Vanishing point detection in corridor for autonomous mobile robots using monocular low-resolution fisheye vision

Dejing Ni¹,², Peng Ji¹ and Aiguo Song³

Abstract
It is crucial for mobile robots to implement vanishing point detection during navigation in corridors. For the fisheye vision, the conventional methods of vanishing point detection usually obtain poor detection results. This is mainly attributed to serious barrel distortion in images acquired from fisheye cameras that are widely used in mobile robot systems. In the proposed system, a novel vanishing point detection algorithm based on the Gabor filter bank and the convolutional neural network is put forward to realize more accurate detection. The Gabor filter bank is used to extract image texture information in the preprocessing step, thereby enhancing the generalization. The convolutional neural network is used to predict the position of the vanishing point in the fisheye images. To improve the real-time performance and guarantee the accuracy, the low-resolution image should be selected as the input image as far as possible. For this purpose, a multi-resolution experiment was carried out. With the appropriate resolution, the proposed vanishing point detector was found still effective even if 60% of the original information was discarded. In addition, an experiment was conducted to verify the generalization on the condition of illumination changing, pedestrians passing, and different corridor appearance. The experiments displayed good effect and generalization on fisheye images captured in the corridor.

Keywords
Vanishing point, convolutional neural network, mobile robots, fisheye vision, Gabor filters

Date received: 24 June 2019; accepted: 23 September 2019

Handling Editor: Eric Fujiwara

Introduction
Autonomous navigation in the corridor using monocular vision has aroused a wide range of attention in the field of mobile robots. There are many parallel lines in the structured indoor environment. In a projected image plane, those parallel lines may no longer parallel but converge to one point that is defined as the vanishing point (VP). Providing important cues for inferring the three-dimensional (3D) structure of a real scene, VP is widely applied in mobile robot navigation,¹ road detection⁵ and camera calibration⁴ and so on. As a result, it becomes common to take VP as the direction cues for mobile robots wandering in the corridor.¹

In VP detection, the mainstream methods are generally classified into two categories, edge-based methods and texture-based methods. The edge-based methods⁴,⁵ usually adopt the histogram technique on the basis of statistics. Specifically, Hough transform or Random Sample Consensus (RANSAC)⁶ is used to detect lines
in the projected image plane; then, the potential intersection of these lines is found mainly by voting methods and considered as VP. By contrast, texture-based methods\textsuperscript{7–9} have no need to detect lines but vote for the most possible VP by the convolution operation of texture filter and each pixel of image. For instance, a gLoG-based VP detection method was proposed by Kong et al.,\textsuperscript{9} in which a generalized laplacian of gaussian (gLoG)-based filter was used to estimate the texture orientation at each pixel, and then, the potential VP was voted by the dense texture orientation map. The texture-based methods are able to realize accurate detection of the VPs on both well-paved urban roads and rural unstructured off-roads. Nevertheless, high computational complexity and poor real-time performance lead to their failure to meet the needs of the autonomous mobile robots that require both vision navigation and scene recognition.

In addition to the above two categories, there is another type of VP detection methods, which are based on optimization algorithms. Typical examples include Dynamic Clustering Artificial Bee Colony (DCABC)-based algorithm\textsuperscript{10} and Multiple Population Genetic Algorithm (MPGA)-based algorithm.\textsuperscript{11} They first consider VP detection as a dynamic line clustering problem and then search for VP using some optimization algorithms such as Artificial Bee Colony algorithm and Genetic algorithm.

The wide view angle of fisheye cameras allows mobile robots to acquire more visual information during navigation. However, those fisheye images are usually accompanied by serious barrel distortion, as shown in Figure 1. In traditional methods, VP detection is considered a dynamic line clustering problem, which greatly reduces the detection effect for those images with serious barrel distortion. Taking the monocular camera as the navigation camera for a mobile robot, this article proposes a novel VP detection algorithm based on the supervised method, which requires only a sample set marked with the location of VP. A VP detector is trained off-line by the supervised method. The Gabor filter bank is used to extract the texture information of corridor in the off-line training phase. This helps the detector used in this study to gain good generalization ability in corridors with illumination changing, pedestrian passing, or appearance changing.

