

Monocular scene flow estimation via variational method

Degui Xiao¹ · Qiuwei Yang¹ · Bing Yang² · Wei Wei³

Received: 1 August 2015 / Revised: 21 October 2015 / Accepted: 17 November 2015 /

Published online: 2 December 2015

© Springer Science+Business Media New York 2015

Abstract Scene flow provides the 3D motion field of point clouds, which correspond to image pixels. Current algorithms usually need complex stereo calibration before estimating flow, which has strong restrictions on the position of the camera. This paper proposes a monocular camera scene flow estimation algorithm. Firstly, an energy functional is constructed, where three important assumptions are turned into data terms derivation: a brightness constancy assumption, a gradient constancy assumption, and a short time object velocity constancy assumption. Two smooth operators are used as regularization terms. Then, an occluded map computation algorithm is used to ensure estimating scene flow only on un-occluded points. After that, the energy functional is solved with a coarse-to-fine variational equation on Gaussian pyramid, which can prevent the iteration from converging to a local minimum value. The experiment results show that the algorithm can use three sequential frames at least to get scene flow in world coordinate, without optical flow or disparity inputting.

Keywords Scene flow · Monocular camera · Time-space consistency · Energy functional · Nonlinear iteration · Intelligent healthy drivings

✉ Bing Yang
yangbing@126.com

Degui Xiao
dgxiao@hnu.edu.cn

Qiuwei Yang
yangqiuwei@hnu.edu.cn

Wei Wei
Weiwei8329@gmail.com

¹ College of Computer Science and Electronic Engineering, Hunan University, Changsha, China

² School of Education, Hubei University, Wuhan, China

³ School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China

1 Introduction

Recent revolutionary development of multimedia technologies has advanced many disciplines and industries, such as health, intelligent vehicles and augmented reality. In health area, with the help of crowd-based and social networking services, healthcare knowledge is more convenient to share, acquire and disseminate among health seekers and providers. To bridge vocabulary gap between health seekers and community generated knowledge, Nie et al. [22] presented a scheme to label question answer pairs by jointly utilizing local mining and global learning approaches. Health today is complementarily characterized by multi-modal data, which enables doctors to concisely comprehend the health conditions of the patients. To help understand chronic diseases progressions based on observational health records in form of multimedia data, Nie et al. [24] proposed an adaptive multimodal multi-task learning model to co-regularize the modality agreement, temporal progression and discriminative capabilities of different modalities. To make health knowledge exchange and reusability, Nie et al. [23] presented a multilingual system to return one multi-faceted answer that was well-structured and precisely extracted from multiple heterogeneous healthcare sources. Patients nowadays actively seek for online health information, and post their disease control experiences. While the vocabulary gap between health seekers and providers has hindered the cross-system operability and the inter-user reusability. Nie et al. [25] presented a novel scheme to code the medical records by jointly utilizing local mining and global learning approaches, which were tightly linked and mutually reinforced. Nie et al. [26] proposed a scheme accurately and efficiently inferring diseases especially for community-based health services. Mobiles and other wearable health sensors are equipped by patients and doctors to track the health and exercises, which makes it possible for real-time monitoring and remote health support. Camera is one kind of such sensors. Yan et al. proposed a novel Multi-Task Learning framework (FEGA-MTL) for classifying the head pose of a person moving freely in an environment monitored by multiple, large field-of-view surveillance cameras [31] and for action recognition [32]. For complex event detection in videos, Yan et al. [33] proposed a novel strategy to automatically select semantic meaningful concepts for the event detection task based on both the events-kit text descriptions and the concepts high-level feature descriptions. To cope with vast amount of unlabeled and heterogeneous data for recognizing human activities from videos, Yan et al. [34] proposed a multitask clustering framework for activity of daily living analysis from visual data gathered from wearable cameras. We think remote scene dynamics may be helpful for doctors to monitor patients and provide instructions for their health, so in this paper, we propose a monocular scene flow estimation method. The concept of scene flow comes from optical flow, it not only solves motion information in 3D camera frustum, but also overcomes the rigid motion assumption in optical flow, which makes it possible to get 6DOF (Dimension of Free) data in scene only form image sequences. Because motion information is based on scene structure itself, scene reconstruction is one of key problems when solving scene flow, beyond that, occlusion among different objects also should be taken into account for multi objects may have different moving state.

The prototype of scene flow comes from Gilad Adiv's study in 1985 [1], which put forward a method to calculate the depth and motion information of camera scene by utilizing binocular optical flow and rigid motion segments, and took a direct search way to match these segments for complexity of the subject and limited hardware. This paper innovatively extends scene flow estimation algorithm to image sequences taken from a single moving camera, and makes no assumption about rigidity of motion itself. At the same time, we express the consistency

assumption as a total energy functional by combining 3D scene structure and scene flow into a monocular camera projection model, besides, we make a smooth regularization in flow estimation, and anisotropy boundary operator is taken as smooth operator, which makes the result more close to nature. When solving the functional, according to Brox's optical flow method [6], we rewrite the main function according to Euler-Lagrange condition and use a coarse-to-fine framework to prevent the total equation converging to a local minimum after getting the iterative equation, so the nonlinearity of functional can be maintained until inner iteration. As the experimental results show and Brox proved, solving the total scene flow energy functional by PDE is effective and reasonable.

This paper is organized by 6 sections: Section 2 describes the state-of-art scene flow relative works from three research fields. Section 3 derives the total energy functional in detail, including inverse depth introduction and three important consistency assumptions. In order to get the numerical solution, we process energy functional according to Euler-Lagrange equation to get a non-linear iterative equation, and linearize it using a coarse-to-fine framework in section 4. Experiment results are shown in section 5. The last section summarizes the innovations and advantages of our algorithm, also points out some shortages under bad environmental conditions, which need to be fixed in our future work.

2 Related works

2.1 Stereo based scene flow

After solving optical flow problems with accurate and fast ways, study about binocular optical flow based scene flow methods [27, 28] were proposed. These algorithms always need a prior depth [14, 30] to get the scene structure, then refer the projection relationship between 3D scene flow and 2D image flow to solve motion parameters with least square method. Vogel et al. presented the dynamic 3D scene by collecting planar rigidly moving local segments [29]. Basha et al. proposed a 3D point cloud parameterization, which allows directly estimating the desired unknowns, their functional enforced multi-view geometric consistency and imposed brightness constancy and piecewise smoothness assumption directly on the 3D unknowns. Except constancy assumption, Birkbeck et al. [5, 9] took known proxy motion into account which enables 3D trajectory reconstruction when only a single view is available. Damn et al. tackled the 6D pose and additional shape degrees of freedom for the object of known class in the scene, combining image data and depth information for the pose and shape recovery. Even though these methods can get an effective solution, they are not general because of their requirement to much prior information.

2.2 RGBD scene flow

Since Microsoft released somatosensory equipment device Kinect [8, 12, 16, 18, 36], studies on RGBD dataset have become popular. Because RGBD data provides complete and reliable information, scene flow estimation based on RGBD data becomes important. Letouzey et al. [20] reconstructed 3D scene on RGBD images, combining geometric information from depth maps with intensity variations in color

images to estimate smooth and dense 3D motion fields, which takes advantage of the geometric information provided by the depth camera to define a surface domain. Herbst et al. [13] proposed a method which generalized two-frame variational 2D flow algorithms to 3D and computed flow reliably using RGBD data, overcoming depth noise. But similar methods work only under indoor situation, for the IR camera which takes depth image is easily influenced by sunshine. Even Yang et al. [35] proposed novel density modulated binary patterns for depth acquisition, the carried phase is not strictly sinusoidal and so the depth reconstructed from the phase contains a systematic error.

