Real-time Stereo Vision System at Tunnel

Yuquan Xu[1], Seiichi Mita[1], Hossein Tehrani[2] and Kazuhisa Ishimaru[3]
[1] *Research Center for Smart Vehicles, Toyota Technological Institute, 2-12-1 Hisakata, Tempaku, Nagoya, Japan*
[2] *Driving Assist & Safety Eng. Div. 1, DENSO Corporation, 1-1, Showa cho, Kariya, Japan*
[3] *Research & Development Dept. 2, Nippon Soken Inc., Nishio, Aichi, Japan*

Keywords: Stereo Vision, Image Deblurring, Optical Flow, Cepstrum.

Abstract: Although stereo vision has made great progress in recent years, there are limited works which estimate the disparity for challenging scenes such as tunnel scenes. In such scenes, owing to the low light conditions and fast camera movement, the images are severely degraded by motion blur. These degraded images limit the performance of the standard stereo vision algorithms. To address this issue, in this paper, we combine the stereo vision with the image deblurring algorithms to improve the disparity result. The proposed algorithm consists of three phases: the PSF estimation phase; the image restoration phase; and the stereo vision phase. In the PSF estimation phase, we introduce three methods to estimate the blur kernel, which are optical flow based algorithm, cepstrum base algorithm and simple constant kernel algorithm, respectively. In the image restoration phase, we propose a fast non-blind image deblurring algorithm to recover the latent image. In the last phase, we propose a multi-scale multi-path Viterbi algorithm to compute the disparity given the deblurred images. The advantages of the proposed algorithm are demonstrated by the experiments with data sequences acquired in the tunnel.

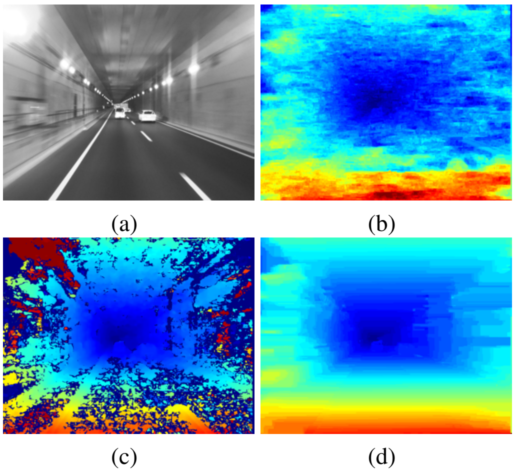
1 INTRODUCTION

In recent years, significant attention has been paid to the development of autonomous vehicles and advanced driver assistance systems (ADAS). Stereo vision is an important research problem for ADAS applications and has received a great deal of attention over the past decade. However, there is little work considering the stereo vision in some challenge environments, such as tunnels or low lightening condition.

For the ADAS applications, the stereo camera pair, mounted on the vehicle, is used to estimate the environment's depth information from images acquired by the stereo pair. However, in certain environments, such as the tunnel, the low illumination and the fast camera movement, result in the blurring of the stereo pair images. The blurry images reduce the quality of stereo's depth estimation. The motion blur is caused by the relative movement between the camera and the object during the camera's exposure time. The blur can be attributed to three tunnel-specific factors in ADAS applications. 1) The speed of the car, which is typically very high, since there are no traffic lights or sharp turns in the tunnel. 2) The distance between the camera and the object. The tunnel area is very limited and the walls and roof of the tunnel are near to the vehicle. 3) The length of exposure time, which is high to account for the low illumination of the tunnel. Stereo vision can be viewed as a matching problem and for blurry scenes there may be multiple matching pixels across the left and right image pair. This phenomenon violates the basic assumption of the stereo vision framework. Fig. 1 shows the comparison results of a tunnel example using the proposed algorithm and other state-of-the-art real-time stereo vision algorithms, Semi-Global Block-Matching Algorithm (SGM) (Hirschmüller, 2008) and Multi-paths Viterbi (MPV) (Long et al., 2014b). In Fig. 1, we can see that the input image is highly blurred especially on the left wall, and the previous stereo vision approaches can not reliably estimate the disparity for such images. However, the proposed algorithm can produce high quality disparity results, even for the challenging scenes.