In addition, psychological studies proved the unnecessary of high-resolution images for human intelligence in basic visual navigation.\textsuperscript{12} The Andre et al.\textsuperscript{13} also pointed out that it was still relatively easy for human beings to complete some visual navigation, despite loss in visual acuity and color vision. In other words, for some relatively simple navigation tasks, such as moving along a straight line, only little visual information is really useful. Therefore, to select the appropriate low resolution (to meet the real-time requirement while not reducing the accuracy), a multi-resolution experiment was performed. The experiment results demonstrated that the proposed VP detection algorithm was still effective even after 60% of the visual

Figure 1. The mobile robot for experiments and distortion images captured by the fisheye camera.
information was removed. This proved that the proposed algorithm was able to achieve a good accuracy and real time without using high-resolution image, freeing up much of the CPU for other computation-intensive tasks, such as landmark recognition.

The remainder of this article is organized as follows. The overall block diagram of the innovative VP detector is introduced in section “Overview of the system,” followed by the fundamental of the detector elaborated in section “The proposed method.” Experiments in section “Experiments” verify the performance of our proposed detector. The conclusion of this study is given in section “Conclusion.”

Overview of the system
The experimental images used in this article were captured by the front-facing camera of a self-designed mobile robot, as shown in Figure 1. A tracked mobile robot is used for reconnaissance detection. Different from the traditional edge-based methods, texture-based methods, and optimization-based methods, our VP detection method takes the VP detection as a regression problem. For this purpose, a novel VP detector based on a convolutional neural network is proposed, which consists of two parts, the image texture information extraction and the merging of the predictions. To eliminate the influence of vertical lines on VP prediction, a multi-scale Gabor filter bank (five scales at seven directions, not including 90 degree) is used to filter the fish-eye image. Based on the mean value of predictions at seven directions of each scale, the final VP is calculated by the mean value of prediction at five scales. Figure 2 presents the overall block diagram of the proposed VP detector in this article.

The proposed method
Multi-scale texture information extraction
The proposed VP detector is based on the supervised method, which makes it necessary to use the training sample set for off-line training. In this article, to eliminate the interference of vertical segments in the image and enhance generalization ability of the VP detector, a Gabor filter bank (five scales at seven directions, not including 90 degree) is employed to extract the texture feature information first. The impulse response of the Gabor filter is defined as a sine wave (two-dimensional (2D) Gabor filter is a sinusoidal plane wave) multiplied by the Gauss function. The 2D Gabor filter is expressed as below

\[
G(x, y) = \frac{1}{2\pi \cdot \sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \exp\left(i2\pi \frac{x}{\lambda} + \psi\right)
\]

where \(\lambda\) is the wavelength of sine wave, \(\theta\) represents the direction of parallel wavelet of Gabor wavelet, \(\psi\) is the phase offset; \(\gamma\) is the aspect ratio of Gabor wavelet.
determining the shape of the Gabor wavelet, and $\sigma$ is standard deviation of the Gaussian function.

**VP detector**

Thanks to the advantage of the deep neural network (especially the convolutional neural network) in the image, the convolutional neural network is used as the VP detector, which traditionally consists of a convolution layer and pool layer. As pointed out by the paper, the pool layer is capable of weakening the position of the image feature. That is, the pool layer is beneficial to the position-related classification problem, instead of the position-independent regression problem (such as the VP detection problem). Therefore, a simple convolutional neural network different from the traditional one (where the convolution layer and the pool layer appear alternately) is proposed. Its structure is shown in Figure 3.

As shown in Figure 3, the first hidden layer, as an input layer, convolves three filters of $5 \times 5$ with stride 2 with the input image and applies a rectifier nonlinearity. The second hidden layer convolves six filters of $5 \times 5$ with stride 2, again followed by a rectifier nonlinearity. The third hidden layer is a full connection layer, with 200 neurons and applying ReLU function as the activation function. The output layer is a fully connected linear layer, with the activation function of sigmoid function. The number of neurons in the output layer is two, which is obtained through the ratio of the horizontal and vertical coordinates of the VP in the input image.

