NUMERICALLY STABLE ESTIMATION OF SCENE FLOW INDEPENDENT OF BRIGHTNESS AND REGULARIZER WEIGHTS

Yusuke Kameda, Ichiro Matsuda and Susumu Itoh

Department of Electrical Engineering, Faculty of Science and Technology,
Tokyo University of Science
2641 Yamazaki, Noda-shi, Chiba 278-8510, JAPAN
ykameda@rs.tus.ac.jp

ABSTRACT

In video images, apparent motions can be computed using optical flow estimation. However, estimation of the depth directional velocity is difficult using only a single viewpoint. Scene flows (SF) are three-dimensional (3D) vector fields with apparent motion and a depth directional velocity field, which are computed from stereo video. The 3D motion of objects and a camera can be estimated using SF, thus it is used for obstacle detection and self-localization. SF estimation methods require the numerical computation of nonlinear equations to prevent over-smoothing due to the regularization of SF. Since the numerical stability depends on the image and regularizer weights, it is impossible to determine appropriate values for the weights. Thus, we propose a method that is independent of the images and weights, which simplifies previous methods and derives the numerical stability conditions, thereby facilitating the estimation of suitable weights. We also evaluated the performance of the proposed method.

Index Terms— Disparity, numerical stability, scene flow, stereo, variational method

1. INTRODUCTION

Research into motion estimation in video has flourished since the 1980s, particularly in the area of optical flow estimation [1, 2]. It is possible to estimate an object's apparent motion, but it is difficult to estimate the three-dimensional (3D) motion of objects using optical flow alone because it lacks depth directional information. Therefore, it is necessary to combine optical flow with a method that can restore depth when estimating 3D motion.

The methods used for restoring depth in images can be divided into two main categories. The first method uses range sensors based on laser and infrared pattern irradiation, and the second method uses stereo cameras. The first method has very high restoration accuracy in favorable conditions, such as still objects located in indoor environments, but it is not suitable for moving objects and the cameras required are more expensive than ordinary cameras. The stereo method estimates depth using the disparity and distance between cameras.

3D motion estimation methods that extend optical flow estimation methods can be categorized into two types. The first method is RGB-D flow estimation [3], which uses laser sensors to obtain color images. The second is scene flow estimation [4, 5], which combines optical flow estimation and stereo vision using a variational method developed in the 1990s [4]. Scene flow is a 3D vector field, which comprises optical flow and the time variation in the disparity field of stereo images [4, 5]. Scene flow can be converted geometrically into 3D motions on object surfaces using disparities and the distance between cameras. Scene flow is applicable to various fields, such as obstacle detection and action analysis. Furthermore, the estimation of the self-motion of a camera based on scene flow may potentially be applied to the self-localization of automated guided robots and vehicles. The application of scene flow to self-driving vehicles has been studied already [5, 6].

The theory of scene flow estimation is based on a variational method. Scene flow is determined from constraints based on the brightness invariance in the corresponding pixels between frames and stereo. However, unique solutions cannot be determined in the same brightness area based on the constraints alone. Thus, scene flow regularizers are employed. Regularizers based on Nagel method and total variation (TV) are used to prevent scene flow over-smoothing by controlling the strength of smoothing based on the norm of the brightness gradient and the flow vector of each pixel, respectively, where these regularizers are referred to as image- and flow-driven. It is difficult to obtain analytical solutions to the Euler-Lagrange (EL) equations derived from minimization problems that comprise these regularizers and constraints, thus numerical solutions are often obtained by iteration [7].

Two problems affect numerical computation using previous methods [5]. First, large nonlinear iterations are required with a TV regularizer. Thus, it is desirable to reduce the computational cost as much as possible while preventing over-smoothing. Second, the numerical stability has not been addressed sufficiently but it depends on the regularizer weights and brightness, where iteration does not converge in

all cases. Therefore, it is difficult to identify suitable regularizer weights. Resolving these problems will allow iterations to converge in a reliable manner and facilitate the determination of suitable weights. This solution will facilitate hardware implementations for scene flow estimation.

In this study, we reconsider these previous complex and nonlinear methods, and we simplify the algorithm. Using stability analysis, we propose a method where the numerical stability is independent of the regularizer weights and brightness. Furthermore, we show that the proposed algorithm can prevent over-smoothing, which is similar to image- and flow-driven regularizers. Moreover, our method facilitates the identification of suitable regularizer weights.