A straight forward approach to address this issue involves the deblurring of the degraded input stereo pairs to improve the disparity results. Motion deblurring is an important research problem in low-level vision research. There are two research sub-problems that are addressed in this research. The first problem involves the estimation of the Point Spread Function

Xu Y., Mita S., Tehrani H. and Ishimaru K.
Real-time Stereo Vision System at Tunnel.
DOI: 10.5220/0006112304020409
In *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2017)*, pages 402-409
ISBN: 978-989-758-227-1



(a) (b)

(c) (d)

Figure 1: Visual comparison of our method to SGM (Hirschmüller, 2008) and MPV (Long et al., 2014b) on an example captured in the tunnel. (a) is the left image of the stereo pair. (b) MPV (Long et al., 2014b). (c) SGM (Hirschmüller, 2008). (d) The proposed method. In the disparity map, red represents near objects and blue represents distant objects.

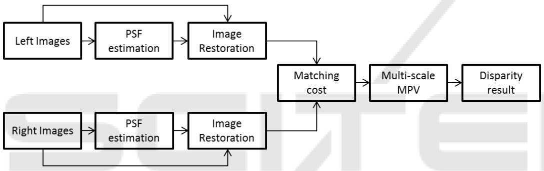


Figure 2: The flowchart of the proposed algorithm.

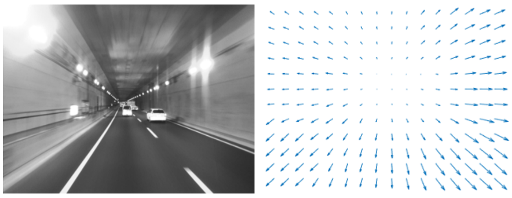
(PSF) or so-called blur kernel which represents the degradation process. The second problem involves the recovery of the unknown latent image. Typically, most of the existing deblurring methods (Cho and Lee, 2009; Xu and Jia, 2010) aim to deal with the spatial-invariant blurs which are modeled as a deconvolution problem. However, in the tunnel case, the blur is spatially-variant. Furthermore, for ADAS applications, the deblurring and disparity estimation should be performed in real-time, that means the running time should be lower than 100ms. However, other deblurring algorithms usually cost minutes to deblur one image, which is unacceptable for real-time application.

In this article, we introduce a real-time algorithm to deal with the tunnel stereo vision problem. The proposed algorithm consists of three phases, i.e., PSF estimation phase, image restoration phase and stereo vision phase. In the PSF estimation phase, the traditional iteration based framework's computational cost is reduced by utilizing three different algorithms to directly estimate the spatially-variant blur kernel by using optical flow, cepstrum and a simple constant kernel. After we estimate the PSF of the image, a fast non-blind image deblurring algorithm is introduced to remove the blur effect of the images. Last we propose a multi-scale MPV algorithm to compute the disparity result. Our proposed approach is real-time which is significantly important for practical applications. We show the flowchart of the proposed algorithm in Fig. 2

2 RELATED WORK

Stereo vision is one of the basic problems in computer vision and has made tremendous progress in last decades. Scharstein and Szeliski (Scharstein and Szeliski, 2002) introduced a categorization scheme for stereo algorithms, which classified the various stereo algorithms into two classes including local and global methods. Local algorithms usually contain four steps: (1) matching cost computation; (2) cost aggregation; (3) disparity computation; (4) the optional post processing. The matching cost computation methods include standard window-based algorithms, normalized cross correlation (NCC), Census Transform algorithm (Zabih and Woodfill, 1994), and recently CNN based cost (Zbontar and LeCun, 2014). The cost aggregation algorithms contain unnormalized box filtering, bilateral filter and guided image filter (He et al., 2013). On the other hand, the global methods make explicit smoothness assumptions and then solve an optimization problem from the matching cost and omit the cost aggregation step. The optimization algorithms include dynamic programming (Veksler, 2005), belief propagation (Sun et al., 2003), Viterbi (Long et al., 2014a; Long et al., 2014b) and graph cuts (Boykov et al., 2001). SGM (Hirschmüller, 2008) is a hybrid of local and global method that expands the single-directional 1D scanline optimization process to multidirectional 1D scan-line optimization.

Removing the blur effects of the images is a well-studied problem. For spatial-invariant blind image deblurring issue, Fergus et al. (Fergus et al., 2006) firstly proposed a successful variational Bayesian framework to estimate the complex motion blur kernel. In (Cho and Lee, 2009), a fast deblurring method was proposed by introducing a prediction step. The similar step was used in (Xu and Jia, 2010) (Xu et al., 2012). For spatial-variant blur issue, Joshi et al. (Joshi et al., 2010) introduced inertial measurement sensors to record the motion path of the camera to recover the latent image. Then the projective motion-blur model was introduced by (Tai et al., 2011) and used in (Whyte et al., 2011) (Hirsch et al., 2011). Zheng et al. (Zheng et al., 2013) proposed a forward motion model which is more relevant to our work, but



(a) (b)

Figure 3: (a) is the blurry input, (b) is the blur kernel in each local area of the image.

their work is too time-consuming for practical application. We refer the readers to the (Wang and Tao, 2014) for the recent progress of the image deblurring.