**Merging of the multi-predictions**

After well trained, the VP detector can be used to predict VP’s location in the fisheye image. First, the input image is converted to grayscale image and then filtered by the Gabor filter bank. At each scale, a frame of texture image is obtained by all directions of the texture image, which are summed and normalized. The formula is expressed as

$$F_i = \frac{\sum_{m=1}^{M} f_i^m}{M}$$

where $M$ is set to 7, representing the number of filter directions, and $f_i^m$ refers to the $m$th texture feature image at $i$th scale; $F_i$ is the texture feature image at the $i$th scale.

Through the five-scale Gabor filter bank, five-frame texture feature image is acquired after input image is filtered and five prediction results are obtained after the five-frame texture feature image is sent to the VP detector. The final VP location is represented by the mean value of the five prediction results, as shown in the following

$$\left(p_x, p_y\right) = \left(\frac{\sum_{i=1}^{N} p_i}{N} \ast I_{W}, \frac{\sum_{i=1}^{N} p_i}{N} \ast I_{H}\right)$$
where $N = 5$ means five scales, $I_W$ and $I_H$ are the width and height of the input image, respectively, and $(p_x, p_y)$ is the final VP location.

**Experiments**

**Sample set and Gabor filter results**

In this study, to get the training sample set, the mobile robot was controlled to do S-shaped movement in the corridor, and a video (with serious barrel distortion) captured by the fisheye camera was recorded. A total of 1786 frames ($100 \times 100$) were extracted from the video, and then, the location of VP was manually marked in each frame.

Overall 1/5 frames from 1786 frames ($100 \times 100$) were randomly selected as the test sample set, and the remaining frames (in different parts of the same corridor, fluorescent lighting, no pedestrians passing, as shown in the first row of Figure 4(a)) were as the training sample set. Also, corresponding coordinates of the VP were considered the label of the training sample set.

In addition, to test the generalization ability of the proposed VP detector, 130 frames with illumination changing, 130 frames with pedestrians passing by, and 130 frames captured in other corridor with pedestrians passing by were collected as well, as shown in Figure 4(b).

Figure 4 clearly indicates that straight lines of corridor are regular when VP is located at the center of image but distorted when VP's location deviates slightly from the center of image. This also explains why the conventional VP detection algorithms make poor performance when processing the images with barrel distortion.

The Gabor filter bank used in this article involves five scales and seven directions. The phase offset is not taken into account, that is, $\psi = 0$ and $\gamma = 0.7$. The relationship between the standard deviation of the Gauss function and the wavelength of the sine wave was expressed as

$$\sigma = \frac{1}{\pi} \sqrt{\frac{\ln 2}{2 \cdot 2^b + 1}} \cdot \lambda$$  \hspace{1cm} (4)

$$\lambda = L_{\text{max}} \cdot \sqrt{2^{i-1}}, i = 1, \cdots, 5$$  \hspace{1cm} (5)

where $b = 1.5$, $L_{\text{max}} = 2$, $\theta = (22.5, 45, 67.5, 0, 112.5, 135, 157.5)$.

After filtered by the Gabor filter bank, texture feature frames of five scales and seven directions were obtained as shown in Figure 5. Then, the texture feature frames of all directions at each scale were fed into the neural network for training and testing after normalization. In this article, the learning rate of 1 and the number of iterations of 10 were used.

**Performance of our detector and contrast algorithm on the test set**

Cross-validation is a resampling procedure used to evaluate machine learning models on a limited dataset. In K-fold cross-validation, the original dataset is randomly partitioned into $K$ subsamples. Of the $K$ subsamples, the $K-1$ subsamples are used as training data, for testing the model, the remaining part of the subsample is used as the validation data to compute a performance measure. The cross-validation process is then repeated $K$ times (the folds), and then, the $K$ results can be averaged as the final estimation. In this study, the $K$-fold cross-validation method was used to verify the proposed VP detector, where the $K$ was 5. In other words, 1/5 frames were randomly selected from 1786 frames each time as a validation set, and the remaining samples were as training samples. The final result was represented by the mean value of the five prediction results. The selected test samples covered the starting, middle, and end position of the corridor.

