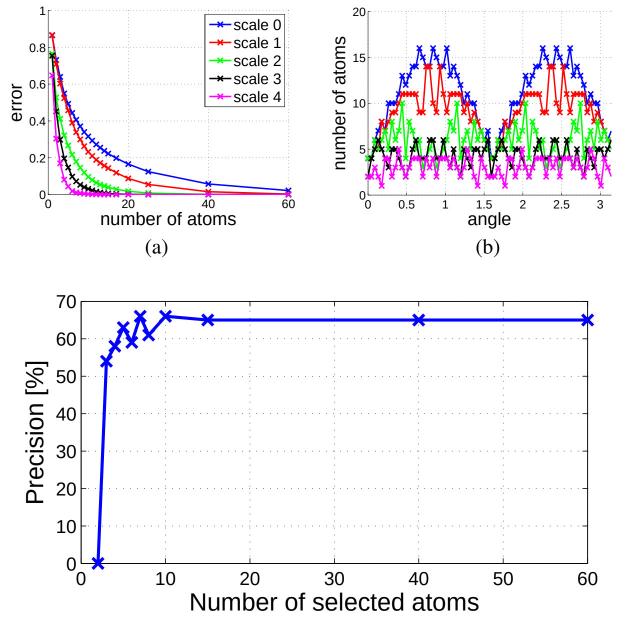
Here we discuss how we reduce the computational complexity of voting. Our voting consists of two stages. First, we vote for all possible vanishing point candidates from C, however only sparse superpixel representation is used for voting. Thus, we get a rough estimation where the vanishing point approximately lies. Next, we find the maxima of the objective function and establish a subregion (rectangle of a size j) around this point. Then, we check, if there are some points with a higher score than some fraction (0.8) of the global maxima. If such points exist, we establish another subregion around this point and iteratively re-scan all other pixels again, until there are no more pixels with a score higher than the fraction of the global maxima. Finally, the union of such regions (usually, there are only $1-3$) is used in the second stage of voting – the score of these pixels is cleared and voting is performed again, however, original dominant orientations are used instead of superpixels. Hence, we are able to estimate the vanishing point with the same precision and a huge reduction in computational complexity.

The differences in a computational complexity of various voting schemes are significant. Both, Rasmussen [10] and Kong et al. [12], [13] as well, consider all image points as vanishing point candidates. The latter approach is faster, since not all points below the vanishing point candidate can vote - the voting region is reduced to a half-disk with radius r, however, all pixels of $w \times h$ image are still considered as vanishing point candidates. In contrast to preceding voting schemes, our approach reduces the number of vanishing point candidates, since we consider only points with a high confidence score. Moreover, by utilization of integral image filtering, we do not need to estimate the dominant orientation at all pixels as in the case of FFT, however, we can easily discard all pixels with low variance. This variance filter can also be efficiently computed in an integral image domain. Based on our observations, usually approximately 50% of pixels are rejected. In the second step, the size of the voting region is reduced by factor f, since we use superpixels. Thus, we get a rough estimation of vanishing point coordinates in a very cheap way and only the union of subregions with score close to the maxima are re-voted. Usually, there are only $1-3$ subregions used for refinement, which have the same computational complexity as Kong et al. [12], [13], however only a few pixels are considered as candidates (consider $1-3$ subregions of size 8×8 against the full region with size of 128×128).

IV. PERFORMANCE EVALUATION

Our algorithm is tested on the same data set, used by Kong et al. [12], [13]. The data set consists of 1003 images, among them, about 430 images are from photographs taken on a scouting trip along a possible Grand Challenge route in the Southern California desert and the other part is downloaded from internet by Google Image. All images are normalized to the same height and width of 128, as is suggested in [14]. The ground truth data were obtained by manual labeling: 5 persons marked the vanishing point location, a median of these results is used as the initial ground-truth position. The two farthest manually marked locations to the initial ground-truth position are removed as outliers. Finally, the ground-truth location is computed as the mean of the other three locations. It should be noted, that we compare our results against Kong et al. [12], [13] method, without road segmentation, since several methods [14], [15] exist that employs vanishing point in a different way.

