Fast Vanishing Point Estimation Based on Particle Swarm Optimization

Xun PAN†[a], Student Member, Wasi[b], and Harutoshi OGAI[c], Nonmembers

SUMMARY Vanishing point estimation is an important issue for vision based road detection, especially in unstructured roads. However, most of the existing methods suffer from the long calculating time. This paper focuses on improving the efficiency of vanishing point estimation by using a heuristic voting method based on particle swarm optimization (PSO). Experiments prove that with our proposed method, the efficiency of vanishing point estimation is significantly improved with almost no loss in accuracy. Moreover, for sequenced images, this method is further improved and can get even better performance, by making full use of inter-frame information to optimize the performance of PSO.

key words: vanishing point estimation, particle swarm optimization, inter-frame information

1. Introduction

Autonomous driving has been a hot research area over the past few decades. Among various parts of autonomous driving, road detection is a crucial one. Different sensors are used in different road detection methods, such as radar sensor [1], ladar sensor [2] and vision sensor. Among these sensors, vision sensors are mostly used for road detection [3].

Vanishing point estimation plays an important role in vision based road detection. Studies on vanishing point estimation of the road have lasted for more than 10 years. Some methods are based on edges in the road scene [6]. In this way, edges are extracted by edge detector, then road boundaries or lane markings are detected as straight lines by using Hough transform [7], and finally the vanishing point is estimated as the intersection of the line pairs. This group of methods may perform well in structured road, however in unstructured roads without lane markings, they won’t behave appropriately.

In order to overcome the shortcomings of edge-based method, texture-based methods have been proposed. Texture is a more powerful cue than a straight edge. It doesn’t rely on land markings or well paved road surface. And it is more robust to illumination changes. So texture features can work well in different road and weather conditions. C. Russmusen proposed a texture based vanishing point estimation method [5] in 2004. It can estimate the vanishing point in both structured and unstructured roads. However, due to some disadvantages in the voting strategy in his method, the pixel in the upper part of the image sometimes will be estimated as vanishing point inappropriately. Kong proposed a local adaptive soft voting method [17] to solve this problem in 2009. However, just as C. Russmusen, Kong also consider almost all pixels in the image as the candidates in the voting, which leads to a rather long time to complete the voting. Consequently, methods on improving the efficiency of the vanishing point estimation have been proposed. Miksik reduced the vanishing point candidates in voting by using super pixels [8]. This method can improve the efficiency of Kong’s method by more than 40 times. Moghadam et al. used skyline to reduce the candidates of vanishing point [10]. Furthermore, Haar functions, integral image [9] and orthogonal Gabor [4] filters are also used to improve the efficiency of the vanishing point estimation [9].

This paper also focuses on improving the efficiency of the conventional vanishing point estimation based on C. Russmusen and Kong’s method. By using a heuristic voting method based on PSO, we significantly improve the efficiency, and lose almost nothing in accuracy. Moreover, performance on sequenced images are improved further by making use of inter-frame information. The rest of this paper is organized as follows: In Sect. 2, we introduce the texture orientation estimation by using Gabor filters. In Sect. 3, a particle swarm optimization (PSO) based voting strategy is proposed. In Sect. 4, performance optimization of sequenced images is discussed. In Sect. 5, we analyse the experiment data and evaluate the performance of our proposed algorithm. Finally, conclusions are drawn in Sect. 6.

2. Texture Orientation Estimation

The basic principle of vanishing point estimation is shown in Fig. 1. The point who has the most orientations pointing to it is considered as the vanishing point of the road. To estimate the vanishing point, we need to calculate all the local texture orientations (blue arrows in Fig. 1) and check which point has most orientations pointing to it.