In terms of a test image, the distance between the predicted VP location and ground-truth location was computed. The detected VP was defined as correct.
when the distance was smaller than a threshold. Thus, an accuracy rate on the test samples was obtained. By changing the threshold from 0% to 30% of the image width, an accuracy rate was obtained as shown in Figure 6. X axis represents the threshold, and Y axis refers to the accurate rate of the prediction. The Euclidean distance was used to measure the distance.

A test was carried out to verify the performance difference of the proposed VP detector on different resolutions. Figure 6(a)–(c) shows the performance curves on the resolution of 10 × 10, 20 × 20, 30 × 30, 40 × 40, 50 × 50, 60 × 60, 70 × 70, and 100 × 100, respectively. Considering the impact from the kernel size of the Gabor filter on the final results, three kernel sizes (5, 8, and 10) were used, and the performance curve of which is shown in Figure 6(a)–(c), respectively. It was observed that the VP detector exhibited similar performance when the resolution was reduced from 100 × 100 to 20 × 20 but poor performance when the resolution was 10 × 10.

To intuitively compare the impacts of different kernel sizes and resolutions on the VP detector, a metric, degree of error (DE), was introduced and expressed as

\[ DE = \frac{\text{min}_{th}\{100\%\}}{\sqrt{n_w + n_h}} \]

where \( \text{min}_{th}\{100\%\} \) is the minimum distance threshold when the detector achieved the 100% accuracy rate on the test sample set and \( n_w \) and \( n_h \) stand for the width and height of image, respectively. With the decrease in the \( DE \), the VP detector achieved 100% accuracy on the test set more quickly when the threshold condition was relaxed. In other words, the smaller the \( DE \), the better the performance of the VP detector.

The \( DE \) of the detector used in this article with different kernel sizes and resolutions was shown in Figure 7. For human beings, the larger the resolution is, the more information the image contains, so the VP is more easy to be detected. As shown in Figure 7, the performance of the proposed VP detector was basically consistent with the human’s cognition. In other words, the larger the resolution is, the better the detector performs. In addition, the performance difference of the detector was not large in the case of three kernel sizes; comparatively speaking, it was a little better under the kernel size of 5. This is probably because a big kernel tended to blur the texture of the image, which was not conducive to the generalization ability of the detector.

As shown in Figures 6 and 7, when the resolution was reduced from 100 × 100 to 40 × 40, the VP detector performed similarly regarding the accuracy, and the \( DE \) was only up to about 10% (for 40 × 40, the maximum error was only 10% × 40 = 4 pixels). However, when the resolution was reduced from 30 × 30 to 10 × 10, the \( DE \) experienced a significant increase. This confirmed that the proposed detector for low resolution (40 × 40) of fisheye images remained effective even if 60% information of the original image (100 × 100) was lost.

To compare the performance difference of our detector and other algorithms, a contrast experiment was carried out on the fisheye test sample set. GS,17 Expectation Maximization (EM),18 interior and exterior region (IER),5 and gLoG9 were selected for comparison. The GS algorithm by Barnard17 takes the Gaussian sphere as the accumulated space. Using the
EM algorithm, Kosecka and Zhang\textsuperscript{18} presents an efficient approach for detection of VPs from a single view assuming an uncalibrated camera in man-made environments. The IER algorithm by Li\textsuperscript{5} realizes the VP detector using one-dimensional (1D) histogram to detect the three orthogonal vanishing point from the both interior region (IR) and exterior region (ER). The gLoG algorithm applies a new gLoG filter to estimate the texture orientation at each pixel of an image and the vanishing point can be detected based on the estimated texture. All the algorithms were based on the original resolution ($100 \times 100$), and slight modification was made to the source code of EM and gLoG to adapt to this resolution.