3 STEREO VISION ALGORITHM

In this section, we introduce a coarse-to-fine framework to estimate the disparity result based on the deblurred images. We adopt the structural similarity (SSIM) to measure the matching cost between left and right deblur images.In this coarse-to-fine framework, we generate the 3 layers Gaussian pyramid of the deblurred images, and each coarser layer is half size of the finer layer. Then we utilize the MPV (Long et al., 2014a) method to estimate the disparity result from the coarsest to the finest layer. The cost function of proposed algorithm is:

$$E(D) = E_D(D) + E_{TV}(D) \tag{1}$$

Where D represents the disparity result, $E_D(D)$ represents the SSIM cost of deblurring images and $E_{TV}(D)$ represents the TV constraints of disparity.

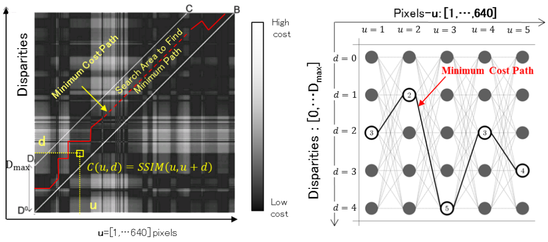


Figure 4: Viterbi path to find optimum disparity.

The problem of stereo matching can be modeled as finding the disparity map D that minimizes the energy function E(D). In this case, the Viterbi algorithm can be used to search the optimum solution (Son and Seiichi, 2006). We build a graph to represent the disparities of the image pixels along one line, where each node in this graph represents a disparity assigned to a pixel and each edge represents a candidate disparity change between two pixels in the same path. Viterbi algorithm operates on individual rows and find the optimum "path" of nodes through disparity space from one side of the image to the other as shown in Fig 4. To find the optimum disparity value, we solve following cost function:

$$d_t = \arg\min_d V(t-1, d_{t-1}) + SSIM(t,d) + \lambda|d - d_{t-1}| \tag{2}$$

where d_t represents the disparity at pixel t, $V(t-1, d_{t-1})$ represents the energy of a node with pixel $t-1$ with disparity d_{t-1}, SSIM(t,d) represents the SSIM cost with pixel t and disparity d and $\lambda|d - d_{t-1}|$ is the smoothing penalty term.

To increase the robustness of the Viterbi algorithm, four bi-directional (horizontal, vertical, and 2 diagonals) Viterbi paths are used to provide good coverage of the 2D image. The flowchart of the Multi-path Viterbi algorithm is shown in Fig. 5.

With the results on the coarser layer, we only consider a small candidate disparity range. This multi-scale MPV algorithm is much faster than (Long et al., 2014a) and more stable to get better results on some textureless area.

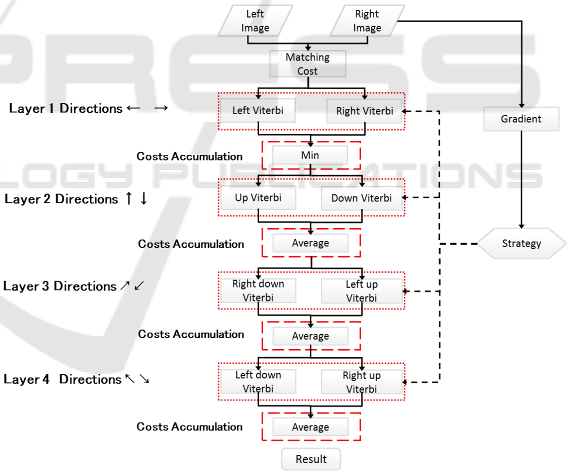


Figure 5: The flowchart of the Multi-path Viterbi algorithm.

