Numerically stable multi-channel depth scene flow with adaptive weighting of regularization terms

Yusuke Kameda
Dept. Electrical Engineering,
Faculty of Engineering,
Tokyo University of Science
6-3-1 Niijuku, katsushika-ku,
Tokyo, Japan
https://orcid.org/0000-0001-8503-4098

Ichiro Matsuda
Dept. Electrical Engineering,
Faculty of Science and Technology,
Tokyo University of Science
2641 Yamazaki, Noda-shi,
Chiba, Japan
matsuda@ee.noda.tus.ac.jp

Susumu Itoh
Dept. Electrical Engineering,
Faculty of Science and Technology,
Tokyo University of Science
2641 Yamazaki, Noda-shi,
Chiba, Japan
s.itoh@rs.tus.ac.jp

***Abstract*—Scene flow is a three-dimensional (3D) vector field with velocity in the depth direction and optical flow that represents the apparent motion, which can be estimated from RGB-D videos. Scene flow can be used to estimate the 3D motion of objects with a camera; thus, it is used for obstacle detection and self-localization. It can potentially be applied to inter prediction in 3D video coding. The scene-flow estimation method based on the variational method requires numerical computations of nonlinear equations that control the regularization strength to prevent excessive smoothing due to scene-flow regularization. Because numerical stability depends on multi-channel images and computational parameters such as regularization weights, it is difficult to determine appropriate parameters that satisfy the stability requirements. Therefore, we propose a numerical computation method to derive a numerical stability condition that does not depend on the color of the image or the weight of the regularization term. This simplifies the traditional method and facilitates the setting up of various regularization weight functions. Finally, we evaluate the performance of the proposed method.**

***Index Terms*—numerical stability, scene flow, RGB-D, variational method, multi-channel**

I. Introduction

Research on video motion estimation has been actively conducted since the 1980s, specifically in the field of optical-flow estimation [1]. Although it is possible to estimate the apparent motion of objects, it is difficult to estimate the three-dimensional (3D) motion of objects because optical flow alone does not contain depth information. Therefore, optical flow must be combined with a method that can derive the depth information when estimating 3D motion. The methods used to restore the image depth can be divided into two main categories. The first uses a stereo camera, and the second uses a range sensor based on laser and infrared-pattern illuminations. The former method utilizes the disparity and distance between two cameras for depth estimation. Although this is a relatively inexpensive method, it is not effective in estimating parallaxes of dark places or distant objects. The latter method can in fact be used in dark places. Various devices, such as an in-vehicle sensor (in the case of distant objects) and a finger sensor (in the case of close objects) have been developed to implement this method. In the 3D video coding method, to generate a video from an arbitrary viewpoint, data is handled in a form that combines images and depth maps. 3D video data consists of multiple viewpoints and their depth maps [2].

3D motion-estimation methods that extend the optical flow-estimation method can be divided into two types. In the first method, stereo images and those disparities are used to estimate the stereo-based scene flow [3]. In the second method, images taken with laser sensors and depth maps are used to estimate the depth-based scene flow [4]. These are combined with optical-flow estimation using a variational method developed in the 1990s [3]. Scene flow is a 3D vector field that consists of optical flow representing the apparent velocity and the velocity in the depth direction. Scene flow is the 3D motion on the object surface. Scene flow can be applied to various fields such as obstacle detection and action analysis. Furthermore, the estimation of the camera self-motion based on scene flow can potentially be applied to the self-localization of auto-guided robots and vehicles. The application of scene flow to autonomous vehicles has been studied [5]. Further, the application to inter prediction in 3D video coding is being studied [6], and applications to various fields are expected.

The theory of scene-flow estimation is based on variational methods. Scene flow is determined from data terms representing constraints based on color invariance of the same object between frames. However, it is not possible to determine a unique solution for areas of the same color based solely on data terms. Therefore, scene-flow regularization is used. Some regularizers based on the Nagel type method [7] and nonlinear total variation (TV) [8] control the smoothing intensity based on the luminance gradient of the image and the norm of the flow vector, respectively, to prevent excessive scene-flow smoothing. These are called image-driven and flow-driven regularizers, respectively [8]. These techniques are based on the idea that regularization terms must be assigned less weight at the motion boundaries. Some methods based on Deep Convolutional Neural Network (DCNN) have also been proposed [9]. Because DCNN is suitable for object recognition and classification, an improvement in the estimation accuracy,

This work was supported by JSPS KAKENHI Grant Numbers 16K00250, JP17K12717, and 20K19829.

specifically at object boundaries, has been claimed. However, even with a lightweight DCNN, it is difficult to implement embedded devices such as the IoT; thus, traditional signal processing that can perform high-speed operations has some advantages. A hybrid method that takes advantage of the ideas of both the variational method and the DCNN has also been proposed [10]. Because it is difficult to obtain an analytical solution to the variational Euler-Lagrange (EL) equation from the minimization problem, numerical solutions are often obtained by performing iterations [8].