Figure 8(a) provides the comparison results between the proposed VP detector ($100 \times 100$, kernel size: 5) and other algorithms. GS outperformed EM, IER, and gLoG, but its accuracy was still less than 100% even if the threshold was relaxed to 30%. The EM, IER, and gLoG made relatively poor performance in the fisheye image test set with serious barrel distortion, and gLoG and EM achieved almost the same accuracy rate.

In order to verify the detection effect of using low-resolution fisheye images, another contrast experiment
was carried out using the low resolution (60 × 60, kernel size: 5), with the results shown in Figure 8(b). The accuracy of the other algorithms was found significantly decreasing except the proposed detector. The IER algorithm seemed powerless to the low-resolution fisheye images, since its accuracy gained no improvement with the relaxing of threshold. As shown in Figure 8(a) and (b), it was concluded that GS, EM, IER, and gLoG indeed were not qualified to detecting the VP of low-resolution fisheye image.

The proposed VP detector was designed for the autonomous navigation of mobile robots in the corridor, making the real-time feature greatly significant. Therefore, a real-time contrast experiment was carried out involving the above-mentioned four algorithms. To reduce the test difference as far as possible, all the five algorithms were still based on the resolution of 60 × 60 and 100 × 100, respectively, and tested on the same Windows PC with an Intel Core i5 CPU (3.20 GHz) and 4-GB memory space. The codes of all algorithms were implemented in MATLAB. Figure 9(a) and (b) present the performance of these five algorithms under the small resolution (60 × 60) and the original resolution (100 × 100), respectively. For intuitive demonstration, the error bar graph of the contrast algorithms in these two cases was obtained, as shown in Figure 9(c).

The processing time of all algorithms increased with the growing of resolution. In addition, under the original resolution (100 × 100), gLoG performed worst with an average frame rate of only 0.37 frames per second; the IER algorithm performed best with an average frame rate of 43.38 frames per second; the proposed algorithm achieved an average frame rate of 16.47 frames per second, basically meeting the real-time requirement.

**Generalization ability of the proposed VP detector**

The ultimate goal of the proposed VP detector is to enable the mobile robot to autonomously navigate in the corridor with the fisheye camera. The generalization ability of the detector plays an important role. To verify the generalization ability of the VP detector, three test sample sets were collected, each of which was 130 frames (60 × 60). The first set was collected under dim lighting conditions (with a sudden illumination change caused by turning the light on); the second set was accompanied by pedestrian passing noise and the third set was collected in other corridors and accompanied by pedestrian passing noise.

Figure 10 shows the performance of the proposed VP detector on the three test sets. The red curve presents the prediction error of the VP detector on the first test set, which, as shown in segment A, suddenly increased to 10 pixels when the light was suddenly turned on. This may be because the camera in the light suddenly experienced a switch process from excessive exposure to normal. The green curve is the prediction error on the second test set, which, as shown in segment B, increased first and then decreased. This could be explained by a pedestrian approaching the camera and then leaving the camera (from near to far away). The blue curve provides the prediction error on the third test set, which, as presented in segment C, experienced an increase first and went down. Different from segment B, the segment C was caused by a pedestrian passing in front of the camera (from left to right, instead of from near to far away). The segment C obtained higher error than the segment B, which may be attributed to an occlusion process when the pedestrian crossed through the camera. The blue curve shows...
that the error was basically maintained at about 5 pixels when no pedestrian was passing. This was a little different from the other two test sets (illumination changing and pedestrian passing). This proved that the proposed VP detector remained effective under the conditions of illumination changing, pedestrian passing, and different corridor appearance. Thus, the VP detector exhibited good generalization ability.