4 PSF ESTIMATION

In this section, we estimate the motion kernel of the tunnel images. For a given image with long exposure time, the estimating the motion blur kernels are not trivial. Fortunately in the tunnel case, the exposure times are limited by the frame rate. Additionally, the vehicle typically travels in a straight path in the tunnel. Therefore, the motion observed during the exposure will be relatively smooth and the PSF in a small local area can be approximated as the linear kernel, as

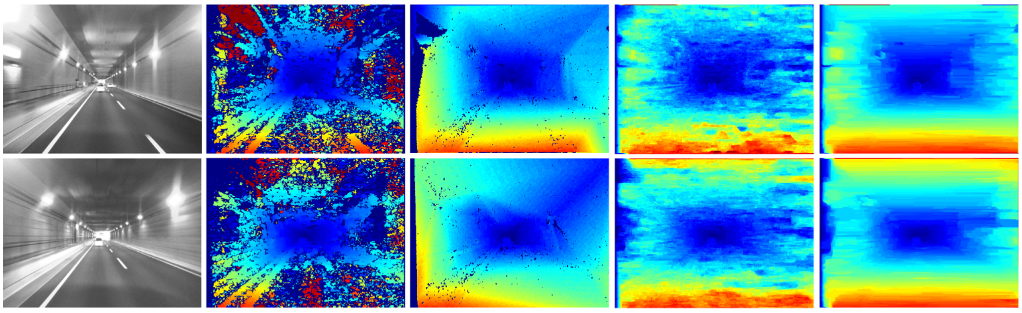


Figure 6: The first column is the original image. The second to last column are the disparity results of SGM (Hirschmüller, 2008), ELAS (Geiger et al., 2011), MPV (Long et al., 2014b) and the proposed method respectively.

shown in Fig. 3. This approximation helps in the case of the spatially-varying blur since estimating an arbitrary kernel for each individual pixel would be prohibitively expensive. Based on this assumption, the blur kernel at each pixel is a straight line which is determined by two parameters, i.e., the length and the angle. Then the blurry input can be formulated as:

$$b = \frac{1}{Z}\sum_{i}[(\omega_i * l) \otimes f_i] + n \tag{3}$$

where b is the blurry input, ω_i is the window function to select a sub-region of the image, l is the latent image we want to recover, i is the index of the sub-region, $*$ denotes the element-wise multiplication, $\otimes$ denotes the convolution process, Z is normalization factor of the window function, n is the additive noise, f_i represents the blur kernel of the $i-th$ sub-region of the image, respectively. For the PSF, we have:

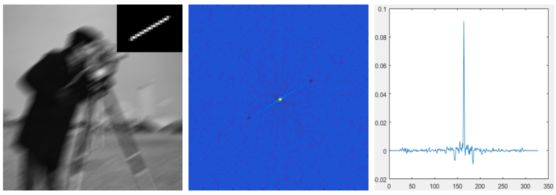
$$f(x,y) = \begin{cases} 1/L & others \\ 0 & y = x\tan\theta, \quad 0 \leq x \leq L\cos\theta \end{cases} \tag{4}$$

where x and y are pixel index and L is the length of the kernel, θ is the angle of the kernel, respectively. Then we introduce three algorithms to estimate this kind of blur kernel for the blurry input.

4.1 Optical Flow

Since the motion blur is caused by the relative motion of the camera and objects, it follows that the knowledge of the actual camera motion between consecutive image pairs that can provide significant information when performing image deblurring. We assume that the camera moves nearly along a straight line between two successive frames and the optical flow of these two frames can represent the blur kernel for the blurry input. To compute the optical flow, we have the cost function as Equ. (5)

$$E(\mathbf{w}) = E_D(\mathbf{w}) + \lambda E_S(\mathbf{w}) \tag{5}$$



(a) (b) (c)

Figure 7: (a) is the blurry image and the blur kernel in its right up. (b) is the cepstrum of blurry input. (c) is the value alone the blur angle in the cepstrum.

where $\mathbf{w}$ is the flow vector, λ is the smoothing parameter, $E_D(\mathbf{w})$ is the data term and $E_S(\mathbf{w})$ is smoothing term, respectively. For $E_D(\mathbf{w})$ and $E_S(\mathbf{w})$ we have:

$$\begin{aligned} E_D(\mathbf{w}) = \|I_2(\mathbf{u}+\mathbf{w}) - I_1(\mathbf{w})\|^2 + \\ \alpha\|\nabla I_2(\mathbf{u}+\mathbf{w}) - \nabla I_1(\mathbf{w})\|^2 \end{aligned} \tag{6}$$

$$E_S(\mathbf{w}) = |\nabla \mathbf{w}| \tag{7}$$

where I_1 and I_2 are two successive frames, $\nabla = (\partial_x, \partial_y)$ represents the gradient of the image, $\alpha \in [0,1]$ denotes a linear weight. The smoothing term is the Total Variation (TV) constraint. We use the Successive Over-Relaxation (SOR) method to solve Equ. (5). Since the input image is blurry and we actually don't need the optical flow for every pixel of the image [1], we downsample the input left and right image to 1/4 size which can not only reduce the blurry effect in the image but also reduce the computation time of the algorithm. Then we upsample the optical flow to represent the blur kernel of the image.