2. VARIATIONAL PROBLEM FOR SCENE FLOW

We assume that stereo cameras have the same specification and that they are rectified and synchronized. Thus, the vertical disparities of the stereo images are zero. Furthermore, we assume that the disparity field has already been estimated for each frame. Under these assumptions, we construct the energy minimization problem for scene flow estimation using the brightness consistency and a scene flow regularizer. The outline of the minimization problem framework is based on a previously reported method [5].

2.1. Constraints for Stereo Video

Let $L(\boldsymbol{x}, t)$ be the brightness value of the left image at pixel $\boldsymbol{x} = (x, y)^\top \in \Omega \subset \mathbb{R}^2$ and time t, where Ω is the image domain. If the brightness values of objects are invariant with a short time difference Δt,

$$L(\boldsymbol{x}, t) = L(\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x}, t)\Delta t, t + \Delta t) \tag{1}$$

is satisfied for any pixel using the apparent motion (flow vector) $\boldsymbol{u}(\boldsymbol{x}, t) = (u(\boldsymbol{x}, t), v(\boldsymbol{x}, t))^\top$. Equation (1) is called an optical flow constraint [1]. The concept described by (1) is illustrated in Figure 1 (a).

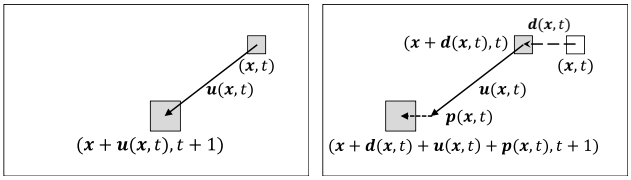
Similarly, for the brightness value $R(\boldsymbol{x}, t)$ of the right image, if there is a disparity $\boldsymbol{d}(\boldsymbol{x}, t) = (d(\boldsymbol{x}, t), 0)^\top$,

$$\begin{aligned} &R(\boldsymbol{x} + \boldsymbol{d}(\boldsymbol{x}, t), t) \\ =&R(\boldsymbol{x} + \boldsymbol{d}(\boldsymbol{x}, t) + (\boldsymbol{u}(\boldsymbol{x}, t) + \boldsymbol{p}(\boldsymbol{x}, t))\Delta t, t + \Delta t) \end{aligned} \tag{2}$$

is satisfied by using $\boldsymbol{u}(\boldsymbol{x}, t)$ and the disparity change $\boldsymbol{p}(\boldsymbol{x}, t) = (p(\boldsymbol{x}, t), 0)^\top$. The concept described by (2) is illustrated in Figure 1 (b). Furthermore, if the brightness values of the corresponding stereo pixels are the same at time $t + \Delta t$,

$$\begin{aligned} &L(\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x}, t)\Delta t, t + \Delta t) \\ =&R(\boldsymbol{x} + \boldsymbol{d}(\boldsymbol{x}, t) + (\boldsymbol{u}(\boldsymbol{x}, t) + \boldsymbol{p}(\boldsymbol{x}, t))\Delta t, t + \Delta t) \end{aligned} \tag{3}$$

is satisfied. In this study, we assume $\Delta t = 1$ for simplicity. Scene flow is the paired apparent motion (optical flow) $\boldsymbol{u}$ of the left image and the disparity change p. We denote $\boldsymbol{v} = (\boldsymbol{u}^\top, p)^\top = (u, v, p)^\top$.



(a) Motion of pixel in the left image (b) Motion of pixel in the right image

Fig. 1: Pixel correspondence of stereo. The disparity increases over time as the square moves downward and to the left.

2.2. Energy minimization using a variational method

We assume that the scene flow and image are second-order differentiable. Denoting $E_{LF}^2(\boldsymbol{v}, \boldsymbol{x}, t)$, $E_{RF}^2(\boldsymbol{v}, \boldsymbol{x}, t)$ and $E_{DF}^2(\boldsymbol{v}, \boldsymbol{x}, t)$ as the squared errors of both sides of (1) to (3), respectively, we define the data term as

$$E_D(\boldsymbol{v}, \boldsymbol{x}, t) = E_{LF}^2 + c(\boldsymbol{x}, t)\left(E_{RF}^2 + E_{DF}^2\right), \tag{4}$$

where we abbreviate $(\boldsymbol{v}, \boldsymbol{x}, t)$. The coefficient $c(\boldsymbol{x}, t)$ is 0 if $d(\boldsymbol{x}, t)$ is undefined because of occlusion and other factors, and 1 otherwise. $c(\boldsymbol{x}, t)$ controls whether it is necessary to apply the pixel correspondence to the right image.