The numerical solution may converge to a value different from the solution of the proposed mathematical model, unless the problem of numerical stability depending on the input image and parameters is solved. In addition, different convergence values may be obtained depending on the computer environment, which means that the performance in the situation where numerical stability cannot be guaranteed may not be evaluated accurately. A heuristic search for input images and computational parameters that provides numerical stability typically has considerably high computational complexity. It is difficult to specify the stability condition numerically. Therefore, it is necessary to derive stability conditions based on the theory of numerical analysis. Conventionally, numerical stability has been discussed in optical-flow estimation of multi-channel color images in block units instead of regularization terms [11]. [12] has conducted a stability analysis of an iterative method based on a semi-implicit solution for optical-flow estimation form grayscale images. Numerical stability analysis has been performed on a stereo scene flow of a monochrome image [13]. However, in a scene-flow estimation from multi-channel images and depth maps, the problem of numerical stability caused by setting the weight to an arbitrary value and a function of position has not been solved, and the Courant-Friedrichs-Lewy (CFL) condition has not been proved.

The contribution of this study is that it proves the CFL condition that is independent of the values of images and depth maps, the weight between channels, and the weight of the regularization term for the scene-flow estimation from weighted multi-channel images and depth maps. Naturally, the weight indicating the importance of the channels can be included in the conversion coefficient of the color space. However, there may be an application wherein the inter-channel weight is dynamically changed as a function of position, time, or the like. Moreover, in terms of the clarity of meaning of this weight in the cost function, weighting data terms is better than using one's own color space. In addition, because the weight of the regularization term can be arbitrarily set, a research field of how to set the weight function of the regularization term can be established. Unlike the DCNN and non-linear flow-driven methods, the proposed method separates the design of the regularization weight function from the mathematical model and computation method. This makes motion estimation possible using an arbitrary weight function; it is also expected that the accuracy of motion estimation will be improved by an adaptive weight function at the object boundaries. In addition, it is not necessary to search for hyperparameters for stable computation, which leads to various industrial developments such as embedded hardware and real-time computation in application fields such as automatic driving and stereoscopic video coding.

II. Mathematical model of depth scene flow

Let the multi-channel information at position $\boldsymbol{x} = (x, y)^\top \in \Omega \subset \mathbb{R}^2$ on the image at time t be $\boldsymbol{I}(\boldsymbol{x}, t) = (I_1(\boldsymbol{x}, t), I_2(\boldsymbol{x}, t), I_3(\boldsymbol{x}, t))^\top \in \mathbb{R}^3$, and $\boldsymbol{I}$ is a vector-valued function that can be first-order differentiated with respect to t and $\boldsymbol{x}$. Subsequently, three channels will be discussed in the same manner as general multi-color images, but the number of channels can be set arbitrarily. Assuming the invariance of object color with respect to the small time change Δt, the following equation holds.

$$\boldsymbol{I}(\boldsymbol{x}, t) = \boldsymbol{I}(\boldsymbol{x} - \boldsymbol{U}(\boldsymbol{x}, t)\Delta t, t - \Delta t) \tag{1}$$

Here,

$$\boldsymbol{U}(\boldsymbol{x}, t) = (u(\boldsymbol{x}, t), v(\boldsymbol{x}, t))^\top = (dx/dt(\boldsymbol{x}, t), dy/dt(\boldsymbol{x}, t))^\top \tag{2}$$

is the apparent speed on the image, called optical flow [1]. In this equation, $\boldsymbol{U}$, when sampled in pixels, is called a backward flow and represents a correspondence between the pixel and a position in a previous frame.