4.2 Cepstrum

To deblur the image, we first segment the blurry input into several overlapped patches and assume that in each sub-region the blur kernel is a straight line.

[1] Actually we deblur the input image block by block, so we only need an optical flow for every small local area of the input, not every pixel

Based on this assumption, we can compute the cepstrum of the image to identify the angle and length of this kind of linear kernel. The cepstrum of an image is defined as (Rom, 1975):

$$C(g) = F^{-1}(log(|F(g)|) \tag{8}$$

where F and F^{-1} denote the Fourier transform and inverse Fourier transform , $C(g)$ is the cepstrum of the image g. In practice, the cepstrum of the image is usually expressed as:

$$C(g) = F^{-1}(log(|F(g)|+1) \tag{9}$$

This linear degrade process for each sub-patch is described as Equ. (10)

$$s_b = s_l \otimes f + n \tag{10}$$

where s_b and s_l denote the sub-patch of the blurred and clear image.

In the case of ignoring the additive noise n, we can formulate Equ. (10) to cepstrum domain as:

$$C(s_b) = C(s_l) + C(f) \tag{11}$$

where $C(s_b)$, $C(s_l)$ and $C(f)$ represent the cepstrum of blurred image s_b, s_l and f. Therefore the cepstrum of blurred image is modeled as the sum of the cepstrum of the clear image and the cepstrum of a blur kernel, which means the convolution operation becomes additive in cepstrum domain, thus the blur is easy to detect.

Fig. 7 shows an example of how we use the cepstrum to detect the blur kernel of the image. Fig. 7 (a) shows a blurry and noisy image. The blur kernel has 20 pixels length and 30-degree angle and the variance of the Gaussian noise is 0.03. We show the cepstrum of the blurry input in Fig. 7 (b), in which we can clearly see a straight line along the 30-degree. In practice, we adopt the Radon transform to detect this angle as the blur angle and then we rotate the cepstrum to make this line to be horizontal. Fig. 7 (c) shows the values on this line of the cepstrum and we can see two negative peaks in this line. We compute the distance between these two negative peaks and the blur length is half of the distance. Finally, we can estimate the angle and length of the blur kernel. For a blurry tunnel input, we segment it to some overlapped cloacal areas and use the cepstrum to estimate the blur kernel of this sub-region.

4.3 Proposed Method

In autonomous vehicle applications, the stereo vision algorithm is required to generate the disparities in real-time continuously and stably. Although we have proposed two algorithms to estimate the blur kernel, the computational time is not real-time or less than 100ms. Therefore, we further reduce the computational complexity of the deblurring process for the stereo pairs. More specifically, we adopt a constant kernel as an approximate motion kernel for all the pixels of the tunnel images as shown in Fig. 8 (d). In this constant kernel, we pre-define a point in the image to represent the vanishing point along the traveling direction of the vehicle and the angle of the blur kernel in each subregion can be determined by the vector from the vanishing point to the central point. The length of the kernel is determined by the vehicle speed and the shutter speed, and we can read from the camera and the vehicle CAN. In Fig. 8, we can see that using this simple kernel the disparity result is similar for the optical flow and cepstrum methods. We also compare the computation time of these three algorithms including the image restoration and stereo vision phase in Table 1 and size of the input stereo image is 640 × 480.

Table 1: Computation time.

Methods	Time	Environment
Optical flow	9.8s	Intel i7 4790 CPU
Cepstrum	7.8s	Intel i7 4790 CPU
Proposed	1.4s	Intel i7 4790 CPU
Proposed	96ms	GTX TITAN X GPU

5 NON-BLIND DEBLURRING ALGORITHM

After we get the PSF estimation, we define the following cost function to restore the clear image:

$$E(l) = ||b - \frac{1}{Z}\sum_i (\omega_i * l) \otimes f_i||^2 + \gamma|\nabla l| \tag{12}$$

where γ denotes the smoothing parameter.