So the first step of vanishing point estimation is to get the texture orientation. The local texture orientation \( \theta(p) \) of an image at pixel \( p(x, y) \) describes the strongest local parallel structure or texture flow [8]. Different methods were proposed for estimating texture orientation. C. Russmusen, Kong, Moghadam et al. [4], [5], [15] used Gabor filters to estimate the texture orientation; M. Nieto et al. used steerable filters [10].
filter banks to estimate it [14]; and Miksik, Wang et al. used Haar-like box to do it [8], [11]. Here we mainly follow the Gabor filter based method. A Gabor filter is a Gaussian kernel function modulated by a sinusoidal plane wave, image analysis with Gabor filters is thought to be similar to perception in the human visual system [12]. The mathematical form of Gabor filter is as follows [13]:

$$\psi_{\omega,\phi}(x, y) = \frac{\omega}{\sqrt{2\pi}c} e^{-\omega^2(4a^2+b^2)/8c^2}(e^{ia\omega} - e^{-\omega^2/2})$$

where \(a = x\cos\phi + y\sin\phi \) and \(b = -x\sin\phi + y\cos\phi\), \(c\) is a constant, \(\omega\) and \(\phi\) are the size and orientation of the kernel. Similar to [15], kernels with 36 even distributed orientations are used to estimate the texture orientation. For each pixel, the largest filtered result of 36 orientations is considered as the most prominent orientation, only pixels whose most prominent orientation is significantly dominant than other orientations are considered having a texture orientation. This will help us filter out unoriented pixels, for example, the sky pixels. Only oriented pixels are treated as voters in the vote for vanishing point.

3. Vanishing Point Voting

After the dominant orientation of each pixel is calculated, this information is used to vote for the vanishing point. The key idea of the voting strategy goes this way: the smaller the angle (\(\alpha\) in Fig. 2) between the voter’s texture orientation (\(\overrightarrow{vo}\) in Fig. 2) and the direction from the voter to the candidate (\(\overrightarrow{vc}\) in Fig. 2), the higher the voting score will the candidate get. To increase the performance, for a vanishing point candidate, only the pixels whose distance to the candidate is within a threshold are considered as voters. Below is the voting function given in [15]:

$$\text{Vote}(V, C) = \begin{cases} 
\frac{1}{1 + \alpha^2 \cdot d(V, C)^2} & \text{if } \alpha \leq \frac{5}{1 + \alpha \cdot d(V, C)} \text{ and } d(V, C) \leq 0.35 \\
0 & \text{otherwise}
\end{cases}$$

where \(V\) represents the voter, \(C\) represents the candidate, \(d(V, C)\) equals to the distance between \(V\) and \(C\) divided by the diagonal length of the image, the meaning of \(\alpha\) is shown in Fig. 2, which represents the angle between the orientation of the voter and the direction from the voter to the candidate. For a given candidate \(C\), the final voting score is the summation of all eligible voters’ contribution which is defined by Eq. (2).

As mentioned in Sect. 1, conventional voting methods consider almost every pixel in the road scene image as a vanishing point candidate [5], [15], [17]–[19], and calculate its voting score. To find out the vanishing point, we one needs to go through every candidate and picks out the one with highest voting score, which makes the whole procedure very time consuming. Authors of [15], [17] reduce the voters by a locally adapted soft-voting scheme, and reduce the candidates by ignoring upper 10% of the image, but it is still far from real-time processing. For a typical 240×180 image, it takes more than 1 minute to complete the whole voting.

Actually there is no need to consider every pixel as a candidate since we are only interested in the candidate whose voting score is highest. So if we can find the target candidate without checking all the candidates, we can save the time. Thus we propose a heuristic voting method instead of conventional exhaustive voting.
Fig. 4 Concept of moving of a searching point by PSO. $S^k$ represents the current searching point. $S^{k+1}$ represents the moved searching point. $V^k$ represents the current velocity. $V^{k+1}$ represents the moving velocity. $V_{pbest}$ represents the velocity based on pbest. $V_{gbest}$ represents the velocity based on gbest.

Figure 3 shows an example of the vanishing point voting score distribution. The brightness in the distribution map represents the voting score, the brightest pixel shows the location of vanishing point. From the map we can observe that there exists some certain pattern of the vanishing point score distribution, though we don’t know what exactly the pattern is. So the problem becomes searching an optimal point from an unknown pattern. Then we think of PSO (particle swarm optimization).