2.3 SLAM based scene flow

As the most important and basic algorithm in robotics [10], with decades of development, SLAM has become an effective algorithm which can accurately position camera only by image information. After getting camera position, cloud points matching can be used to rebuild the scene. Such as, Alcantarilla et al. [2] combined visual slam and dense scene flow to parse surrounding environment. The key idea is to continuously estimate a semi-dense inverse depth map for the current frame, which can be used to track the motion of the camera in turn. Even though SLAM offers much information to scene reconstruction, it needs a lot of posteriori data because it is based on probabilistic theory [15] and its initial evaluation is not particularly accurate, so the probability based SLAM algorithm does not suitable for dense scene flow estimation.

3 Monocular scene flow

We use a integrated energy functional to estimate scene flow, focus on getting inverse depth and scene flow of referenced frame, only from monocular image sequences. We make no assumption about motion rigidity of camera, and express the solution with world coordinate.

3.1 Pinhole model

As the pinhole model of camera in Fig. 1, we can get the relationship between 3D space object point and 2D image point:

$$\underbrace{\begin{bmatrix} 1/dx & 0 & o_u \\ 0 & 1/dv & o_v \\ 0 & 0 & 1 \end{bmatrix}}_k \underbrace{\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_r \underbrace{\begin{bmatrix} R_{C^i} & C_{C^i} \\ 0^t & 1 \end{bmatrix}}_{T_{C^i}} \underbrace{\begin{bmatrix} x_{w^i} \\ y_{w^i} \\ z_{w^i} \\ 1 \end{bmatrix}}_{\bar{x}_{w^i}} \tag{1}$$

$$= z_{C^i} \underbrace{\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix}}_{\bar{x}_{\mu^i}}$$

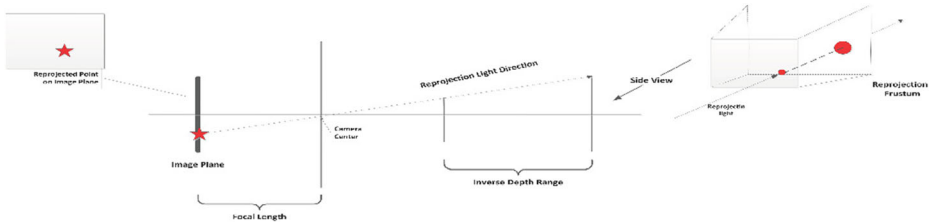


Fig. 1 Inverse depth: the red point is a pixel point on image plane, after back projection, its correspond 3D point will exist on back projected ray; the hexahedron contains the ray is camera frustum, instead, perspective projection can map the 3D point in frustum to the red pixel on image plane

Specially, let $M^i = \rho T_{C^i}$, which stands the 3x4 projection matrix of camera at time i , including external parameters relative to referenced frame and internal parameters of the camera itself.

3.2 Inverse depth

The conception of inverse depth came from Civer’s study about mono SLAM [7] and Richard emphasized it in his DTAM system [21]. According to pinhole model, inverse depth of a defined pixel will be in a range $d \subset \eta \left(\vec{X}_i \right)$ because of projection information loss (as Fig. 1).

For a 3D space object point $\vec{X}_{W^i} = (x_{W^i}, y_{W^i}, z_{W^i})$, the corresponding pixel point is $\vec{X}_{I^i} = (u_i, v_i, 1)$, their relationship can be expressed as:

$$d(u_i, v_i) \vec{X}_{I^i} = M^i \vec{X}_{W^i} \tag{2}$$

Obviously, $d(u_i, v_i)$, same as z_{C^i} in Eq. 1, is a function with image pixel position as its parameters. Thus, we can get a 3D point $\vec{X}_{W^i} = (x_{W^i}, y_{W^i}, d(u_i, v_i))$ by back projecting from a pixel on referenced frame, and then the 3D point will be re-projected to the time i image as a pixel of $X_{I^i} = f_{proj}(\vec{X}_{W^i})$. As we know, point position at next time in 3D space, is defined by current time position and velocity per time unit, so we can give the relationship between two succession time positions of a 3D point as: $\vec{X}_{W^i} = f_{pos}(\vec{X}_{W^0}, \vec{V}^i, t)$. The initial position of the 3D point comes from back projection (suppose the camera is located in the origin of the world coordinate in the first frame):

$$\begin{aligned} \vec{X}_{W^0}(I_0) &= f_{backproject} \left(I_0, d(u_0, v_0) \begin{bmatrix} x_{W^0} \\ y_{W^0} \\ z_{W^0} \end{bmatrix} \right) \\ &= d(u_0, v_0) \begin{bmatrix} u_0/fu - o_u/fu \\ v_0/fv - o_v/fv \\ 1 \end{bmatrix} \end{aligned} \tag{3}$$

Expanding above equation for a pixel, we express the corresponding relationship of a 3D space point cloud and a 2D image pixel as following:

$$\begin{aligned}
 u_i &= \frac{[M^i]_1 \overrightarrow{X_{W^i}}}{[M^i]_3 \overrightarrow{X_{W^i}}} = \frac{[M^i]_1 (\overrightarrow{X_{W^0}} + \overrightarrow{V_i})}{[M^i]_3 (\overrightarrow{X_{W^0}} + \overrightarrow{V_i})} \\
 &= \frac{M^i_{11}(x_{w^0} + V_x) + M^i_{12}(y_{w^0} + V_y)}{M^i_{31}(x_{w^0} + V_x) + M^i_{32}(y_{w^0} + V_y)} + \frac{M^i_{13}(d(u_0, v_0) + V_z) + M^i_{14}}{M^i_{33}(d(u_0, v_0) + V_z) + M^i_{34}} \\
 v_i &= \frac{[M^i]_2 \overrightarrow{X_{W^i}}}{[M^i]_3 \overrightarrow{X_{W^i}}} = \frac{M^i_2 (\overrightarrow{X_{W^0}} + \overrightarrow{V_i})}{M^i_3 (\overrightarrow{X_{W^0}} + \overrightarrow{V_i})} \\
 &= \frac{M^i_{21}(x_{w^0} + V_x) + M^i_{22}(y_{w^0} + V_y)}{M^i_{31}(x_{w^0} + V_x) + M^i_{32}(y_{w^0} + V_y)} + \frac{M^i_{23}(d(u_0, v_0) + V_z) + M^i_{24}}{M^i_{33}(d(u_0, v_0) + V_z) + M^i_{34}}
 \end{aligned} \tag{4}$$

Equation 4 shows that, after back projected to 3D space and moving, a pixel on referenced frame at time 0 can be re-projected as another pixel on time i frame.

3.3 Consistency assumption

This paper makes some reasonable assumptions to scene and projection information like optical flow method. We first propose short time velocity consistency based on spatial coincidence, and then we propose brightness consistency based on illumination invariant in short time constraint. In order to make our algorithm tolerant to the texture variances and brightness noise, we adopt gradient consistency.