In practice, we divide the input blurry input into several overlapping parts and in each part we deblur the image by solving Eq. 13

$$E(s_l) = ||s_b - s_l \otimes f_i||^2 + \gamma|\nabla s_l| \tag{13}$$

We utilize the half-quadratic penalty method (Krishnan and Fergus, 2009) to recover each sub-region. We introduce two auxiliary variables u_x and u_y, and iterated solving following function:

$$\begin{cases} s_l = F^{-1}\left(\frac{\gamma\left(F(\partial_x)*F(u_x)+F(\partial_y)*F(u_y)\right)+F(f)*F(s_b)}{\gamma\left(|F(\partial_x)|^2+|F(\partial_y)|^2\right)+|F(f)|^2}\right) \\ u_x = \arg\min\limits_{u_x} \gamma|u_x| + (u_x - \partial_x s_l)^2 \\ u_y = \arg\min\limits_{u_y} \gamma|u_y| + (u_y - \partial_y s_l)^2 \end{cases} \tag{14}$$

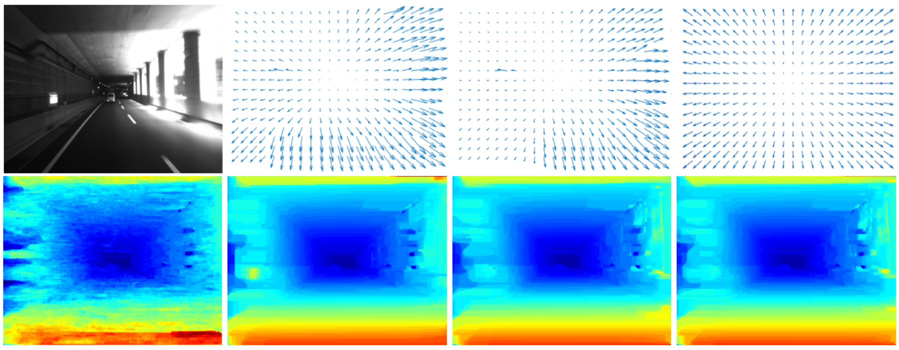
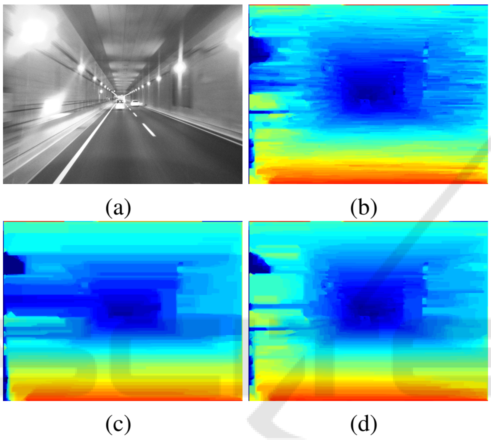


Figure 8: The first column is the original image and the disparity result of MPV (Long et al., 2014b). The second to last column are the PSF and disparity results of optical flow, cepstrum and the proposed kernel methods, respectively.



(a) (b)

(c) (d)

Figure 9: (a) is the blurry and noisy left image. (b) is the stereo result without noise reduction. (c) is the stereo result using the denoise algorithm as pre-processing method. (d) is the stereo result of the proposed method.

After all the sub-regions are restored, we obtain a weighted sum of these sub-regions with Hanning window function to produce the final results. Since in the proposed method the blur kernel is not totally correct in some area, the γ for the proposed method is 10 times bigger than the optical flow and cepstrum methods.

5.1 Noisy Case

In the low light condition such as a tunnel, the captured images are not only blurry but also contain high-level noise. Simply using the denoising algorithm as a pre-processing step will improve the quality the deblurred result, but significant artifacts can be observable along the deblurred image edges and/or the image structure may be over-smoothed (Tai and Lin, 2012).

Since in our simple case our blur kernel is not correct in some parts, we already use a large smoothing parameter to compensate the result. Using the denoising algorithm will further make the final result even more smooth. In this paper, we adopt the following strategy to handle this problem as shown in Alg. 1. First, we denoise the input image using fast non-local means algorithm (Goossens et al., 2010). With the denoising output $\tilde{b}$ we compute a deblurring result $\tilde{l}$. Then, we refine $\tilde{b}$ according to the motion blur constraints from $\tilde{l}$ avoid over-sharpening and/or over-smoothing in the denoised image. Last, we use previous non-blind algorithm to deblur the refined $\tilde{b}$ and get the deblurred image. Specifically, we compute the $\tilde{l}$ by minimizing following cost function:

$$E(\tilde{l}) = ||b - \frac{1}{Z}\sum_i (\omega_i * \tilde{l}) \otimes f_i||^2 + \lambda_1 ||\nabla l||^2 \quad (15)$$

After we get $\tilde{l}$, we refine the blurry image $\tilde{b}$ as:

$$E(\tilde{b}) = ||\tilde{b} - \frac{1}{Z}\sum_i (\omega_i * \tilde{l}) \otimes f_i||^2 + \lambda_2 ||\nabla \tilde{b}||^2 + \lambda_3 ||b - \tilde{b}||^2 \quad (16)$$

Since Equation 15 and 16 are both $l2$ norm function, we can directly compute the closed-form solution using FFT. In Fig. 9, we show the stereo results of this noise reduction method. In Fig. 9 (b), we can see that the stereo result without dealing the noise still contains noise. In Fig. 9 (c), the non-local means algorithm is adopted to be a pre-processing step and the stereo result is over-smoothed. In Fig. 9 (d), our result is relatively better than others with less noises and correct disparity results for blurry and noisy areas.