Since (4) is ill-posed, an additional regularizer $E_S(\boldsymbol{v}, \boldsymbol{x}, t)$ of $\boldsymbol{v}$ is required. The energy functional is defined as

$$J(\boldsymbol{v}, t) = \int_\Omega \left(E_D(\boldsymbol{v}, \boldsymbol{x}, t) + E_S(\boldsymbol{v}, \boldsymbol{x}, t)\right) d\boldsymbol{x} \tag{5}$$

in [5]. In the present study, the regularizer is defined by

$$E_S(\boldsymbol{v}, \boldsymbol{x}, t) = \lambda\left(|\nabla u|^2 + |\nabla v|^2\right) + \gamma|\nabla p|^2, \tag{6}$$

where the positive real λ and γ are the regularizer weights and $\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right)^\top$. For u, v, and p, we abbreviate $(\boldsymbol{x}, t)$. Similarly, we abbreviate $L(\boldsymbol{x}, t)$ to L and $R(\boldsymbol{x} + \boldsymbol{d}, t)$ to R to avoid misunderstandings.

The solution to the minimization problem described by (5) is the solution of the EL equations

$$\lambda\nabla^2 u - L_x E_{LF} - cR_x E_{RF} = 0 \tag{7}$$

$$\lambda\nabla^2 v - L_y E_{LF} - cR_y E_{RF} = 0 \tag{8}$$

$$\gamma\nabla^2 p - cR_x(E_{DF} + E_{RF}) = 0 \tag{9}$$

for each $(\boldsymbol{x}, t)$ [5], where $\nabla^2 = \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right)$ and each subscript x and y of L and R denotes partial space differentiation.

2.3. Pyramid Transform and Image Warping

To compute the EL equations with coarse-to-fine, it is usually necessary to develop a complex scheme, which is constructed using a Taylor expansion and a fixed point method for the nonlinear term (4) [5, 8]. By contrast, we simplify the scheme by focusing on the pyramid transform and image warping.

In the algorithm, the pyramid transform generates low resolution stereo images, which decreases $|\boldsymbol{v}(\boldsymbol{x},t)|$ at the same rate as the resolution. Therefore, we assume that $|\boldsymbol{v}(\boldsymbol{x},t)|$ is sufficiently small. Using scene flows estimated at lower resolution, image warping approximates the frame at $t+1$ to the frame at t in an iterative manner. More specifically, it computes the scene flows between the left images $L(\boldsymbol{x},t)$ and the warped $L_{\mathrm{warp}} := L(\boldsymbol{x}+\boldsymbol{u}, t+1)$, and the right images $R(\boldsymbol{x}+\boldsymbol{d},t)$ and the warped $R_{\mathrm{warp}} := R(\boldsymbol{x}+\boldsymbol{d}+\boldsymbol{u}+\boldsymbol{p}, t+1)$ in the condition with a zero vector field. It is possible to increase the estimation accuracy by adding each scene flow estimated between the warped images. After iterating image warping at a lower resolution, image warping is computed at a higher resolution in a similar manner by upsampling the lower resolution scene flow. Thus, the scene flow is estimated at the original resolution. The concept of this algorithm is formalized in Algorithm 1.

Algorithm 1: Scene flow estimation using multi-resolution image warping.