Assume that the depth (distance) map $Z(\boldsymbol{x}, t)$ is measured from the image plane to objects on the image by an RGB-D camera. The temporal change of depth map is expressed by the following equation.

$$Z(\boldsymbol{x}, t) - Z(\boldsymbol{x} - \boldsymbol{U}(\boldsymbol{x}, t)\Delta t, t - \Delta t) = w(\boldsymbol{x}, t)\Delta t \tag{3}$$

Here, $w(\boldsymbol{x}, t)$ is the velocity in the direction perpendicular to the image plane, and scene flow $\boldsymbol{V} = (u, v, w)^\top$ represents the three-dimensional velocity of objects. Similar to $\boldsymbol{U}$, in this equation, w, when sampled in pixels, is called a backward flow and represents the difference between the depth of the pixel and the corresponding position in the previous frame.

The data term for calculating scene flow is defined as follows [4]. Hereafter, for simplicity, let $\Delta t = 1$.

$$\begin{aligned} J_{\boldsymbol{I}}(\boldsymbol{x}, t, \boldsymbol{U}) =& \|\mathrm{diag}(\boldsymbol{c})(\boldsymbol{I}(\boldsymbol{x}, t) - \boldsymbol{I}(\boldsymbol{x} - \boldsymbol{U}(\boldsymbol{x}, t), t - 1))\|_2^2 \\ =& c_1^2(I_1(\boldsymbol{x}, t) - I_1(\boldsymbol{x} - \boldsymbol{U}(\boldsymbol{x}, t), t - 1))^2 + \\ & c_2^2(I_2(\boldsymbol{x}, t) - I_2(\boldsymbol{x} - \boldsymbol{U}(\boldsymbol{x}, t), t - 1))^2 + \\ & c_3^2(I_3(\boldsymbol{x}, t) - I_3(\boldsymbol{x} - \boldsymbol{U}(\boldsymbol{x}, t), t - 1))^2 \end{aligned} \tag{4}$$

$$J_Z(\boldsymbol{x}, t, \boldsymbol{U}, w) = (Z(\boldsymbol{x}, t) - Z(\boldsymbol{x} - \boldsymbol{U}(\boldsymbol{x}, t), t - 1) - w(\boldsymbol{x}, t))^2 \tag{5}$$

Here, the constant vector $\boldsymbol{c} = (c_1, c_2, c_3)^\top \in \mathbb{R}^3$ is the weight between colors, and $\mathrm{diag}(\boldsymbol{c})$ is a diagonal matrix, the components of $\boldsymbol{c}$ being the diagonal elements.

Because these conditional expressions involve an ill-posed problem in a flat region depending on the image derivative, a regularization term

$$\begin{aligned} & J_R^2(\boldsymbol{x}, t, \boldsymbol{U}, w) \\ =& \|\nabla u(\boldsymbol{x}, t)\|_2^2 + \|\nabla v(\boldsymbol{x}, t)\|_2^2 + \beta(\boldsymbol{x}, t)\|\nabla w(\boldsymbol{x}, t)\|_2^2 \end{aligned} \tag{6}$$

is defined [4]. Here, $\nabla = (\partial_x, \partial_y)^\top$ is the Nabla operator. From these equations, the data terms are defined.

$$J_D(\boldsymbol{x}, t, \boldsymbol{U}, w) = J_{\boldsymbol{I}}(\boldsymbol{x}, t, \boldsymbol{U}) + \mu(\boldsymbol{x}, t) J_Z(\boldsymbol{x}, t, \boldsymbol{U}, w) \quad (7)$$

From the regularization term $J_R(\boldsymbol{x}, t, \boldsymbol{U}, w)$ for u, v, w, the energy functional J of the following equation is defined.

$$\iint_\Omega \{J_D(\boldsymbol{x}, t, \boldsymbol{U}, w) + \lambda(\boldsymbol{x}, t) J_R(\boldsymbol{x}, t, \boldsymbol{U}, w)\} \, d\boldsymbol{x} \quad (8)$$

where $\mu(\boldsymbol{x}, t)$ and $\lambda(\boldsymbol{x}, t)$ are the non-negative weighting factors for each term. Scene flow as a function of u, v, w that minimizes this functional is determined. The difference from the regularization term of the previous research is that the weight of the regularization term λ is a function, and it is necessary to guarantee the numerical stability for any value of lambda.