3.1 PSO Based Voting Strategy

Particle swarm optimization (PSO) is a heuristic computational method which was developed in 1995 by Kennedy and Eberhart [16]. It can search for the optimal solution in a large candidate space by evaluating and moving a swarm of particles. Each particle is treated as a point in a N-dimensional space which adjusts its moving according to its own moving experience as well as the moving experience of other particles. And each particle keeps track of its coordinates in the solution space which are associated with the ‘best’ solution that has achieved so far by that particle. To be precise, here ‘best’ means the highest fitness value according to the fitness function. The fitness function is a function of particle’s position, which is used for evaluating particles. Whenever a particle moves to a new position, its fitness will be updated by the fitness function. Each particle’s best position in its searching history is recorded as ‘pbest’ (particle’s best). And the best one of all particles’ ‘pbest’ is recorded as ‘gbest’ (global best). Both ‘pbest’ and ‘gbest’ are updated in every iteration. The basic concept of PSO lies in accelerating each particle toward its pbest and at the gbest locations, with a random weighted acceleration at each time step as shown in Fig. 4. Figure 5 shows the flowchart of PSO. The velocity function is given below:

$$V_{i}^{k+1} = \omega V_{i}^{k} + c_{1} \cdot r_{1} \cdot (pbest_{i} - S_{i}^{k}) + c_{2} \cdot r_{2} \cdot (gbest_{i} - S_{i}^{k})$$  (3)

where $r_{1}$ and $r_{2}$ are two random numbers within [0,1], $c_{1}$ and $c_{2}$ are constants (here are both set to 2). The following function is usually utilized in Eq. (3):

$$\omega = \omega_{max} - \left[ (\omega_{max} - \omega_{min}) \times \frac{iteration}{iteration_{max}} \right]$$  (4)

where $\omega_{max}$ is the initial weight, $\omega_{min}$ is the final weight, $iteration_{max}$ is the maximal iteration number, $iteration$ is the current iteration number. And

$$S_{i}^{k+1} = S_{i}^{k} + V_{i}^{k+1}$$  (5)

A large inertia weight ($\omega$) facilitates a global search while a small inertia weight facilitates a local search. By linearly decreasing the inertia weight from a relatively large value to a small value through the course of the PSO run gives the best PSO performance compared with fixed inertia weight settings [20].

To apply PSO in the vanishing point voting, the searching space of particles is the whole image plane, each position of the particle represents one vanishing point candidate. The fitness function in PSO is set to be the voting score of the vanishing point candidate. More precisely, the fitness function is given below:

$$f(C) = \sum_{i=1}^{n} Vote(V_{i}, C)$$  (6)
where $Vote(V_i, C)$ has the same meaning with Eq. (2), the subscript $i$ means the $i$-th voter, and $n$ represents the total number of voters.

4. Performance Optimization for Sequenced Images

Since our vanishing point estimation serves for the autonomous driving system, it is more meaningful to estimate the vanishing point on a sequence from a video source rather than on a single image. For the straight road situation, the vanishing point will rarely change its position as the car moves; and even for a curved road, the vanishing point will move gradually, without sudden change between neighboring frames. So the estimation result of the current frame is a strong cue for the estimation of the next frame. To make good use of this inter-frame information, both efficiency and robustness of vanishing point estimation can be improved.

4.1 Efficiency Improvement

The initial particles are randomly generated in the original PSO method. While for sequenced images, the vanishing point positions of neighboring frames will not change significantly. So if this inter-frame information can be brought to the particles, the searching will be significantly improved.

There are two steps to use the inter-frame information:

Step 1: When initialize the particles, particles are randomly distributed within a certain distance to the previous vanishing point’s location rather than distributed in the whole image plane. Usually the distance is set to be 15% of the image’s diagonal. This step “tells” the particles to search around result of previous frame.

Step 2: Decrease the $\omega$ in Eq. (3). As mentioned in last section, a large inertia weight ($\omega$) facilitates a global search while a small inertia weight facilitates a local search. Since the vanishing point of current frame will be close to that in previous frame, we want a local search more than a global search. To decrease the $\omega$ in Eq. (3), we just decrease both $\omega_{\text{max}}$ and $\omega_{\text{min}}$ in Eq. (4).

Figure 6 shows the flowchart of the PSO improved by inter-frame information.