Data: L, R, d, λ, γ, maximum pyramid level $l_{\max}$, warp and successive over-relaxation (SOR) iterations $w_{\max}$, $k_{\max}$, and $m_{\max}$
Result: scene flow $\boldsymbol{v}$

```
for l = 0 to l_max do
    build Gaussian pyramids L^[l], R^[l], and d^[l];
for l := l_max to 0 do
    for w := 1 to w_max do
        Using d^[l] and v^[l], compute L_warp and R_warp;
        v_tmp := {0}∀x;
        for k := 1, m := 1 to k_max, m_max do
            iterate v_tmp using L^[l], L_warp, R^[l], R_warp;
        v^[l] := v^[l] + v_tmp
    generate v^[l-1] by upsampling v^[l];
v := v^[0];
```

2.4. EL equations

For small values of $\boldsymbol{v}$, we construct an energy functional using the first-order approximation of the Taylor expansion of (4),

$$\begin{aligned} E_D \simeq \left(\boldsymbol{u}^\top \nabla L + L_t\right)^2 + c\left(\boldsymbol{u}^\top \nabla R + R_x p + R_t\right)^2 \\ + c\left(\boldsymbol{u}^\top \nabla D + R_x p + D_t\right)^2, \end{aligned} \tag{10}$$

where L_t and R_t are time derivatives approximated by frame subtractions, D_t is the subtraction of stereo images at frame $t+1$, and $D = R - L$. Using the symmetric matrix

$$\boldsymbol{S}_2 = \boldsymbol{S}_0(L) + c\left(\boldsymbol{S}_0(R) + \boldsymbol{S}_0(D)\right), \tag{11}$$

where $\boldsymbol{S}_0(f) = \nabla f (\nabla f)^\top$ is the local structure tensor of f and positive semi-definite, we define

$$\boldsymbol{S} = \left(\begin{array}{c|c} \boldsymbol{S}_2 & \boldsymbol{s} \\ \hline \boldsymbol{s}^\top & 2cR_x^2 \end{array}\right), \text{ where } \boldsymbol{s} = 2cR_x\nabla(R+D), \tag{12}$$

and the constant terms

$$\boldsymbol{b}_1 = L_t \nabla L + c\left(R_t \nabla R + D_t \nabla D\right), \quad b_2 = cR_x(R_t + D_t). \tag{13}$$

Using (12) and (13), the EL equations are simplified as

$$\nabla^2 \begin{pmatrix} \lambda \boldsymbol{u} \\ \gamma p \end{pmatrix} - \boldsymbol{S} \begin{pmatrix} \boldsymbol{u} \\ p \end{pmatrix} - \begin{pmatrix} \boldsymbol{b}_1 \\ b_2 \end{pmatrix} = \boldsymbol{0}. \tag{14}$$

3. ITERATIVE COMPUTATION AND ITS STABILITY

Next, we derive the numerical stability of the iteration of (14). We describe the computation of $\boldsymbol{v}_{\mathrm{tmp}}$ for the innermost iteration in Algorithm 1 and abbreviate $\boldsymbol{v}_{\mathrm{tmp}}$ to $\boldsymbol{v}$. Using a semi-implicit scheme [9] and successive over-relaxation method (SOR), we define the recurrence formula of (14) for $\boldsymbol{u}$ as

$$\boldsymbol{u}^{(k+1)} = \boldsymbol{u}^{(k)} + \frac{\omega_1}{4}\tilde{\nabla}^2 \boldsymbol{u}^{(k)} - \frac{\omega_1}{4\lambda}\left(\boldsymbol{S}_2 \boldsymbol{u}^{(k+1)} + p\boldsymbol{s} + \boldsymbol{b}_1\right), \tag{15}$$

where the superscript (k) denotes the iteration count of $\boldsymbol{u}$, $\omega_1 \in (0,2)$ is an acceleration factor, and $\tilde{\nabla}^2$ is the approximation of the Laplacian using second-order central differences [1]. While iterating $\boldsymbol{u}$, let p be constant. Simplifying (15),

$$\boldsymbol{u}^{(k+1)} = \boldsymbol{A}\left(G_1 \boldsymbol{u}^{(k)} - \frac{\omega_1}{4\lambda}\left(p\boldsymbol{s} + \boldsymbol{b}_1\right)\right) \tag{16}$$

$$\boldsymbol{A} = \left(\boldsymbol{I}_2 + \frac{\omega_1}{4\lambda}\boldsymbol{S}_2\right)^{-1}, \quad G_1 = 1 + \frac{\omega_1}{4}\tilde{\nabla}^2, \tag{17}$$

where $\boldsymbol{I}_2$ is the identity matrix of size 2. Since (11) is positive semi-definite, the eigenvalue of $\boldsymbol{A}$ does not exceed 1 without depending on $\omega_1 > 0$ and $\lambda > 0$. Therefore, the stability condition of (16) is that the spectral radius of the coefficient matrix that comprises G_1 does not exceed 1. Therefore, the stability condition of $\boldsymbol{u}$ is

$$\left|1 + \frac{\omega_1}{4}(-4-4)\right| = |1 - 2\omega_1| \leq 1, \tag{18}$$

thus $\boldsymbol{u}$ is stable if $\omega_1 \leq 1$.