3.3.1 Velocity consistency

As in Fig. 2, for sequential frames, we assume that object points' velocity in world space is constant within a short time interval, so we can formulate the relationship between moving camera and dynamic scene in short period successive frames. Let camera moving information as known quantity, the inverse depth and the object velocity $\overrightarrow{V_o}$ as unknown ones, the same 3D object point can be projected to different image positions for the moving of camera and objects. When getting the camera transform matrix and intrinsic matrix, pinhole model in Eq. 1 can be rewritten as a 3D space point moving relationship as in Eq. 5.

$$\begin{aligned}
 X_{W^i} &= f_{pos}(\overrightarrow{X_{W^0}}, \overrightarrow{V}, t_i) \\
 &= \overrightarrow{X_{W^0}} + t_i \overrightarrow{V} f_{backproject}(I_0, d(u_0, v_0) + t_i \overrightarrow{V}(u_0, v_0))
 \end{aligned} \tag{5}$$

3D space scene flow $\overrightarrow{V}(u_0, v_0)$ is a function of image coordinate, t_i is time interval between referenced time 0 frame to time i , so 3D point location of time i can be seen as a result of non-linear function of velocity and time.

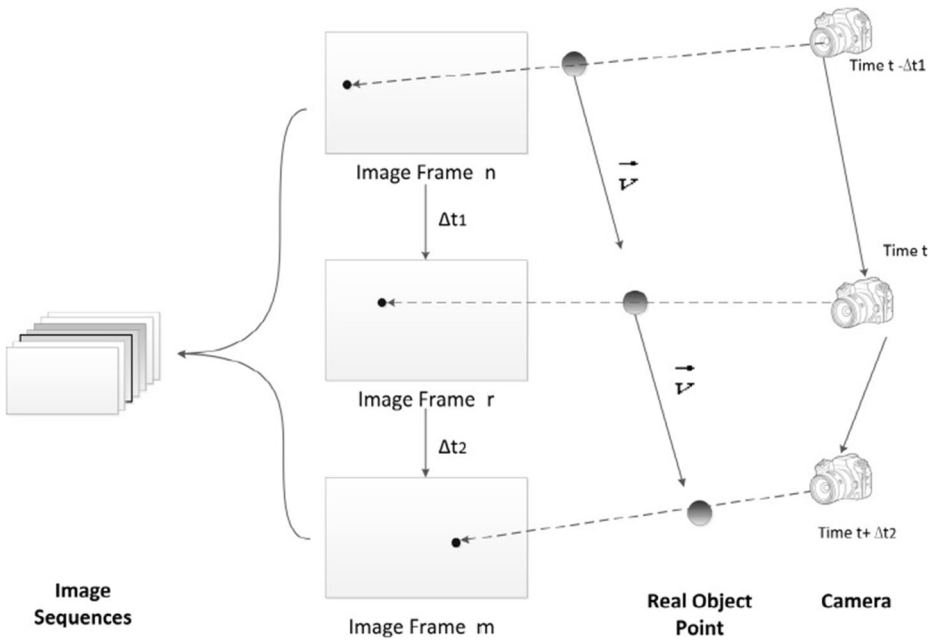


Fig. 2 Projection model with time changing, 3 frames from a successive monocular image sequences are extracted

3.3.2 Brightness consistency

Assuming there exists no mirror reflection in the scene and the environment structure or illumination condition is stable in a very short period of time, after a tiny movement, the image pixels from the same 3D point will have similar light intensity. Based on this assumption, we can conclude the illumination consistency constraint between real frame and its re-projected frame as in Eq. 6.

$$BC_{s1}(d, \vec{V}) = \sum_{i=0}^n m_{cs1}^i \int_{\Omega} \psi \left(|I_i - I_i(\vec{X}_{I_i})|^2 \right) \tag{6}$$

where I_i is the real frame at time i , and $I_i(\vec{X}_{I_i})$ is the re-projected frame based on evaluated inverse depth and scene flow. If the velocity consistency established in n frames, BC_{s1} is the brightness consistency in current frame. As in Eq. 6, integral range represents a rectangular image domain, which means our functional covers the whole image range by calculating Eqs. 3 and 4, with $\psi(x) = \sqrt{(x^2 + \epsilon)}$, ϵ is a tiny value, the function is used to ensure the convexity of the total functional, in other words, the functional has a global minimum, so we can use Euler-Lagrange equation to derivate iterative equation. m_{cs1}^i is a mask to cover occlusion points and prevent the consistency error caused by camera moving.

Referring to brightness consistency, we can also list an equation between referenced frame and re-projected frames as following:

$$BC_{s2}(d, \vec{V}) = \sum_{i=1}^n m_{cs2}^i \int_{\Omega} \psi \left(\left| I_0 - I_i(\vec{X}_{I_i}) \right|^2 \right) \tag{7}$$

3.3.3 Gradient consistency

In real environment, scene flow estimation merely with brightness consistency may be influenced by inevitably noises, so the gradient consistency is introduced in this paper to make algorithm much more robust.

$$GC(d, \vec{V}) = \sum_{i=1}^n m_{gc}^i \int_{\Omega} \psi \left(\left| \nabla I_i - \nabla I_i(\vec{X}_{I_i}) \right|^2 \right) \tag{8}$$

Where ∇ stands for the light intensity gradient of a pixel. Equation 8 restraints pixel gradient itself and makes the functional is not merely robust for image texture, but also for texture distribution.

And now, the data term of the functional consists of three parts:

$$FC = BC_{s1} + BC_{s2} + \alpha_g GC \tag{9}$$

with α_g , a weight factor to balance two kinds of consistency proportion, too much gradient consistency will fuzz up objects' edges, on the contrary, too small weight will weaken robustness. And it's usually determined by a scene structure, normally takes 0.5.

3.3.4 Smooth regularization

The main purpose of smooth regularization is to reduce noises of scene flow and inverse depth, but over-smoothing will blur the edge, which leads an error in solution, so we take an anisotropy operator as smooth function.

As we know, traditional methods often choose isotropy operator like Laplacian to smooth scene flow. However, we adopt an operator similar to Evan Herbst [13], it is anisotropic and reduces the smoothness over object edges:

$$L(\vec{X}_I) = 1 - e^{-c \left(\left(rgb(\vec{X}_I) - rgb(\vec{Y}_I) \right)^2 \right)}, \text{ where } \vec{Y}_I \in N(\vec{X}_I) \tag{10}$$

$N(\vec{x})$ are adjacent pixels of pixel \vec{x} on referenced image frame. So we can get the smooth regularization term as:

$$S_f(\vec{V}) = \psi\left(\vec{V}(\vec{X}_{I_0})_T L(\vec{X}_{I_0}) \vec{V}(\vec{X}_{I_0})\right) \quad (11)$$

Adopting Eq. 11 as smooth term, scene flow's similarity in object internal area can be ensured, on the other hand differences on objects' edges can be kept.

Same as traditional methods, we take a Laplacian to do depth smooth as in Eq. 12, for the correlation between depth and edges is not so strong compared with scene flow.

$$S_z(d) = \psi\left(|\nabla d(X_{I_0})|^2\right) \quad (12)$$

We get the smoothness term as following:

$$FS = \beta_f S_f + \beta_z S_z \quad (13)$$

with β_f and β_z as weight factors.

In conclusion, we can derive the integrated energy functional in Eq. 14.

$$\begin{aligned} E(d, \vec{V}) &= \int_{\Omega} (FC + \alpha_s FS) dudv \\ &= \int_{\Omega} [BC_{s1} + BC_{s2} + \alpha_g GC + \alpha_s (\beta_f S_f + \beta_z S_z)] dudv \end{aligned} \quad (14)$$

Where internal area Ω is the image domain, which means the functional solution is aimed at the whole image.