6 EXPERIMENT

We evaluated the proposed method on the experimental autonomous car with the stereo camera mounted

Algorithm 1: Framework for denoising and deblurring.

```
Input: The noisy and blurry image b
Output: The denoised and deblurred image l
Compute denoised image $\widetilde{b}$ by (Goossens et al., 2010)
Solve $\widetilde{l}$ by minimize Equation 15.
Refine $\widetilde{b}$ by minimize Equation 16.
Solve $l$ by minimize Equation 12.
```

Table 2: Result on the tunnel scene.

Stereo	Out-Noc(3px)	Avg-Noc
Proposed	12.21%	2.12px
SGM (Hirschmüller, 2008)	32.97%	4.77px
ELAS (Geiger et al., 2011)	20.10%	3.52px
MPV (Long et al., 2014b)	24.33%	4.19px

outside the car and the stereo camera is Bumblebee BBX3-13S2C-38. We construct a tunnel stereo vision dataset that contain 200 examples using the Velodyne HDL-32E 3D laser scanner (10 Hz, 32 laser beams, range: 100 m) to get precise distance and construct the ground truth data. We compare the quantitative values of the proposed algorithm and other real-time methods in Table 2 and the proposed algorithm outperforms other existing methods for the tunnel images. We show some of the qualitative results in Fig. 6.

After computing the disparity, we validate the proposed algorithm within a ADAS application. We compute the road surface by using the disparity results. The road surface is detected by using the histogram of disparity in horizontal and vertical direction. We denote the V-disparity (H_v) to be the row-based histogram of the different disparities in each row of the disparity map and the U-disparity (H_u) to be the column-based histogram. Then we use the V-disparity and U-disparity (Long et al., 2014a) Viterbi based optimization method to detect the road surface. The results are shown in Fig. 10. We can see the disparity results of SGM (Hirschmüller, 2008) contain much more holes and errors in the road surface, and generate a slightly more irregular road area and road boundary.

6.1 Extend to Non-tunnel Case

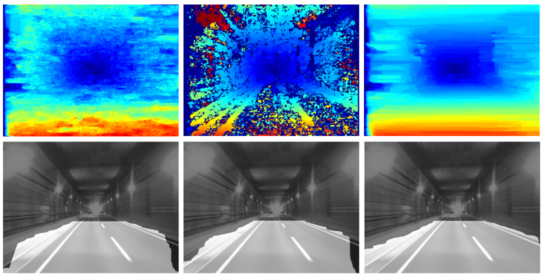


Figure 10: Road surface comparison. The first to last columns represent the disparity result and corresponding road surface estimation of method (Long et al., 2014b), (Hirschmüller, 2008) and the propose simple method.

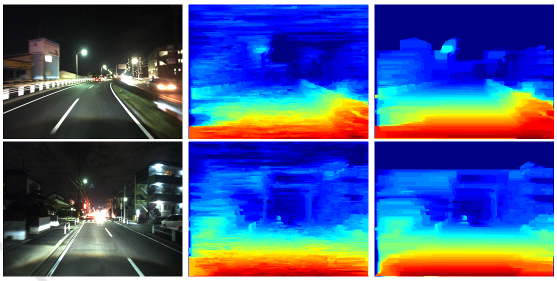


Figure 11: Results on the nighttime images. The first to last columns represent is the input left image of stereo pair, the stereo results of (Long et al., 2014b) and our results.

We evaluate the proposed algorithm on some non-tunnel images in low light conditions such as nighttime, the captured images will also be blurred and noisy. Although the background of the nighttime images are more complicated than the tunnel case and the blur kernel of the proposed method is not suitable in some parts of the image, the proposed algorithm still can significantly improve the stereo results as shown in Fig. 11.

7 CONCLUSION

In this paper, we have presented a novel algorithm to improve stereo vision in the tunnel. Typically, the motion kernel for tunnel cases is spatially-variant and difficult to be estimated in the real-time. Considering the characteristics of the tunnel case, we segment the image into overlapped subregions and assume the motion kernel in each region to be a linear function. Then we utilize three algorithms to estimate the PSF by optical flow, cepstrum and simple constant kernel. We compare the results of these three approaches and find that the proposed simple kernel is enough to produce promising disparity results with reduced computational efficiency. This simple method together with the fast non-blind deblurring algorithm and multi-scale MPV method are the final real-time deblurring-stereo framework for the tunnel case. The experiment results show the proposed algorithm can effectively and efficiently deal with the real tunnel images. Furthermore although the proposed algorithm is designed for tunnel images, it performs well on other images captured at low light conditions.