Similarly, we define the recurrence formula of p as

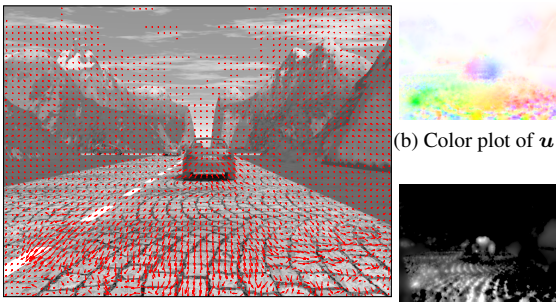
$$p^{(m+1)} = G_2 p^{(m)} - \frac{\omega_2}{4\gamma}\left(\boldsymbol{s}^\top \boldsymbol{u} + 2cR_x^2 p^{(m+1)} + b_2\right), \tag{19}$$

where, as with $\boldsymbol{u}$, $G_2 = 1 + \frac{\omega_2}{4}\tilde{\nabla}^2$, (m) is the iteration count for p, and $\omega_2 \in (0,2)$ is an acceleration factor for the SOR of p. While iterating p, let $\boldsymbol{u}$ be constant. By simplification,

$$p^{(m+1)} = \frac{1}{1 + \frac{c\omega_2 R_x^2}{2\gamma}}\left(G_2 p^{(m)} - \frac{\omega_2}{4\gamma}\left(\boldsymbol{s}^\top \boldsymbol{u} + b_2\right)\right). \tag{20}$$

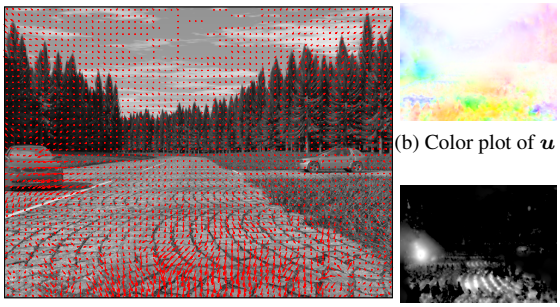
In the same manner as $\boldsymbol{u}$, p is stable if $\omega_2 \leq 1$.

[1] The number of diagonal elements in the coefficient matrix is 4, thus we divide the right-hand side of (15) by 4, excluding the first term.



(a) Optical flow $\boldsymbol{u}$ (b) Color plot of $\boldsymbol{u}$ (c) $p > 0$

Fig. 2: Scene flow of the 11th frame of sequence 1. The camera moves forward and the leading vehicle recedes.



(a) Optical flow $\boldsymbol{u}$ (b) Color plot of $\boldsymbol{u}$ (c) $p > 0$

Fig. 3: Scene flow in the 220nd frame of sequence 2. The left oncoming vehicle is turning right.

Fig. 4: Color illustration of the direction and speed of $\boldsymbol{u}(\boldsymbol{x}, t)$.

Using the semi-implicit scheme, the proposed method can prevent over-smoothing by multiplying the diffusion term by the inverse matrix of the coefficient matrix that comprises $\boldsymbol{S}_2$. Therefore, the term that includes $\boldsymbol{S}_2$ has the property of an image-driven regularizer.

The partial space derivative at each pixel is average of that for the previous and the next frames. The first-order partial derivative is approximated by a Scharr operator [10]. The frame and stereo subtractions are computed from images, which are smoothed by a Gaussian function with a standard deviation of 0.8. To accelerate the iteration process, we use parallel computation with the over-relaxed red-black Gauss-Seidel method [11] for $\boldsymbol{u}^{(k)}$ and $p^{(m)}$.