As in Eq. 14, the integrated energy functional is about object points depth and scene flow, and it is nonlinear. So, we convert it to a variational problem by considering Euler-Lagrange condition and obtain a linear iterative equation to get numerical solution.

3.3.5 Occlusion estimation

In scene flow estimation, occlusion points may appear at any position because of the dynamics of the scene, so it is necessary to get rid of occluded points before the main iterative process. As shown in Algorithm 1, for a moving camera, its optic center change may lead to boundary occlusion, so we compute the COP position for the i th frame as initialization again. After acquiring the boundary occlusion map, every 3D point corresponds to reference image pixel is projected to current time t image coordinate. If two 3D points have the same 2D image coordinate, the point with further COP is set occluded and its corresponding reference image coordinate will be marked for the time i frame. Thus, scene flow estimation without occluded points becomes more accurate, which only need few steps before real estimation begins.

Algorithm 1 Occluded Map Estimation**Input:** Image Resolution**Initialization:** $m_s c^i(p_i)$ - the occluded sign of p_i at time i , 1 means the point is occluded

- 1: $m_s c^i \leftarrow 1$ { $m_s c^i$ -the occlusion map of time i frame, 0 means the pixel in reference coordinate frame is occluded }
- 2: $c \leftarrow NULL$ { c -the temporary map to record occlusion point coordinate }
- 3: $COP^i \leftarrow 0$ { COP- the optic center of time i frame in 3D world space }
- 4: $RangeX \leftarrow [0, imageWidth]$ {RangeX - the image range of X direction }
- 5: $RangeY \leftarrow [0, imageHeight]$ {RangeY - the image range of Y direction }
- 6: **for** every point p_0 at reference frame **do**
- 7: $P(p_0) \leftarrow$ 3D point correspond to p_0 in 3D world space
- 8: $p_i \leftarrow$ reproject($P(p_0)$)
- 9: **if** $c(p_i) \neq NULL$ **then**
- 10: $P_1 \leftarrow$ 3D point correspond to $m_s c^i(p_i)$ in 3D world space
- 11: $P_2 \leftarrow$ 3D point correspond to p_i in 3D world space
- 12: **if** $distance(P_1, COP^i) < distance(P_2, COP^i)$ **then**
- 13: P_1 is nearer to COP^i than P_2
- 14: $m_s c^i(p_i) \leftarrow 0$
- 15: **else**
- 16: $c(p_i) \leftarrow p_i$
- 17: $m_s c^i(p) \leftarrow 0$
- 18: **end if**
- 19: **else**
- 20: $c(p_i) \leftarrow p_i$
- 21: $m_s c^i(p_0) \leftarrow 1$
- 22: **end if**
- 23: **if** $m_s c^i(p_i) = 0$ OR p_i in RangeX OR p_i in RangeY **then**
- 24: $m_s c^i(p_i) \leftarrow 0$
- 25: **end if**
- 26: **end for**
- 27: **return** $m_s c^i(p_i)$ for time i frame

4 Solving the energy functional

Now we get the integrated energy functional to solve scene flow of monocular moving camera, through mathematical model brightness and smooth assumption in section 3, it is a convex nonlinear functional. Firstly, abbreviate the equation:

$$\begin{aligned}
 \varphi_i &= I_i - I_i(\vec{X}_{I_i}) \\
 \hat{\varphi}_i &= I_0 - I_i(\vec{X}_{I_i}) \\
 \nabla\varphi_i &= \nabla I_0 - \nabla I_i(\vec{X}_{I_i})
 \end{aligned}
 \tag{15}$$

So we can rewrite the energy Functional as following:

$$E(\vec{V}, d) = \int_{\Omega} F(\vec{V}, d) dudv = \int_{\Omega} \left[\sum_{i=1}^n \psi(\varphi_i^2) + \sum_{i=0}^n \psi(\hat{\varphi}_i^2) + \sum_{i=1}^n \psi(\nabla\varphi_i^2) + \psi(\vec{V}^T L \vec{V}) + \psi(|\nabla d|^2) \right] dudv
 \tag{16}$$

Because Eq. 16 has continuity and differentiability in the field of definition, we can rewrite it according to Euler-Lagrange as an equation to inverse depth d:

$$\begin{aligned}
 0 &= \frac{\delta F}{\delta d} - \frac{\delta}{\delta x} \left(\frac{\delta F}{\delta d} \right) = \sum_{i=0}^n \psi'(\varphi_i^2) \varphi_i (\delta\varphi_i / \delta d) + \sum_{i=1}^n \psi'(\hat{\varphi}_i^2) \hat{\varphi}_i (\delta\hat{\varphi}_i / \delta d) \\
 &\quad + \sum_{i=0}^n \psi'(\varphi_i^2) \nabla\varphi_i (\delta\nabla\varphi_i / \delta d) - \alpha_s \beta_z \operatorname{div} \left(\psi'(|\nabla d|^2) \right) \nabla d
 \end{aligned}
 \tag{17}$$

for v_x in \vec{V} , the iterative equation is:

$$\begin{aligned}
 0 &= \frac{\delta F}{\delta v_x} - \frac{d}{dx} \left(\frac{\delta F}{\delta v_x} \right) = \sum_{i=0}^n \psi'(\varphi_i^2) \varphi_i (\delta\varphi_i / \delta v_x) + \sum_{i=1}^n \psi'(\hat{\varphi}_i^2) \hat{\varphi}_i (\delta\hat{\varphi}_i / \delta v_x) \\
 &\quad + \sum_{i=0}^n \psi'(\nabla\varphi_i^2) \nabla\varphi_i (\delta\nabla\varphi_i / \delta v_x) + \psi'(\vec{V}^T L \vec{V}) L \vec{V}
 \end{aligned}
 \tag{18}$$

Derivation for the other value v_y and v_z are the same. Thus, we convert the functional solving problem to an optimization problem, which means we should find the optimal solution under above equations.

5 Numerics

The existence of local minimum often lead to errors in solving optimization problems, so we use a L2 norm ψ to ensure functional convexity, which makes iterative process converge to a

global minimum. Because the back projection and projection are non-linear, we use the first order Taylor expansion to linearize scene flow and depth.

$$\begin{aligned} \vec{V}^{k+1} &= \vec{V}^k + \delta \vec{V}^k \\ d^{k+1} &= d^k + \delta d^k \end{aligned} \tag{19}$$

Major components in the optimization can be rewritten as:

$$\begin{aligned} \varphi_i^k &= I_i^k - I_i(\vec{X}_{I_i}^k)^k \\ \hat{\varphi}_i^k &= I_0 - I_i(\vec{X}_{I_i}^k)^k \\ \nabla \varphi_i^k &= \nabla I_0 - \nabla I_i(\vec{X}_{I_i}^k)^k \end{aligned} \tag{20}$$