4.2 Robustness Improvement

The estimation of the previous vanishing point will not only help to improve the efficiency of current estimation, but also improve the robustness. By introducing a weight in the vanishing point voting we can achieve this. Under the assumption that the location of the vanishing point won’t change much in two neighboring frames, we can set a certain distance, for candidates within this certain distance, the voting score won’t be suppressed; while for those who are far from the previous vanishing point, the voting score will be suppressed according to the distance. A longer distance will result in a smaller voting weight. Equation (7) shows the weight function,

$$w(C, V_{\text{pre}}) = \begin{cases} e^{-\frac{d^2 - \lambda^2}{\sigma^2}} & \text{if } d \geq \lambda \\ 1 & \text{otherwise} \end{cases}$$

(7)

where $\sigma$ and $\lambda$ are constants. And $\lambda$ is the Euclidean distance ($d$) between the current candidate ($C$) and previous vanishing point ($V_{\text{pre}}$), normalized by the image size, which is defined as $\tilde{d} = d / D$, where $D$ is the diagonal length of the image. In Eq. (7), $\sigma^2$ is set to be 1.3 and $\lambda$ is set to be 0.05, with this parameter setting, the weight will range between 0.5 and 1. Both of these two parameters are tunable, $\sigma$ determines decay rate of the weight, and $\lambda$ represents the predicted area of vanishing point. Finally Eq. (6) will become the following equation:

$$f(C) = w(C, V_{\text{pre}}) \times \sum_{i=1}^{n} Vote(V_i, C)$$

(8)
5. Parameter Tuning and Performance Evaluation

5.1 Experiments on Single Image

To apply PSO in vanishing point voting, we need to determine the particle number and iteration times. More particles and iterations will increase the rate of finding the optimal point (i.e. the vanishing point), while at the same time will increase the searching time. To make a good compromise between accuracy and efficiency, experiments were conducted. We tested on 30 different road scenes which contain desert road, forest road, field road, snow covered road and so on. For a certain pair of particle number and iteration number, 200 times tests were conducted to get the average accuracy and speed. To evaluate accuracy of proposed method, besides the true accuracy, the match rate with the result of conventional method was also taken into account. It is because the match rate with result of conventional method reflects the PSO’s ability of locating the optimal point. Figure 7 shows accuracy, match rate and speed under different parameter settings. To trade off the influence of different hardware, we use the ratio between the processing time of Kong’s method [15] and PSO based method to measure the speed, as shown in Eq. 9, where $T_{\text{con}}$ and $T_{\text{PSO}}$ are processing time of conventional method and PSO based method respectively. The criterion of accurate is, if the distance between PSO based result and real vanishing point is within 2% (normalized by the image size), we treat it as accurate; and the criterion of match rate with result of conventional method is similar. Note that in some situations, even if the PSO based result is far away from the conventional result, the PSO based result is still accurate (mentioned in Sect. 5.2).

$$\text{Speed} = \frac{T_{\text{con}}}{T_{\text{PSO}}}$$  \hspace{1cm} (9)

Figure 8 shows an example of applying PSO in vanishing point voting. The top shows the 3-D distribution of the voting score of all the pixels (the original image and 2-D distribution are shown in first column of Fig. 3), the height represents the voting score which is calculated by Eq. (6), the horizon plane maps to image plane; the bottom shows the particles of PSO, each circle represents one particle, the color marks the iteration index of the particle. For the conventional exhaustive voting, we need to calcu-
late the voting score of each point in the image, as shown in the top of Fig. 8; while for the heuristic PSO based voting, we only need to calculate some discrete points in the image, as shown in the bottom of Fig. 8. This results in a great improvement in efficiency. In Fig. 8’s example, the original image is at the size of 240×180, for conventional exhaustive voting, we need to calculate 240×180 = 43200 points, while for a PSO with 10 particles and 20 iterations, we only need to calculate 200 points, which brings us about 200 times efficiency improvement in voting. Figure 9 shows some example of comparison between conventional method and PSO based method. And Table 1 shows the executing time of these two methods for different road scenes. From these examples we can see that the PSO based method can get almost the same result as conventional method, but with a much higher speed.