REFERENCES

Boykov, Y., Veksler, O., and Zabih, R. (2001). Fast approximate energy minimization via graph cuts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(11):1222–1239.

Cho, S. and Lee, S. (2009). Fast motion deblurring. In *ACM Transactions on Graphics (TOG)*, volume 28, page 145. ACM.

Fergus, R., Singh, B., Hertzmann, A., Roweis, S. T., and Freeman, W. T. (2006). Removing camera shake from a single photograph. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 787–794. ACM.

Geiger, A., Roser, M., and Urtasun, R. (2011). Efficient large-scale stereo matching. In *Computer Vision–ACCV 2010*, pages 25–38. Springer.

Goossens, B., Luong, H., Aelterman, J., Pižurica, A., and Philips, W. (2010). A gpu-accelerated real-time nlmeans algorithm for denoising color video sequences. In *International Conference on Advanced Concepts for Intelligent Vision Systems*, pages 46–57. Springer.

He, K., Sun, J., and Tang, X. (2013). Guided image filtering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(6):1397–1409.

Hirsch, M., Schuler, C. J., Harmeling, S., and Scholkopf, B. (2011). Fast removal of non-uniform camera shake. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 463–470. IEEE.

Hirschmüller, H. (2008). Stereo processing by semiglobal matching and mutual information. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(2):328–341.

Joshi, N., Kang, S. B., Zitnick, C. L., and Szeliski, R. (2010). Image deblurring using inertial measurement sensors. *ACM Transactions on Graphics (TOG)*, 29(4):30.

Krishnan, D. and Fergus, R. (2009). Fast image deconvolution using hyper-laplacian priors. In *Advances in Neural Information Processing Systems*, pages 1033–1041.

Long, Q., Xie, Q., Mita, S., Ishimaru, K., and Shirai, N. (2014a). A real-time dense stereo matching method for critical environment sensing in autonomous driving. In *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, pages 853–860. IEEE.

Long, Q., Xie, Q., Mita, S., Tehrani, H., Ishimaru, K., and Guo, C. (2014b). Real-time dense disparity estimation based on multi-path viterbi for intelligent vehicle applications. In *Proceedings of the British Machine Vision Conference*. BMVA Press.

Rom, R. (1975). On the cepstrum of two-dimensional functions (corresp.). *Information Theory, IEEE Transactions on*, 21(2):214–217.

Scharstein, D. and Szeliski, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47(1-3):7–42.

Son, T. T. and Seiichi, M. (2006). Stereo matching algorithm using a simplified trellis diagram iteratively and bi-directionally. *IEICE transactions on information and systems*, 89(1):314–325.

Sun, J., Zheng, N.-N., and Shum, H.-Y. (2003). Stereo matching using belief propagation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(7):787–800.

Tai, Y.-W. and Lin, S. (2012). Motion-aware noise filtering for deblurring of noisy and blurry images. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 17–24. IEEE.

Tai, Y.-W., Tan, P., and Brown, M. S. (2011). Richardson-lucy deblurring for scenes under a projective motion path. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(8):1603–1618.

Veksler, O. (2005). Stereo correspondence by dynamic programming on a tree. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 384–390. IEEE.

Wang, R. and Tao, D. (2014). Recent progress in image deblurring. *arXiv preprint arXiv:1409.6838*.

Whyte, O., Sivic, J., and Zisserman, A. (2011). Deblurring shaken and partially saturated images. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 745–752. IEEE.

Xu, L. and Jia, J. (2010). Two-phase kernel estimation for robust motion deblurring. In *Computer Vision–ECCV 2010*, pages 157–170. Springer.

Xu, Y., Hu, X., Wang, L., and Peng, S. (2012). Single image blind deblurring with image decomposition. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 929–932. IEEE.

Zabih, R. and Woodfill, J. (1994). Non-parametric local transforms for computing visual correspondence. In *Computer VisionłECCV'94*, pages 151–158. Springer.

Zbontar, J. and LeCun, Y. (2014). Computing the stereo matching cost with a convolutional neural network. *arXiv preprint arXiv:1409.4326*.

Zheng, S., Xu, L., and Jia, J. (2013). Forward motion deblurring. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1465–1472.