III. PDE derivation by the variational method

By the image pyramid transformation, the apparent motion of the half size image is half that of the original image. Therefore, at resolutions where the apparent motion u, v is small enough, the above data term is given by the following equations (by Taylor expansion).

$$\begin{aligned}
J_{\boldsymbol{I}}(\boldsymbol{x}, t, \boldsymbol{U}) &= \left\| \mathrm{diag}(\boldsymbol{c}) \left((\nabla(\boldsymbol{I})^\top)^\top \boldsymbol{U} + \partial_t \boldsymbol{I} \right) \right\|_2^2 \\
&= c_1^2 (u \partial_x I_1 + v \partial_y I_1 + \partial_t I_1)^2 + \\
&\quad c_2^2 (u \partial_x I_2 + v \partial_y I_2 + \partial_t I_2)^2 + \\
&\quad c_3^2 (u \partial_x I_3 + v \partial_y I_3 + \partial_t I_3)^2 \\
&= \sum_i c_i^2 \left(\frac{dI_i}{dt} \right)^2 \qquad (9)
\end{aligned}$$

$$\begin{aligned}
J_Z(\boldsymbol{x}, t, \boldsymbol{U}, w) &= (u \partial_x Z + v \partial_y Z + \partial_t Z - w)^2 \\
&= \left(\frac{dZ}{dt} - w \right)^2 \qquad (10)
\end{aligned}$$

Here, ∂_t represents the time partial derivative. The solution of the cost-functional-minimization problem is the solution of the following Euler-Lagrange system of partial differential equations. [1]

$$\lambda \nabla^2 u - \sum_i c_i^2 \frac{dI_i}{dt} \partial_x I_i - \mu \left(\frac{dZ}{dt} - w \right) \partial_x Z = 0 \quad (11)$$

$$\lambda \nabla^2 v - \sum_i c_i^2 \frac{dI_i}{dt} \partial_y I_i - \mu \left(\frac{dZ}{dt} - w \right) \partial_y Z = 0 \quad (12)$$

$$\lambda \beta \nabla^2 w + \mu \left(\frac{dZ}{dt} - w \right) = 0 \quad (13)$$

Here, $\nabla^2 = \nabla^\top \nabla$ is a Laplacian, and $(\boldsymbol{x}, t)$ is omitted from the functions $u, v, w, \boldsymbol{I}, Z, \mu, \beta$ for simplicity.

The following expression is obtained by considering $\boldsymbol{V} = (u, v, w)^\top$ and summarizing it.

$$\mathrm{diag}(\lambda, \lambda, \lambda\beta) \nabla^2 \boldsymbol{V} - \boldsymbol{S}\boldsymbol{V} = \boldsymbol{b} \quad (14)$$

Here, the constant term

$$\begin{aligned}
\boldsymbol{b} &= \begin{pmatrix} \sum_i c_i^2 \partial_t I_i \partial_x I_i + \mu \partial_t Z \partial_x Z \\ \sum_i c_i^2 \partial_t I_i \partial_y I_i + \mu \partial_t Z \partial_y Z \\ -\mu \partial_t Z \end{pmatrix} \\
&= \begin{pmatrix} \sum_i c_i^2 \partial_t I_i \nabla I_i + \mu \partial_t Z \nabla Z \\ -\mu \partial_t Z \end{pmatrix}. \qquad (15)
\end{aligned}$$

The following equation is obtained using the local structure tensor $\boldsymbol{S}_0(I_i) = \nabla I_i (\nabla I_i)^\top$ of the image.

$$\boldsymbol{S} = \begin{pmatrix} \sum_i c_i^2 \boldsymbol{S}_0(I_i) + \mu \boldsymbol{S}_0(Z) & \mu \nabla Z \\ \mu (\nabla Z)^\top & \mu \end{pmatrix} \quad (16)$$

The symmetric matrix $\boldsymbol{S}$ can be decomposed as follows.

$$\boldsymbol{S} = \sum_i c_i^2 \boldsymbol{S}_{I_i} + \mu \boldsymbol{S}_Z, \quad (17)$$

$$\boldsymbol{S}_{I_i} = \begin{pmatrix} \boldsymbol{S}_0(I_i) & \boldsymbol{0} \\ \boldsymbol{0}^\top & 0 \end{pmatrix}, \quad (18)$$

$$\boldsymbol{S}_Z = \begin{pmatrix} \boldsymbol{S}_0(Z) & \nabla Z \\ (\nabla Z)^\top & 1 \end{pmatrix}. \quad (19)$$