Thus, with the first order Taylor expansion, the value at $(k+1)$ th iterative time can be formed by that of k th iterative time:

$$\begin{aligned} \varphi_i^{k+1} &\approx \varphi_i^k + \delta d \varphi_i^k \nabla d^k + \delta_{v_x} \varphi_i^k \nabla v_x^k \\ &+ \delta_{v_y} \varphi_i^k \nabla v_y^k + \delta_{v_z} \varphi_i^k \nabla v_z^k \\ \hat{\varphi}_i^{k+1} &\approx \hat{\varphi}_i^k + \delta v_x^k \hat{\varphi}_i^k \nabla v_x^k + \delta_{v_x} \hat{\varphi}_i^k \nabla v_x^k \\ &+ \delta_{v_y} \hat{\varphi}_i^k \nabla v_y^k + \delta_{v_z} \hat{\varphi}_i^k \nabla v_z^k \\ \nabla \varphi_i^{k+1} &\approx \nabla \varphi_i^k + \delta v_x^k \nabla \varphi_i^k \nabla v_x^k + \delta_{v_x} \nabla \varphi_i^k \nabla v_x^k \\ &+ \delta_{v_y} \nabla \varphi_i^k \nabla v_y^k + \delta_{v_z} \nabla \varphi_i^k \nabla v_z^k \end{aligned} \tag{21}$$

At last, by setting the threshold value, we can get follow iterative equation for inverse depth:

$$\begin{aligned} 0 &= \sum_{i=0}^n \psi' \left((\varphi_i^{k+1})^2 \right) \varphi_i^{k+1} (\delta \varphi_i^k / \delta d) \\ &+ \sum_{i=1}^n \psi' \left((\hat{\varphi}_i^{k+1})^2 \right) \hat{\varphi}_i^{k+1} (\delta \hat{\varphi}_i^k / \delta d) \\ &+ \sum_{i=0}^n \psi' \left((\nabla \varphi_i^{k+1})^2 \right) \nabla \varphi_i^{k+1} (\delta \nabla \varphi_i^k / \delta d) \\ &- \alpha_s \beta_z d \ i \ v \left(\psi' (\nabla d^{k+1}) \right) \nabla d^{k+1} \end{aligned} \tag{22}$$

The iterative process is presented in [Appendix](#).

6 Experiments

Experiments were carried out on self-synthetic data sets, general synthetic data sets and real scene data sets. Three data sets have different settings for the scene and the movement of objects. Experimental results were compared with those of current general two scenes flow estimation methods by calculating the root mean square error (RMSE).

Table 1 Experimental environment

Hardware		Software	
CPU	2.40GHz (4CPUs)	OS	Windows 7 Ultimate 32
RAM	4096 MB	IDE	Visual Studio 2010
D3D	DirectX 11	Tool	OpenCV 2.4.1 WPF Unity3D
Process:	Intel core i5-2430 M	Language	C++,C,Matlab

Table 1 gives the experimental environment.

Table 2 shows the test dataset statistics.

6.1 Data setup

Three data sets include two synthetic data sets and one real data set. Dataset 1 is generated from Unity3D, by combining the existing camera calibration parameters into virtual camera projection matrix. Dataset 2 comes from Reinhard Klette's (Image Sequence Analysis Test Site) dataset [19], which is a subset of EISATS. Dataset 3 is real scene from the series of 2011_09_26_drive_0018 (1.1 GB) in KITTI [11]. Table 2 shows the basic properties of each data set, since EISATS provides complete information with ground truth and ego-motion, comparative experiment and error calculation were carried out on dataset 2.

6.1.1 Dataset 1 detailed description

Dataset 1 simulates a dynamic ball in the front of a static plane, by putting calibration parameters of a real camera into a virtual camera in Unity3D, the focal length of the camera is [657, 658]. As Fig. 3 shows, the plane center position is (0,0,80), the initial position of the ball is (-1,3,-1), the ball is moving with a constant speed (1.0, 2.0, 0.0), and the camera is static. We assume that there is no distortion of camera and no offset of optic center. In order to make the scene visible to the camera, we also use a point light as ambient light. The material of background plane is a carpet with repeated texture, which is used to prove our estimation results will not be influenced by texture distribution. When the experiment began, we got the integer value map of scene by shade, and then set it as the initial depth of our algorithm.

Table 2 Dataset statistics

Dataset	Type	Source	Resolution	Additional Information
1	Synthetic		(640, 480)	Calibration parameters, Depth
2	Synthetic		(1392,512)	Ego-motion, Calibration parameters, Groundtruth information
3:	Real		(1392,512)	Calibration parameters

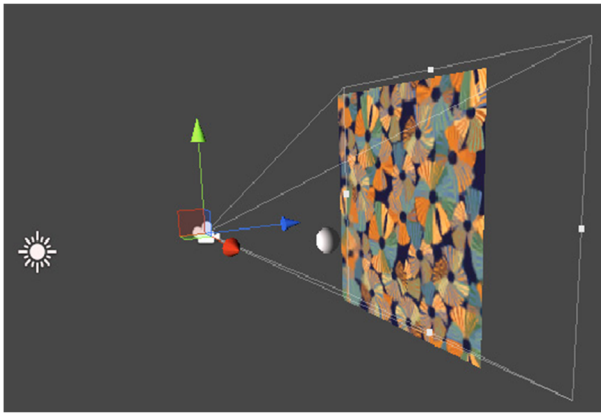


Fig. 3 Dataset 1 setup. Only one spot light in the scene as ambient light, and the ball is moving in front of a carpet textured plane, the camera frustum matrix is set with real camera calibration parameters

6.1.2 Dataset 2 detailed description

The Image Sequence Analysis Test Site (EISATS) offers sets of image sequences for the purpose of comparative performance evaluation of stereo vision, optic flow, motion analysis, or further

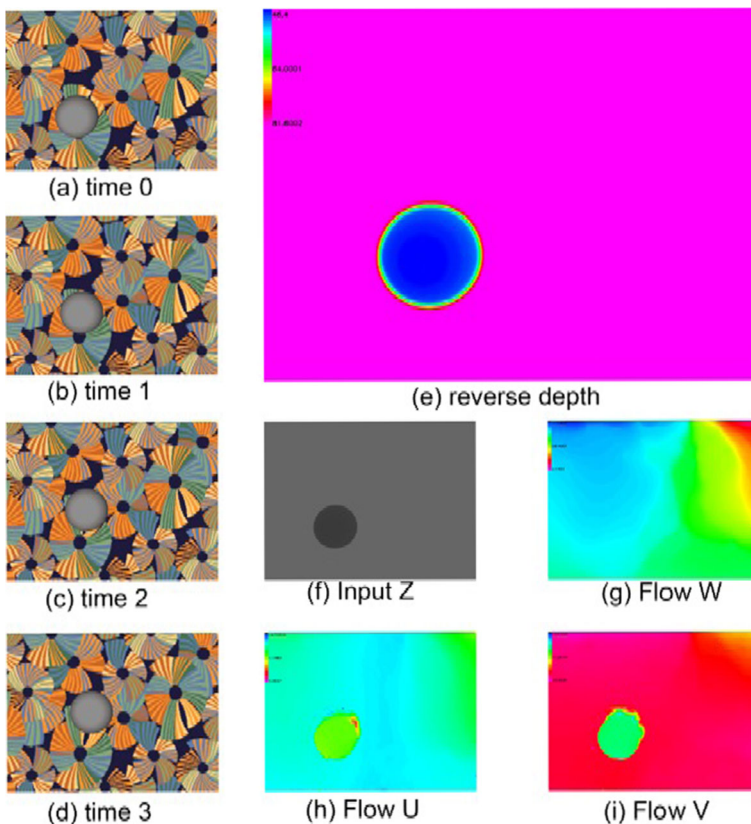


Fig. 4 Our monocular scene flow estimation results on dataset 1. **h**, **i** and **g** are estimated X,Y,Z directional scene flows