4. EXPERIMENTS AND DISCUSSION

In these experiments, the computational conditions were the same as the values reported in [5] to facilitate comparisons, i.e., $\lambda = 0.06$, $\gamma = 0.6$, $k_{\max} = m_{\max} = 3$, $w_{\max} = 2$, and $l_{\max} = 4$. The initial and boundary conditions for $\boldsymbol{v}$ were a zero vector field and homogeneous Neumann conditions, respectively. The images used were EISATS set 2, sequences 1 and 2 [12]. The computational environment comprised a system with an Intel Core i7-3770 3.4 GHz processor, Visual C++ 2013, and OpenCV 2.4.8. The frame rate of the proposed method is 5 frames per second (fps) for 640×480 pixles. It is almost 4 times faster than [5], in which the frame rate is 5 fps for 320×240 pixels.

Figures 2 and 3 show the results obtained using the proposed method. We used the color representation [2] shown in Figure 4 for the dense drawing of $\boldsymbol{u}$. The regions where $p > 0$ correspond to the approaching objects. In the region with the leading vehicle in Figure 2, the scene flow was not estimated accurately because of the transparency of the rear window. For these images, λ and γ, the iteration diverged if $\omega_1 \geq 2$ or $\omega_2 \geq 2$. This was because the spectral radius $\rho(\boldsymbol{A})$ declined to less than 1, depending on the images and weights, and (18) was relaxed. As the result, the iteration was stable if $\omega_1 \leq 1$ and $\omega_2 \leq 1$, in theory, and it was possible to accelerate the process using the SOR method.

We evaluated the errors between the estimated $\boldsymbol{u}(\boldsymbol{x}, t)$ and ground truth $\boldsymbol{u}^*(\boldsymbol{x}, t)$ at each pixel. The errors were the mean endpoint error (EE) [2] and

$$\frac{1}{|\Omega|} \sum_{\boldsymbol{x} \in \Omega} |\boldsymbol{u} - \boldsymbol{u}^*| \tag{21}$$

and the mean spatiotemporal angle error (SAE) [2]

$$\frac{1}{|\Omega|} \sum_{\boldsymbol{x} \in \Omega} \arccos \left(\frac{\boldsymbol{u}^\top \boldsymbol{u}^* + 1}{\sqrt{|\boldsymbol{u}|^2 + 1} \sqrt{|\boldsymbol{u}^*|^2 + 1}} \right). \tag{22}$$

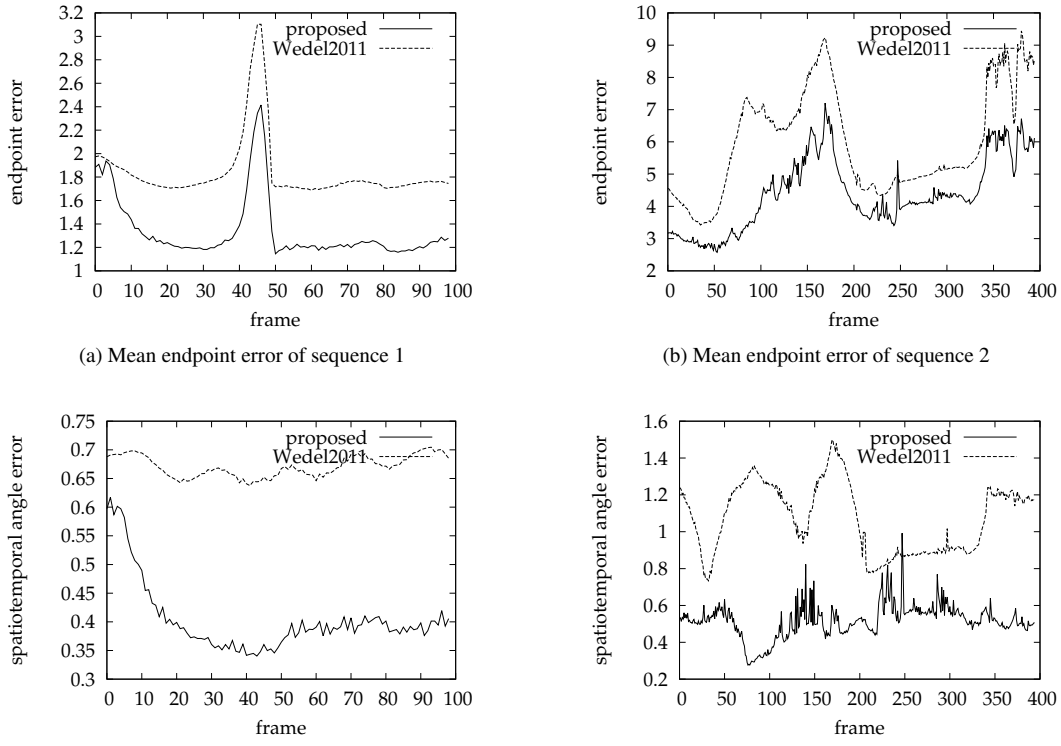
In these cases, the pixels where $\boldsymbol{u}^*(\boldsymbol{x}, t)$ did not exist were excluded. Figure 5 shows the evaluations for the proposed and previous methods [5]. The computational conditions used for the previous method were the same as the described above, i.e., the optimized values reported in [5], except the 'inner iteration' (which does not exist in our method) was 15. However, the iteration of the previous method diverged with the condition $\omega = 1.99$, thus we set $\omega = 1$. Compared with the previous method, the proposed method had better accuracy in terms of the average EE and SAE. Therefore, the proposed method has numerical stability and greater accuracy.