5.2 Experiments on Sequenced Images

For sequenced images input, both accuracy and efficiency are improved by making full use of inter-frame information. Comparing with estimating on single image, the greatest advantage of using inter-frame information is it can perform well when the road scene is complicated and the point with maximal votes may not be the vanishing point. So in the experiment of sequenced images, a video of driving in the countryside with bad weather is used as the image source. In the test images, the road is ponded and unstructured, surrounded by trees; the sky is grained due to the clouds. These complicated surroundings will generate extra noisy textures for the estimation.

Figure 10 shows an example of accuracy improvement by introducing an weight to the vanishing point voting. The first two rows show the estimation without inter-frame information. From left to right are original images, vanishing point score distributions and vanishing point estimation results (red ones are result of conventional method, blue ones are result of PSO based method) respectively. And row 1 and row 2 are two neighboring frames from a video. And row 3 shows the estimation of the second frame with information from the first frame. From left to right are the weight mask, vanishing point score distribution (equals to second row’s distribution multiplied by the weight mask) and estimation result.

From Fig. 10 we can see that the complicated road scene will lead to false estimation of the vanishing point, for example, the trees in the second frame lead to a false estimation (red point in row 2 column 3). From the vanishing point score distribution (row 2 column 2) we can see that even though the location of true vanishing point still has a high score, the disturbance’s score is higher. It is difficult to handle this kind of situation when dealing with single frame. While for sequenced images, we can suppress this kind of disturbance which is a little far away from the true vanishing point. As shown in the left bottom of Fig. 10, a weight mask is calculated according to the estimation result of previous frame using Eq. (7), the brightness represents the weight, the weight of the brightest area around previous vanishing point is 1, and decreases as the distance increases. After we recalculate the vanishing point score distribution with the weight mask, we can see that the disturbance is suppressed and the

<table>
<thead>
<tr>
<th>NO.</th>
<th>PSO</th>
<th>conv.</th>
<th>NO.</th>
<th>PSO</th>
<th>conv.</th>
<th>NO.</th>
<th>PSO</th>
<th>conv.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.515</td>
<td>68.7</td>
<td>11</td>
<td>0.479</td>
<td>69.0</td>
<td>21</td>
<td>0.497</td>
<td>69.7</td>
</tr>
<tr>
<td>2</td>
<td>0.546</td>
<td>70.1</td>
<td>12</td>
<td>0.469</td>
<td>69.0</td>
<td>22</td>
<td>0.504</td>
<td>69.4</td>
</tr>
<tr>
<td>3</td>
<td>0.486</td>
<td>68.6</td>
<td>13</td>
<td>0.529</td>
<td>69.6</td>
<td>23</td>
<td>0.511</td>
<td>69.8</td>
</tr>
<tr>
<td>4</td>
<td>0.535</td>
<td>69.6</td>
<td>14</td>
<td>0.464</td>
<td>66.6</td>
<td>24</td>
<td>0.456</td>
<td>69.7</td>
</tr>
<tr>
<td>5</td>
<td>0.506</td>
<td>69.4</td>
<td>15</td>
<td>0.492</td>
<td>69.5</td>
<td>25</td>
<td>0.482</td>
<td>69.7</td>
</tr>
<tr>
<td>6</td>
<td>0.461</td>
<td>69.7</td>
<td>16</td>
<td>0.483</td>
<td>67.5</td>
<td>26</td>
<td>0.495</td>
<td>69.7</td>
</tr>
<tr>
<td>7</td>
<td>0.517</td>
<td>69.4</td>
<td>17</td>
<td>0.502</td>
<td>70.5</td>
<td>27</td>
<td>0.512</td>
<td>69.9</td>
</tr>
<tr>
<td>8</td>
<td>0.468</td>
<td>69.0</td>
<td>18</td>
<td>0.516</td>
<td>69.8</td>
<td>28</td>
<td>0.503</td>
<td>69.3</td>
</tr>
<tr>
<td>9</td>
<td>0.438</td>
<td>66.8</td>
<td>19</td>
<td>0.516</td>
<td>70.0</td>
<td>29</td>
<td>0.487</td>
<td>68.7</td>
</tr>
<tr>
<td>10</td>
<td>0.493</td>
<td>69.1</td>
<td>20</td>
<td>0.502</td>
<td>69.8</td>
<td>30</td>
<td>0.493</td>
<td>69.7</td>
</tr>
</tbody>
</table>

PSO average time | 0.4950 |
conv. average time | 69.25 |
It is worth to point out that the PSO result of row 2 (blue point) appears in the location of true vanishing point is a coincidence. It is because the score of the true vanishing point is also a local maximum and very close to the global maximum (red point), PSO failed to find the global optimization in this situation. Similar situations happen occasionally in the experiments.