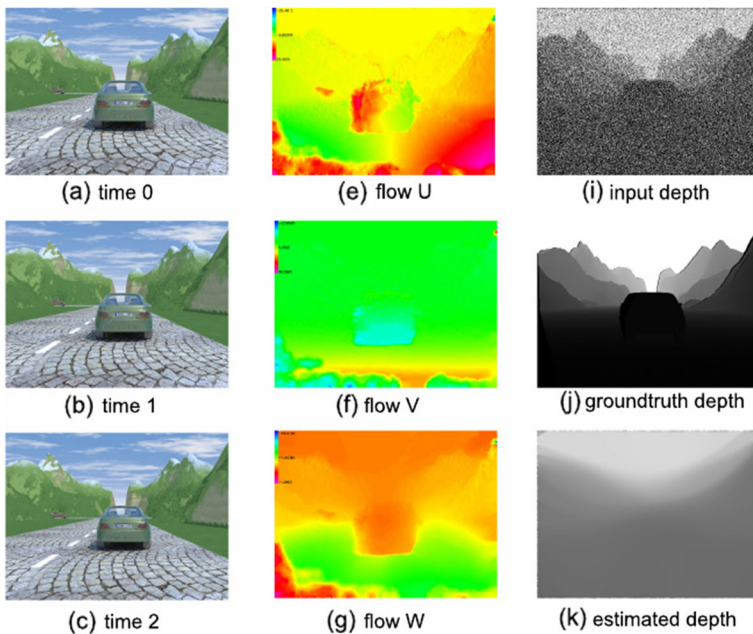


Fig. 5 Our monocular scene flow estimation results on dataset 2. **a-c** Input frames. **e-g** Estimated scene flow results. **i-k** corresponding depths

techniques in computer vision. We chose the Synthesized (gray-level and color) sequences, because it offers camera calibration information, camera ego-motion and ground truth information.

6.1.3 Dataset 3 detailed description

The KITTI offers real stereo traffic datasets under different situations and we chose a typical crossroad scene dataset to testing.

6.2 Experimental results

In the experiment, we make camera motion information and monocular successive image frames as input, the output is text representation of the scene flow estimation results and

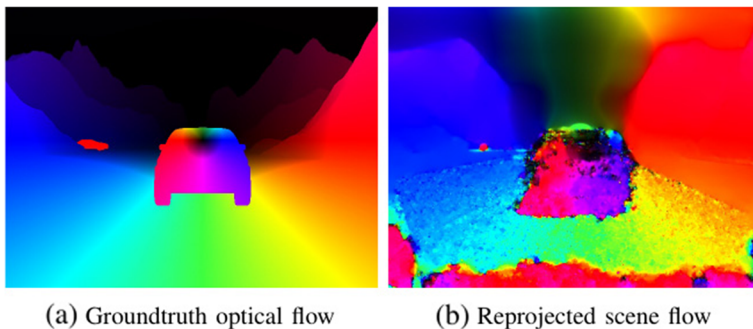


Fig. 6 Re-projected scene flow on image plane

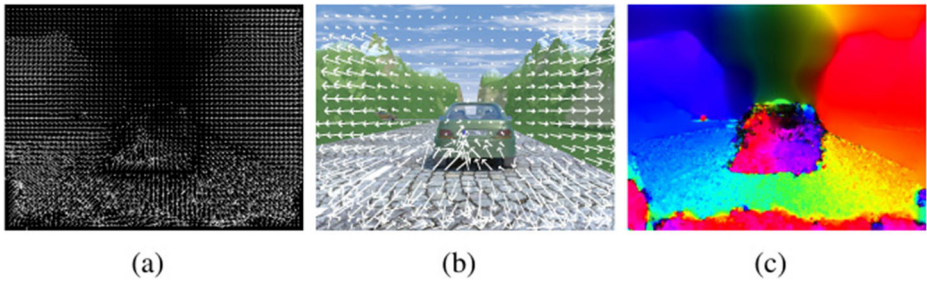


Fig. 7 Scene flow estimated results of our method on dataset 2

reverse depth information. By mapping results onto a two-dimensional plane displaying in HSV color space, the moving distinction can be seen directly. In this paper, we computed accuracy of the scene flow by calculating the root mean square error (RMSE) as Eq. 23. Since dataset 2 provides accurate ground truth information, two state-of-art scene flow estimation algorithms were compared with ours on dataset 2.

$$RMSE = \sqrt{\frac{\int_{\Omega} (\vec{V}_{result} - \vec{V}_{groundtruth})^2}{n}} \quad (23)$$

6.2.1 Analysis of the experimental results on dataset 1

As in Fig. 4, we chose four sequential frames from dataset 1 to test the algorithm, in order to accelerate the iterate procedure, the integer representation of the depth map was used as the initial depth. The experimental results shows that scene flow and inverse depth can be seen clearly in HSV representation. That means, under static camera condition, our algorithm can restore a more realistic point cloud motion information (e.g., The ball has no movement in the Z direction in scene flow result, which is same as real condition), and get more accurate depth information.

6.2.2 Analysis of the experimental results on dataset 2

As in Fig. 5, we first extracted three consecutive frames from EISATS stereo datasets as input frames, then added white noise and blurring to the ground truth depth image, made it as initial depth (for more close to real scene). When iteration began, the ego-motion was combined into

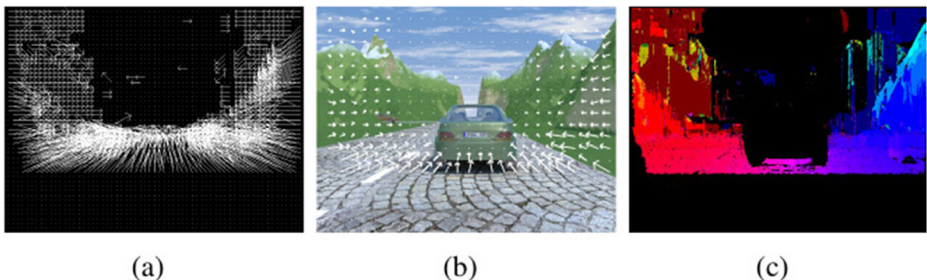


Fig. 8 Scene flow estimated results of GCSF on dataset 2

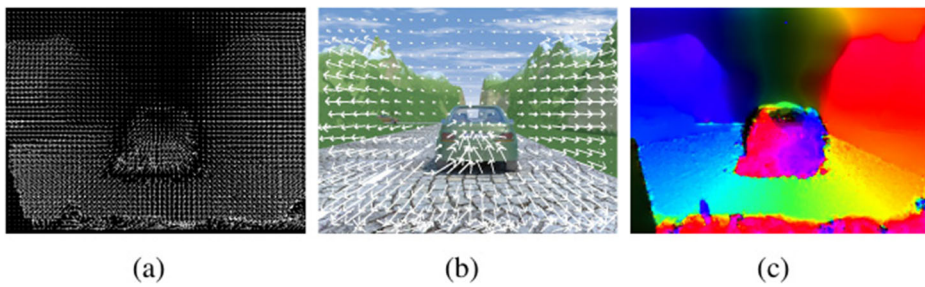


Fig. 9 Scene flow estimated results of MVSF on dataset 2

camera matrix. Figure 5 shows, monocular scene flow can estimate scene flow under dynamic camera relatively accurate, even with noise interference.