IV. Numerical analysis

In general, the repetition matrix when the expression (14) is discretized does not always have a positive definite value. Therefore, we define the following recurrence formula based on the acceleration factor and the semi-implicit method [13].

$$\begin{aligned}
\boldsymbol{V}^{(k+1)} &= \boldsymbol{V}^{(k)} + \frac{\omega}{4} \nabla^2 \boldsymbol{V}^{(k)} \\
&\quad - \frac{\omega}{4} \mathrm{diag}^{-1}(\lambda, \lambda, \lambda\beta) \left(\boldsymbol{S}\boldsymbol{V}^{(k+1)} + \boldsymbol{b} \right) \qquad (20)
\end{aligned}$$

Here, $\omega \in (0, 2)$ is the acceleration factor.

By rearranging the equations, the following equations are obtained.

$$\begin{aligned}
&\left(\boldsymbol{E}_3 + \frac{\omega}{4} \mathrm{diag}^{-1}(\lambda, \lambda, \lambda\beta) \boldsymbol{S} \right) \boldsymbol{V}^{(k+1)} \\
&= \left(\boldsymbol{E}_3 + \frac{\omega}{4} \nabla^2 \right) \boldsymbol{V}^{(k)} - \frac{\omega}{4} \mathrm{diag}^{-1}(\lambda, \lambda, \lambda\beta) \boldsymbol{b} \qquad (21)
\end{aligned}$$

Here $\boldsymbol{E}_3$ is a 3×3 identity matrix. Transforming into the form of an iterative matrix yields

$$\boldsymbol{V}^{(k+1)} = \boldsymbol{A}^{-1} \left(\boldsymbol{G}\boldsymbol{V}^{(k)} - \frac{\omega}{4} \mathrm{diag}^{-1}(\lambda, \lambda, \lambda\beta) \boldsymbol{b} \right), \quad (22)$$

$$\boldsymbol{A} = \left(\boldsymbol{E}_3 + \frac{\omega}{4} \mathrm{diag}^{-1}(\lambda, \lambda, \lambda\beta) \boldsymbol{S} \right), \quad (23)$$

$$\boldsymbol{G} = \left(\boldsymbol{E}_3 + \frac{\omega}{4} \nabla^2 \right). \quad (24)$$

The numerical stability of this recurrence formula is guaranteed if the spectral radius $\rho(\boldsymbol{A}^{-1}\boldsymbol{G}) \leq 1$, which represents the maximum absolute value of the eigenvalues of the iterative matrix $\boldsymbol{A^{-1}G}$.

[1]

$$\begin{aligned}
\frac{\partial}{\partial u} J - \frac{\partial}{\partial x} \frac{\partial}{\partial(\partial_x u)} J - \frac{\partial}{\partial y} \frac{\partial}{\partial(\partial_y u)} J = 0 \\
\frac{\partial}{\partial v} J - \frac{\partial}{\partial x} \frac{\partial}{\partial(\partial_x v)} J - \frac{\partial}{\partial y} \frac{\partial}{\partial(\partial_y v)} J = 0 \\
\frac{\partial}{\partial w} J - \frac{\partial}{\partial x} \frac{\partial}{\partial(\partial_x w)} J - \frac{\partial}{\partial y} \frac{\partial}{\partial(\partial_y w)} J = 0
\end{aligned}$$