5. CONCLUSION

In this study, we analyzed the numerical stability of scene flow and verified our proposed method. The proposed method facilitates the stable computation of any images and regularizer weights, and the more accurate estimation of scene flow. This approach allows the determination of suitable weights.

REFERENCES

[1] B.K.P. Horn and B.G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.



(a) Mean endpoint error of sequence 1

(b) Mean endpoint error of sequence 2

(c) Mean spatiotemporal angle error of sequence 1

(d) Mean spatiotemporal angle error of sequence 2

Fig. 5: Evaluations of $\boldsymbol{u}$ in the estimated scene flows and ground truths for each frame in EISATS set 2, sequences 1 and 2.

[2] S. Baker, D. Scharstein, J.P. Lewis, S. Roth, M.J. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," *International Journal of Computer Vision*, vol. 92, no. 1, pp. 1–31, 2011.

[3] E. Herbst and D. Fox, "RGB-D flow : dense 3-D motion estimation using color and depth," in *IEEE International Conference on Robotics and Automation*, 2013, pp. 2276–2282.

[4] I. Patras, E.A. Hendriks, and G. Tziritas, "A joint motion / disparity estimation method for the construction of stereo interpolated images in stereoscopic image sequences," in *International Conference on Pattern Recognition*, 1996, pp. 359–368.

[5] A. Wedel, T. Brox, T. Vaudrey, C. Rabe, U. Franke, and D. Cremers, "Stereoscopic scene flow computation for 3D motion understanding," *International Journal of Computer Vision*, vol. 95, no. 1, pp. 29–51, 2011.

[6] S. Sivaraman and M.M. Trivedi, "Looking at vehicles on the road: a survey of vision-based vehicle detection, tracking, and behavior analysis," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1773 – 1795, 2013.

[7] J. Weickert and C. Schnörr, "Variational optic flow computation with a spatio-temporal smoothness constraint," *Journal of Mathematical Imaging and Vision*, vol. 14, no. 3, pp. 245–255, 2001.

[8] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," in *European Conference on Computer Vision*, *Lecture Notes in Computer Science* vol. 3024, pp. 25–36, 2004.

[9] Y. Kameda, A. Imiya, and N. Ohnishi, "A convergence proof for the Horn-Schunck optical-flow computation scheme using neighborhood decomposition," in *12th International Workshop on Combinatorial Image Analysis*, *Lecture Notes in Computer Science* vol. 4958, pp. 262–273, 2008.

[10] H. Scharr, *Optimal Operators in Digital Image Processing*, Doctral thesis, Rupertus Carola University of Heidelberg, 2000.

[11] Y. Saad, *Iterative Methods for Sparse Linear Systems: Second Edition*, Society for Industrial and Applied Mathematics, 2003.

[12] T. Vaudrey, C. Rabe, R. Klette, and J. Milburn, "Differences between stereo and motion behaviour on synthetic and real-world stereo sequences," in *23rd International Conference Image and Vision Computing New Zealand*, pp. 1–6, 2008.