First of all, we evaluate, how many atoms are needed to sufficiently approximate Gabor wavelets. Clearly, the lower the number of approximating atoms is, the higher the speed of filtering, however the worse approximation. We run the OOMP and set desired number of basis (varied from 1 to 60) and measure the sum of absolute differences between Gabor wavelet and the reconstructed approximation. Fig 3 shows this error in each scale separately. It is obvious, that the precision varies over the scales. Hence, we do not choose the number of atoms, however set the precision and OOMP simply stops, if the precision of approximation is below this threshold ($t = 0.35^{1+0.1s}$). This is useful, since the number of selected atoms is not varied just over the scales, however even over the angles – it is clear, that filters with some orientations need more basis than others.



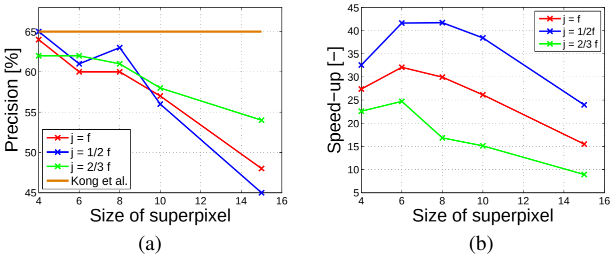
(a)

(b)

Fig. 3. The dependency of a number of selected basis and precision.

Figure 3 shows how the approximation error $\frac{\|\psi-\widehat{\psi}\|}{\|\psi\|}$ is dependent on a number of selected atoms. It is clear, that each scale is approximated with different error (c.f. Fig. 3 a) and each orientation of a filter require a different number of selected basis to ensure the required precision (c.f. Fig. 3 b). Fig. 3 c) compares precision (percentage of images with an estimated vanishing point within a region of 15 pixels around the ground-truth) of a various number of selected atoms. We use Kong et al. [12], [13] voting scheme for this evaluation to be independent on the quality of our voting method. It is obvious, that only 8 atoms are enough for reliable estimation of a vanishing point.

Next, we evaluate the precision and a speed of our voting scheme against Kong et al. [12], [13]. The speed of our voting scheme is dependent on the size of superpixels and the subregions used for refinement. Figure 4 shows, that the best trade-off between precision and speed-up is obtained for $f = 8$ and the size of subregions $j = 1/2f$. Important is, that our voting scheme is 41.7 times faster than Kong et al. [12], [13] while we loose only 3% in precision.



(a) (b)

Fig. 4. The dependency of a precision, size of superpixels and refinement regions (a) and a speed-up of a voting scheme against Kong et al. [12], [13] (b).

V. CONCLUSION

We have optimized computational complexity of a vanishing point estimation algorithm, which is widely used in the domain of a visual navigation. In the first part of the paper, we have employed binary approximation of Gabor wavelets, which enables faster computation via integral image trick. The tightest bottleneck of a vanishing point estimation was voting. We have proposed a cascaded voting scheme, which consists of two steps: 1) rough estimation of a vanishing point and 2) refinement. Such optimization makes the voting scheme more than 40× faster than Kong et al. [12], [13] and 50× faster than Rasmussen [10], while the precision is only $3-5\%$ worse than Kong et al. approach. Important is, that the proposed method is not hardware dependent and might be significantly faster with parallel processing.

VI. ACKNOWLEDGMENTS

We would like to thank Hui Kong for providing test images with ground truth and Ang Li and Jyrki Lahtonen for fruitful discussions.

REFERENCES

[1] H. Dahlkamp, A. Kaehler, D. Stavens, S. Thrun, and G. R. Bradski, "Self-supervised monocular road detection in desert terrain," in *Robotics: Science and Systems*, G. S. Sukhatme, S. Schaal, W. Burgard, and D. Fox, Eds. The MIT Press, 2006.

[2] T.-C. Dong-Si, D. Guo, C. H. Yan, and S. H. Ong, "Robust extraction of shady roads for vision-based ugv navigation," in *IROS*. IEEE, 2008, pp. 3140–3145.

[3] B. Ma, S. Lakshmanan, and A. O. Hero, "Simultaneous detection of lane and pavement boundaries using model-based multisensor fusion," *IEEE Transactions on Intelligent Transportation Systems*, 2000.