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

```
Data: $\boldsymbol{I}(\boldsymbol{x}, t-1)$, $\boldsymbol{I}(\boldsymbol{x}, t)$, $\boldsymbol{c}$, $Z(\boldsymbol{x}, t-1)$, $Z(\boldsymbol{x}, t)$, $\lambda$,
      $\beta$, $\mu$, initial backward scene flow $\boldsymbol{V}(\boldsymbol{x}, t)$,
      maximum pyramid level $l_{\max}$, iterations $k_{\max}$
Result: estimated backward scene flow $\boldsymbol{V}(\boldsymbol{x}, t)$
for $l = 0$ to $l_{\max}$ do
    build Gaussian pyramids $\boldsymbol{I}[l](\boldsymbol{x}, t-1)$, $\boldsymbol{I}[l](\boldsymbol{x}, t)$,
      $Z[l](\boldsymbol{x}, t-1)$ and $Z[l](\boldsymbol{x}, t)$;
    build Gaussian pyramid initial flow $\boldsymbol{V}[l](\boldsymbol{x}, t)$;
for $l := l_{\max}$ to $0$ do
    Using $\boldsymbol{U}[l](\boldsymbol{x}, t)$, compute $\boldsymbol{I}_{\text{warp}}[l](\boldsymbol{x}, t-1)$ and
      $Z_{\text{warp}}[l](\boldsymbol{x}, t-1)$ by warping;
    $\boldsymbol{V}_{\text{tmp}} := \{\boldsymbol{0}\}\forall\boldsymbol{x}$;
    for $k := 1$ to $k_{\max}$ do
        iterate $\boldsymbol{V}_{\text{tmp}}$ using $\boldsymbol{I}[l](\boldsymbol{x}, t)$, $\boldsymbol{I}_{\text{warp}}[l](\boldsymbol{x}, t-1)$,
          $Z(\boldsymbol{x}, t)$, $Z_{\text{warp}}[l](\boldsymbol{x}, t-1)$;
    $\boldsymbol{V}[l] := \boldsymbol{V}[l] + \boldsymbol{V}_{\text{tmp}}$;
    generate $\boldsymbol{V}[l-1]$ by upsampling $\boldsymbol{V}[l]$;
$\boldsymbol{V} := \boldsymbol{V}[0]$;
```

Consider $\boldsymbol{S}$ that constitutes $\boldsymbol{A}$. The eigenvalues of $\boldsymbol{S}_{I_i}$ are $\|\nabla I_i\|^2$ and 0; thus, this is a semipositive definite symmetric matrix. The eigenvalues of $\boldsymbol{S}_Z$ are $\|\nabla Z\|^2 + 1$ and 0; thus, this is also a semi-definite symmetric matrix. From $c_i^2 > 0$ and $\mu > 0$, $\boldsymbol{S}$ is a positive definite symmetric matrix [14]. Therefore, from $\omega > 0, \lambda > 0, \beta > 0$, $\boldsymbol{A}$ is a positive definite symmetric matrix with an eigenvalue 1 or more. Therefore, the eigenvalue of $\boldsymbol{A}^{-1}$ lies in (0,1], independent of the pixel value $\boldsymbol{I}, Z$ of images and depth maps and the parameters $\lambda, \beta, \mu, \omega$. Therefore, $\rho(\boldsymbol{G}) \leq 1$ is a sufficient condition for numerical stability, and only $\boldsymbol{G}$ needs to be analyzed.

Then, obtaining the second-order central-difference approximation of the Laplacian of $\boldsymbol{G}$ is similar to solving a certain type of two-dimensional reaction-diffusion equation by the explicit method of the forward time centered space difference method. Therefore, the stability condition $\rho(\boldsymbol{G}) \leq 1$ is the same as the stability condition of the explicit method of the two-dimensional diffusion equation; the Laplacian coefficient is less than $\frac{1}{4}$. From the above, the sufficient condition for $\rho(\boldsymbol{A}^{-1}\boldsymbol{G}) \leq 1$ is $\omega \leq 1$. Therefore, it can be inferred that the stability condition does not depend on parameters such as input-image and cost-functional weights.

The whole algorithm including motion compensation (warping) and multi-resolution estimation with Gaussian pyramid [15], [16] is summarized as in the conventional method [13] as shown in the Algorithm 1.

V. Stability demonstration and performance

To demonstrate the above theory, Fig.1 presents the average values of the entire image of the iteration error $\|\boldsymbol{V}^{(k+1)} - \boldsymbol{V}^{(k)}\|_1$ with respect to the number of iterations. Here, the input video is the first and second frames of alley_2 of MPI Sintel Flow Dataset [17], presented in Fig. 3.