Since we use previous result as an indicator, it is obviously that if the previous result is not correct, it will have a bad influence on the current estimation, especially when the wrong estimation is far from the correct position. So the quality of the first frame is the most significant. Figure 11 shows an example of wrong estimation misled by the former frames. From figure we can see the negative influence of former frames’ wrong estimation, estimation error in the 1st frame misleads the estimation of the 2nd and 3rd frames which are almost accurate without inter-frame information. To deal with this problem, we can use the conventional exhaustive method for the first frame to improve the reliability as much as possible, or even indicate the true vanishing point manually.

To verify the validity of using inter-frame information, we tested 260 frames of sequenced images from the video mentioned in the beginning of this subsection. Four methods were tested, they were conventional method, conventional method with inter-frame information, PSO based method, PSO based method with inter-frame information. And considering the randomness of PSO, two PSO based methods were repeated 20 times. And When using the inter-frame information, the the first frame was estimated by the conventional method. The criterion of accuracy was, if the distance between estimated vanishing point and manually marked vanishing point was within 5% (normalized by the image size), we considered the estimation as accurate. The particle number and iteration number in PSO were set as 10 and 30 respectively. Figure 12 shows the performance comparison of different methods. From the figure we can see that by using inter-frame information, the accuracy of the estimation is significantly improved; and by using PSO, the runtime is significantly reduced.

Besides the accuracy, efficiency can also be improved by initializing the particles according to previous result and decrease the $\omega$ in Eq. (3). Figure 13 shows the performance comparison of using and not using inter-frame information for the same road scene, 200 times repeated. In the top chart, the particle number is fixed to 5, the horizon axis is the iteration times; in the bottom chart, the iteration number is fixed 30, the horizon axis is the particle number. The vertical axis is the match rate with the result of conventional method. Here accuracy is not used as measurement because the accuracy of only one road scene is not meaningful and the purpose of these two charts is to demonstrate inter-frame information can help PSO finding the optimal point. From the two charts we can see that by using inter-frame information in PSO, we can get more accurate result with same
particle number and iteration number. In other words, we can use less particles and iterations to achieve same performance as single image based estimation, so as to improve the efficiency.

6. Conclusion

In this paper we point out the efficiency problem of conventional vanishing point estimation methods and propose a new PSO based voting method to solve it. The proposed method is evaluated by comparing it with the conventional method on both vanishing point’s location and processing time. Experiments show that the PSO based method can significantly improve the efficiency with little loss in accuracy. We also propose an inter-frame information based method to reduce the loss in accuracy by the PSO based method, and further improve the efficiency. Experiments show that the proposed method can not only improve the efficiency of the conventional method, but also improve the robustness in some complicated situations that conventional method cannot perform well.

References

Xun Pan was born in Zhenjiang, China, on February 23th, 1989. He received a Bachelor degree of Science from Nanjing University (China) in 2011 and a Master degree of Engineering from Waseda University in 2015. He is presently a Doctor student in the Graduate School of Information, Production and Systems, Waseda University. He has worked on vanishing point estimation and sequenced images optimization.

Wa Si was born in Hefei, China, on May 6th, 1987. She received a Bachelor degree of Science from Fudan University (China) in 2007 and a Master degree of Engineering from Waseda University in 2010. She is presently a Ph.D student in the Graduate School of Information, Production and Systems, Waseda University. She has worked on parameter tuning and performance optimization of PSO.

Harutoshi Ogai was born in Kitakyushu, Japan, on March 1st, 1952. He received a Ph.D degree of Engineering from Tokyo Institute of Technology in 1997 and is presently a professor in the Graduate School of Information, Production and Systems, Waseda University. His research area includes process control, process modeling and simulation, process analysis and data mining, automatic control and bridge diagnosis. He is also member of SICE, ISCIE, ISIJ, JSAI and IPSJ.