In addition, since the dataset 2 provides a complete ground truth optical flow motion information, the scene flow accuracy can be evaluated by calculating the RMSE on 2D projected image [3]. So we first re-projected scene flow on image plane as in Fig. 6, and then computed the RMSE with ground truth flow under different pixel threshold respectively. For evaluating the effectiveness of paper method, we also computed RMSEs of two state-of-art algorithms, GCSF and MVSF. Our method result is shown in Fig. 7. GCSF is a simple seed growing algorithm for estimating scene flow in a stereo setup, and it needs two calibrated and synchronized cameras to observe a scene, simultaneously computes disparity map between the image pairs and optical flow maps between consecutive images [17]. GCSF's estimated result is shown in Fig. 8. MVSF includes a 3D point cloud parameterization of the 3D structure, which can directly estimate the desired unknowns, and its energy functional enforces multi-view geometric consistency and imposes brightness constancy and piecewise smoothness assumptions directly on the 3D unknowns [4]. MVFS's estimated result is shown in Fig. 9. According to the scene flow assessment methods of KITTI, we computed the percentage of erroneous pixels in total for three algorithms, under different pixel error threshold. As Fig. 10 displays, our algorithm can achieve similar accuracy with state-of-art stereo scene flow algorithms, with only one camera, which removes the complexity of stereo calibration and camera synchronization.

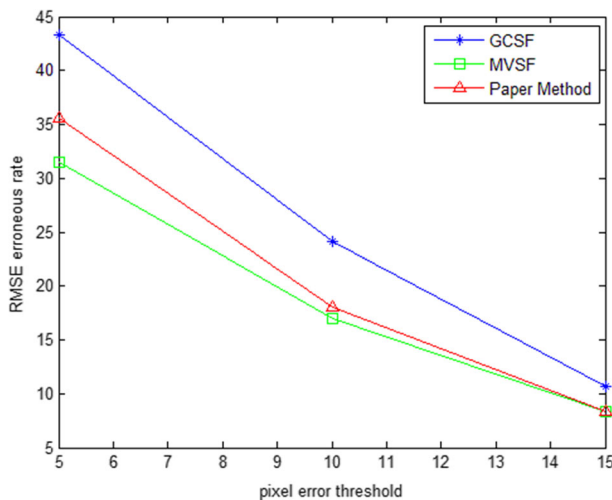


Fig. 10 Comparison of three algorithms. X axis is pixel error threshold and Y is erroneous percentage

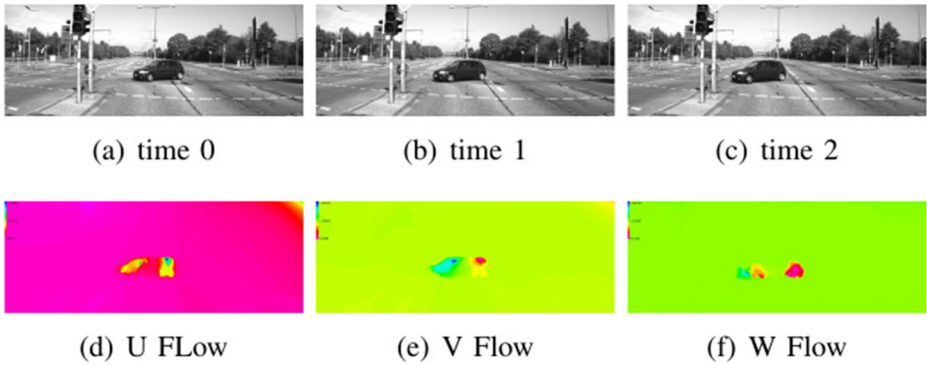


Fig. 11 Our monocular scene flow estimation results on dataset 3. **d-f** are estimated X,Y,Z directional scene flows

6.2.3 Analysis of the experimental results on dataset 3

We also process experiments on real scene of datasets 3, since there is no ego-motion in data, we adopted a static sequence to verify the algorithm, and the initial depth set as unified 200 cm. As Fig. 11 shows, the monocular scene flow algorithm can still get a more accurate estimate of the results in the real scene, without any other information except for camera intrinsic matrix.

7 Conclusion

This paper proposes a scene flow estimation algorithm for monocular image sequences, and innovatively combines inverse depth to consistency functional. Different from traditional methods, this monocular scene flow method:

- 1) Needs only one camera with existing navigation system, which makes it more flexible in traffic environment;
- 2) Restores the depth and scene flow simultaneously by putting inverse depth into total functional, gets the cloud points position and moving information at the same time;
- 3) Takes an anisotropic operator for scene flow smoothing and an isotropy operator for inverse depth smoothing, which maintains disparity between objects in the scene, and reduces noise in object area, making the results more close to nature;
- 4) Makes 3 reasonable assumptions according to dynamic scene attributes, extends coarse-to-fine framework to monocular scene flow estimation and gets the global minimum of the total energy functional as numerical solution.

The solution accuracy depends on velocity consistency in this algorithm, when scene objects take tiny and continuous movements, the estimation result will be good. If an object takes a non-rigid motion, the algorithm may be not so ideal. The future work of this paper will focus on overcoming similar problems and pay more attention to its application in related areas, such as societal health.

Acknowledgments The authors would like to thank the anonymous reviewers for their insightful comments and suggestions. This work is supported in part by National Natural Science Foundation of China (Grant No.61272062, 61300036), the Projects in the National Science & Technology Pillar Program (Grant No.2013BAH38F01).

Appendix

Iterative process

Our iterative process is divided into two layers. The outer layer is constructed by the Gaussian pyramid from coarse to fine, iterations at this level is for getting unknown quantity itself. At each layer of the pyramid, the inner iteration can get the unknown variables incremental from SOR iteration. As Fig. 12 shows, the Gaussian pyramid layers is built according to input outer layer iteration value, during the build process, we calculate the scaling factor for each layer. In order to ensure correct correspondence between the image space and the world space, the scaling factor should not only work for image resolution, but also for focal length of the camera and the optical center position. At the inner iteration process, the scene flow quantity initial values are set to zero. By setting the value of inner iteration times, starting from the lowest resolution level of Gaussian pyramid, SOR iteration obtains the unknown value increment before convergence or reaching the number of iterations. Every final value of inner iteration will be added to current outer layer, and set as initial value of next outer layer. Algorithm 2 shows the whole iteration process.

It is necessary to determine the number of iterations of the inner and outer layers in the iterative process. The number of iterations of outer layers determines the

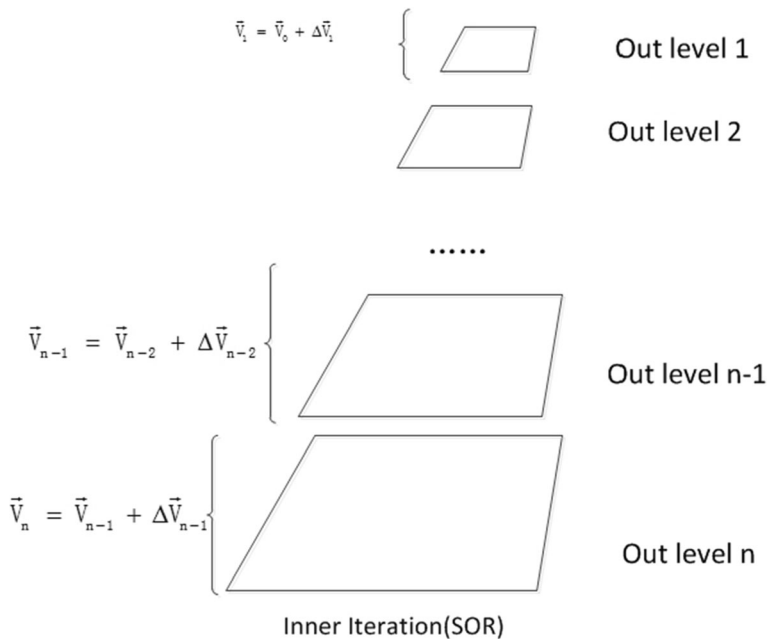


Fig. 12 The two-level iteration, the outer iteration is processed on Gaussian pyramid layers, the inner iteration is processed on each outer layer by SOR method

pyramid layers. Our experiments set the outer iteration number as 10 due to memory limitations. Figure 13 shows the relationship between erroneous percentage and inner iteration numbers, the iterations are processed at the last outer layer. We set the number of inner iteration as 10, for the polyline shows: iterations more than 10 will cause over smooth.