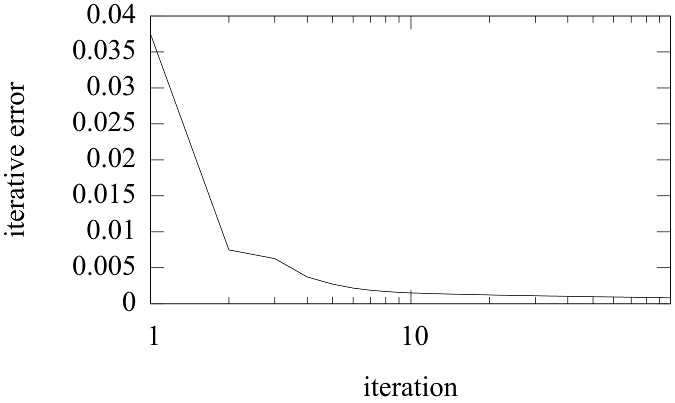


Fig. 1: Average iterative error

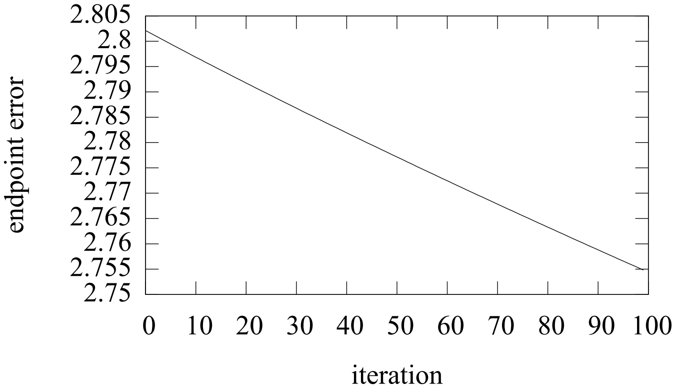
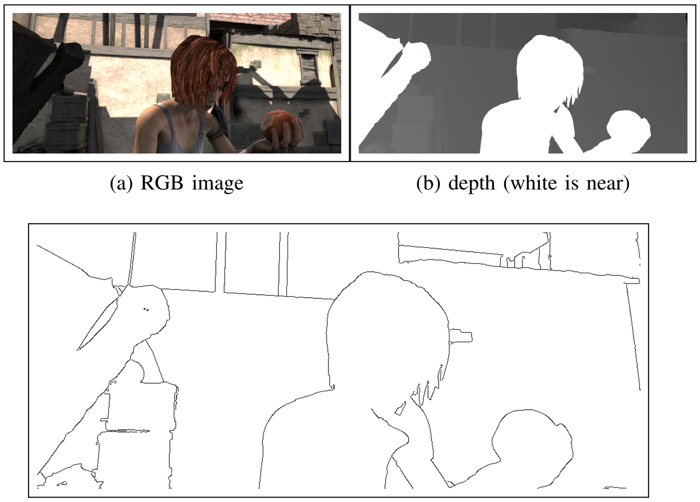


Fig. 2: Average endpoint error [pixel / frame]

In this experiment, the RGB-D channel values were normalized to the range $[0, 1]$, and then, the color space was converted from RGB to YUV. To accelerate the iteration process, we used parallel computation with the over-relaxed red-black Gauss-Seidel method [18]. Without parameter tuning, in particular, we set $c_i = 1, \mu(\boldsymbol{x}) = 1\forall\boldsymbol{x}, \lambda = 1, \omega = 1, \beta = 1$. It can be inferred from the figure that the numerical solutions of pixels do not diverge; they converge to a specific value even if the number of iterations increases. Figure 2 presents the results of a performance evaluation performed using average endpoint error (the L2 norm of the error vector) with the true value of the optical flow part $\boldsymbol{U}$ under the same conditions. The figure indicates that the accuracy improves with each iteration. The reason why the average endpoint error gradually decreases with respect to the number of iterations is that the numerical-solution update is propagated to the texture-less region by the regularization term.

This study makes it possible to define lambda functions, and it presents a small but simple example. Because the weight of the regularization term must be reduced at the boundary of the motion, the weight function is defined by detecting the edge of the depth map. Because the value of the depth map of this frame is [0, 40], the two thresholds of the Canny edge detector



(a) RGB image

(b) depth (white is near)

(c) weight map by Canny edge detector

Fig. 3: The 1st frame of RGB-D video 'alley_2'

[19] were set to 0 and 5, respectively, and the weight of the edge portion was set to 1/128 of the other area. The weight map is shown in Fig.3c. The numerical solution converged even when the weight function was changed for each pixel. In addition, the average endpoint error was reduced from 2.76 to 2.67 pixel / frame by using the pixel adaptive weight in this manner.

VI. Conclusion

In this study, we proposed a multi-channel depth scene flow estimation method whose numerical stability is independent of multi-channel images, depth maps, and computation parameters and demonstrates the stability conditions theoretically. We also experimentally demonstrated that the numerical solutions converged. Furthermore, in the regularization term of the scene flow, it was demonstrated that the pixel adaptive regularization weight can be set independently of the computation method, and the effect was confirmed. However, the pixel-adaptation weights at this stage correspond only to isotropic regularization. To independently set the pixel adaptation weight of the anisotropic regularization, we intend to design the regularization term in the future.