[4] J. Crisman and C. Thorpe, "Unscarf, a color vision system for the detection of unstructured roads," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 3, April 1991, pp. 2496 – 2501.

[5] ——, "Scarf: A color vision system that tracks roads and intersections," *IEEE Trans. on Robotics and Automation*, vol. 9, no. 1, pp. 49 – 58, February 1993.

[6] A. Broggi, M. Bertozzi, and A. Fascioli, "Argo and the millemiglia in automatico tour," *IEEE Intelligent Systems*, vol. 14, no. 1, pp. 55–64, 1999.

[7] M. Bertozzi, A. Broggi, A. Fascioli, and S. Nichele, "Stereo vision-based vehicle detection," in *in IEEE Intelligent Vehicles Symposium*, 2000, pp. 39–44.

[8] D. Lieb, A. Lookingbill, and S. Thrun, "Adaptive road following using self-supervised learning and reverse optical flow," in *Robotics: Science and Systems*, S. Thrun, G. S. Sukhatme, and S. Schaal, Eds. The MIT Press, 2005, pp. 273–280.

[9] A. Wedel, T. Pock, C. Zach, H. Bischof, and D. Cremers, "An improved algorithm for tv-l1 optical flow," *Statistical and Geometrical Approaches to Visual Motion Analysis: International Dagstuhl Seminar*, pp. 23–45, 2009.

[10] C. Rasmussen, "Grouping dominant orientations for ill-structured road following," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2004.

[11] ——, "A hybrid vision + ladar rural road follower," in *IEEE International Conference on Robotics and Automation*, 2006.

[12] H. Kong, J.-Y. Audibert, and J. Ponce, "Vanishing point detection for road detection," *IEEE Conference on Computer Vision and Pattern Recognition (2008)*, no. 3, pp. 96–103, 2009.

[13] ——, "General road detection from a single image," *IEEE Transactions on Image Processing*, vol. 19, no. 8, pp. 2211–2220, 2009.

[14] O. Miksik, P. Petyovsky, L. Zalud, and P. Jura, "Robust detection of shady and highlighted roads for monocular camera based navigation of ugv," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[15] Q. Wu, W. Zhang, and B. V. K. V. Kumar, "Example-based clear path detection assisted by vanishing point estimation," in *ICRA*. IEEE, 2011, pp. 1615–1620.

[16] C. Rasmussen and T. Korah, "On-vehicle and aerial texture analysis for vision-based desert road following," in *IEEE International Workshop on Machine Vision for Intelligent Vehicles*, 2005.

[17] T. S. Lee, "Image representation using 2D gabor wavelets," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, pp. 959–971, 1996.

[18] J. P. Jones and L. A. Palmer, "An evaluation of the two-dimensional gabor filter model of simple receptive fields in cat striate cortex," *J Neurophysiol*, vol. 58, no. 6, pp. 1233–1258, December 1987.

[19] B. R. Payne, Belkasim, S. G. Owen, Weeks, and Y. Zhu, *Accelerated 2D Image Processing on GPUs*, Jan. 2005, vol. 3515.

[20] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR 2001*, vol. 1, no. C, pp. I–511–I–518, 2001.

[21] F. Tang and H. Tao, "Non-orthogonal binary expansion of gabor filters with applications in object tracking," *2007 IEEE Workshop on Motion and Video Computing WMVC07*, pp. 24–24, 2007.

[22] F. Tang, R. Crabb, and H. Tao, "Representing images using nonorthogonal haar-like bases." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 12, pp. 2120–2134, 2007.

[23] L. Rebollo-Neira and D. Lowe, "Optimized orthogonal matching pursuit approach," *IEEE Signal Processing Letters*, vol. 9,4, pp. 137–140, 2002.

[24] M. Andrle and L. Rebollo-Neira, "A swapping-based refinement of orthogonal matching pursuit strategies," *Signal Process.*, vol. 86, pp. 480–495, March 2006.

[25] O. Barinova, V. Konushin, A. Yakubenko, K. Lee, H. Lim, and A. Konushin, "Fast automatic single-view 3-d reconstruction of urban scenes," in *Proceedings of the 10th European Conference on Computer Vision: Part II*, ser. ECCV '08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 100–113.