**Conclusion**

For monocular fisheye images, VP detection is regarded as a regression problem. A novel VP detection algorithm based on the convolutional neural network is proposed in this article. To select the appropriate low resolution (to meet the real-time requirement while not reducing the accuracy), a multi-resolution experiment was conducted. With the appropriate resolution, the proposed VP detector was found to remain effective even if 60% of the original information was discarded. To verify the generalization of the detector, another experiment was performed under the condition of illumination changing, pedestrians passing, and different corridor appearance. The proposed VP detector exhibited good effect and generalization ability on fisheye images captured in the corridor. In addition, a comparison real-time experiment was carried out, in which the average frame rate of the VP detector reached 16.47 frames per second. This basically met the real-time requirement.

**Figure 9.** Computational time comparison between the proposed method and the state-of-the-art methods. (a) Results on the low resolution (60 × 60). (b) Results on the original resolution (100 × 100). (c) The error bar graph of the state-of-the-art methods on the low resolution and original resolution.

**Figure 10.** Test results of generalization ability.
Declaration of conflicting interests
The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the National Natural Science Foundation of China (No.61325018, No.61272379, No. 61903207).

References
1. Ebrahimpour R, Rasoolinezhad R, Hajiabolhasani Z, et al. Vanishing point detection in corridors: using Hough transform and K-means clustering. IET Comput Vis 2012; 6: 40–51.
2. Bui TH, Saitoh T and Nobuyama E. Vanishing point-based road detection for general road images. IEICE T Inf Syst 2014; E97.D: 618–621.
3. Moghadam P, Starzyk JA and Wijesoma WS. Fast vanishing-point detection in unstructured environments. IEEE T Image Process 2012; 21: 425–430.
4. Li B, Peng K, Ying XH, et al. Simultaneous vanishing-point detection in unstructured environments. Lecture notes in computer science. Berlin: Springer-Verlag, 2010, pp.151–160.
5. Li B, Peng K, Ying X, et al. Vanishing point detection using cascaded 1D Hough Transform from single images. Pattern Recogn Lett 2012; 33: 1–8.
6. Bazin JC and Pollefeys M. 3-line RANSAC for orthogonal vanishing point detection. In: IEEE international conference on intelligent robots and systems, Vilamoura, 7–12 October 2012, pp.4282–4287. New York: IEEE, 2012.
7. Kong H, Audibert J and Ponce J. General road detection from a single image. IEEE T Image Process 2010; 19: 2211–2220.
8. Kong H, Audibert JY and Ponce J. Vanishing point detection for road detection. In: IEEE conference on computer vision and pattern recognition, Miami, FL, 20–25 June 2009, pp.96–103. New York: IEEE, 2009.
9. Kong H, Sarma SE and Tang F. Generalizing Laplacian of Gaussian filters for vanishing-point detection. IEEE T Intell Transp 2013; 14: 408–418.
10. Han L, Fan T, Zheng S, et al. Vanishing point detection based on an artificial bee colony algorithm. J Electron Imaging 2015; 24, 033024.
11. Lu K, Li J, An X, et al. Road vanishing-point detection: a multi-population genetic algorithm based approach. In: 2013 Chinese automation congress (CAC), Changsha, China, 7–8 November 2013, pp.415–419. New York: IEEE.
12. Murali VN and Birchfield ST. Autonomous exploration using rapid perception of low-resolution image information. Auton Robot 2012; 32: 115–128.
13. Andre J, Owens DA and Harvey LO. Visual perception: the influence of H.W. Leibowitz. Washington, DC: American Psychological Association, 2003.
14. Qian K, Song A, Bao J, et al. Small teleoperated robot for nuclear radiation and chemical leak detection. Int J Adv Robot Syst 2012; 9: 70.
15. Haghighb M, Zonouz S and Abdel-Mottaleb M. CloudID: trustworthy cloud-based and cross-enterprise biometric identification. Expert Syst Appl 2015; 42: 7905–7916.
16. Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning. Nature 2015; 518: 529–533.
17. Barnard ST. Interpreting perspective images. Artif Intell 1983; 4: 435–462.
18. Kosecka J and Zhang W. Video compass. In: Proceedings of the 7th European conference on computer vision, vo. 4, Copenhagen, Denmark, 28–31 May 2002, pp.476–490. Berlin, Heidelberg: Springer-Verlag.