References

[1] B. K. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, no. 1-3, pp. 185–203, aug 1981. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/0004370281900242

[2] G. Tech, Y. Chen, K. Muller, J.-R. Ohm, A. Vetro, and Y.-K. Wang, "Overview of the Multiview and 3D Extensions of High Efficiency Video Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 1, pp. 35–49, jan 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7258339/

[3] 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.

[4] E. Herbst, X. Ren, and D. Fox, "RGB-D flow: Dense 3-D motion estimation using color and depth," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, may 2013, pp. 2276–2282. [Online]. Available: http://ieeexplore.ieee.org/document/6630885/

[5] 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, dec 2013. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6563169

[6] S. KASAI, Y. KAMEDA, T. ISHIKAWA, I. MATSUDA, and S. ITOH, "Pixel-Wise Interframe Prediction based on Dense Three-Dimensional Motion Estimation for Depth Map Coding," *IEICE Transactions on Information and Systems*, vol. E100.D, no. 9, pp. 2039–2043, 2017. [Online]. Available: https://www.jstage.jst.go.jp/article/transinf/E100.D/9/E100.D_2016PCL0007/_article

[7] H.-H. Nagel and W. Enkelmann, "An Investigation of Smoothness Constraints for the Estimation of Displacement Vector Fields from Image Sequences," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 5, pp. 565–593, sep 1986. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4767833

[8] 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. [Online]. Available: link.springer.com/article/10.1023/A:1011286029287

[9] L. Shao, P. Shah, V. Dwaracherla, and J. Bohg, "Motion-Based Object Segmentation Based on Dense RGB-D Scene Flow," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3797–3804, oct 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8411477/

[10] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, vol. D. IEEE, jun 2018, pp. 8934–8943. [Online]. Available: http://arxiv.org/abs/1709.02371 https://ieeexplore.ieee.org/document/8579029/

[11] K. R. T. Aires, A. M. Santana, and A. A. D. Medeiros, "Optical flow using color information," in *Proceedings of the 2008 ACM symposium on Applied computing - SAC '08*. New York, New York, USA: ACM Press, 2008, p. 1607. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1363686.1364064

[12] L. Le Tarnec, F. Destrempes, G. Cloutier, and D. Garcia, "A Proof of Convergence of the Horn–Schunck Optical Flow Algorithm in Arbitrary Dimension," *SIAM Journal on Imaging Sciences*, vol. 7, no. 1, pp. 277–293, jan 2014. [Online]. Available: http://epubs.siam.org/doi/10.1137/130904727

[13] Y. Kameda, I. Matsuda, and S. Itoh, "Numerically stable estimation of scene flow independent of brightness and regularizer weights," in *22nd European Signal Processing Conference, {EUSIPCO} 2014, Lisbon, Portugal, September 1-5, 2014*, 2014, pp. 1068—-1072. [Online]. Available: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=6952373

[14] R. S. Varga, *Matrix Iterative Analysis*, ser. Springer Series in Computational Mathematics. Springer Berlin Heidelberg, 2009. [Online]. Available: https://books.google.co.jp/books?id=URlGAAAAQBAJ

[15] E. Simoncelli, E. Adelson, and D. Heeger, "Probability distributions of optical flow," in *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE Comput. Sco. Press, 1991, pp. 310–315. [Online]. Available: http://ieeexplore.ieee.org/document/139707/

[16] P. J. Burt, "Fast filter transform for image processing," *Computer Graphics and Image Processing*, vol. 16, no. 1, pp. 20–51, may 1981. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/0146664X81900927

[17] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A Naturalistic Open Source Movie for Optical Flow Evaluation," in *European Conf. on Computer Vision (ECCV)*, ser. Part IV, LNCS 7577, 2012, pp. 611–625. [Online]. Available: http://link.springer.com/10.1007/978-3-642-33783-3_44

[18] Y. Saad, *Iterative Methods for Sparse Linear Systems: Second Edition*. Society for Industrial and Applied Mathematics, 2003. [Online]. Available: http://books.google.co.jp/books?id=h9nwszYPblEC

[19] J. Canny, "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, nov 1986. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4767851