Algorithm 2 Iteration Process

Input: n, k, d_0 { n - the total number of outer iterations, k - the total number of inner iterations, d_0 - the initial depth(from depth map or a unified value)}

Initialization: $\log_2 ImageSize < n$ { Make sure Gaussian pyramid is suitable for the images' resolution}

- 1: Build Gaussian pyramid of input images by setting layers as n
 - 2: $i \leftarrow 0$ { i - current number of outer iterations }
 - 3: $\vec{V}_0 \leftarrow \vec{0}, \Delta\vec{V}_0 \leftarrow \vec{0}_0$ { \vec{V}_i - the unknown value of i th outer iteration, $\Delta\vec{V}_i^j$ - the unknown value of j th inner iteration of i th outer iterationn }
 - 4: **while** $i < n$ **do**
 - 5: $j = 0$ { j - the current number of inner iterations }
 - 6: **while** $j < k$ and $\|\Delta\vec{V}_i^j - \Delta\vec{V}_i^{j-1}\| > \epsilon$ { ϵ - the threshold of inner iteration, a tiny value} **do**
 - 7: do iteration of $\Delta\vec{V}_i^j$
 - 8: $j = j + 1$
 - 9: **end while**
 - 10: $\vec{V}_i = \Delta\vec{V}_{i-1}^k + \vec{V}_{i-1}$
 - 11: $\Delta\vec{V}_{i+1}^j \leftarrow \vec{0}$
 - 12: **end while**
 - 13: **return** (\vec{V}_n, d_n)
-

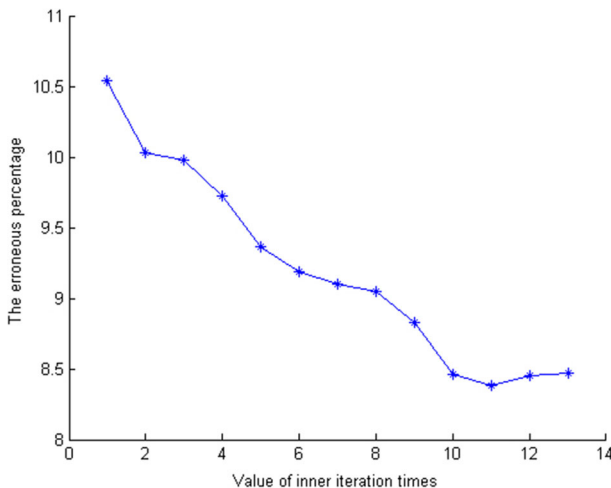


Fig. 13 Erroneous percentages of different number of inner iterations at the last layer of outer iteration

References

1. Adiv G (1985) Determining three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE Trans Pattern Anal Mach Intell* 4:384–401
2. Alcantarilla PF, Yebes JJ, Almazn J, Bergasa LM (2012) On combining visual slam and dense scene flow to increase the robustness of localization and mapping in dynamic environments. *IEEE International Conference on Robotics and Automation (ICRA)* 1290–1297
3. Baker S, Scharstein D, Lewis J, Roth S, Black MJ, Szeliski R (2011) A database and evaluation methodology for optical flow. *Int J Comput Vis* 92(1):1–31
4. Basha T, Moses Y, Kiryati N (2013) Multi-view scene flow estimation: a view centered variational approach. *Int J Comput Vis* 101(1):6–21
5. Birkbeck N, Cobzas D, Jagersand M (2011) Basis constrained 3d scene flow on a dynamic proxy. *IEEE International Conference on Computer Vision (ICCV)* 1967–1974
6. Brox T, Bruhn A, Papenbergh N, Weickert J (2004) High accuracy optical flow estimation based on a theory for warping. *Eur Conf Comput Vis* 25–36. doi:10.1007/978-3-540-24673-2_3
7. Civera J, Davison AJ, Montiel J (2008) Inverse depth parametrization for monocular slam. *IEEE Trans Robot* 24(5):932–945
8. Cruz L, Lucio D, Velho L (2012) Kinect and rgbd images: challenges and applications. *IEEE Conference on Graphics, Patterns and Images Tutorials (SIBGRAPI-T)* 36–49
9. Dame A, Prisacariu V A, Ren C Y, Reid I (2013) Dense reconstruction using 3d object shape priors. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 1288–1295
10. Geiger A, Ziegler J, Stiller C (2011) Stereo scan: dense 3d reconstruction in real-time. *IEEE Conference on Intelligent Vehicles Symposium* 963–968
11. Geiger A, Lenz P, Stiller C, Urtasun R (2013) Vision meets robotics: the kitti dataset. *Int J Robot Res* 32(11):1231–1237
12. Henry P, Krajinin M, Herbst E, Ren X, Fox D (2012) Rgb-d mapping: using kinect-style depth cameras for dense 3d modeling of indoor environments. *Int J Robot Res* 31(5):647–663
13. Herbst E, Ren X, Fox D (2013) Rgb-d flow: dense 3-d motion estimation using color and depth. *IEEE International Conference on Robotics and Automation (ICRA)* 2276–2282
14. Hornacek M, Rhemann C, Gelautz M, Rother C (2013) Depth super resolution by rigid body self-similarity in 3d. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 1123–1130
15. Huang S, Dissanayake G (2007) Convergence and consistency analysis for extended kalman filter based slam. *IEEE Trans Robot* 23(5):1036–1049
16. Izadi S, Kim D, Hilliges O, Molyneux D, Newcombe R, Kohli P, Shotton J, Hodges S, Freeman D, Davison A (2011) Kinect fusion: real time 3d reconstruction and interaction using a moving depth camera. *ACM symposium on User interface software and technology* 559–568
17. Jan Č, Sanchez-Riera J, Horaud R (2011) Scene flow estimation by growing correspondence seeds. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 3129–3136
18. Khoshelham K, Elberink S O (2012) Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors* 1437–1454
19. Klette R (2015) <http://ccv.wordpress.fos.auckland.ac.nz/eisats/>. Accessed 2 May 2014
20. Letouzey A, Petit B, Boyer E (2011) Scene flow from depth and color images. *Proc Br Mach Vis Conf* 46:1–11. doi:10.5244/C.25.46
21. Newcombe RA, Lovegrove SJ, Davison AJ (2011) Dtam: dense tracking and mapping in real-time. *IEEE International Conference on Computer Vision (ICCV)* 2320–2327
22. Nie L, Akbari M, Li T, Chua T (2014) A joint local–global approach for medical terminology assignment. *MedIR@SIGIR* 24–27
23. Nie L, Li T, Akbari M, Shen J, Chua T (2014) WenZher: comprehensive vertical search for healthcare domain. *ACM Conference on Research and Development in Information Retrieval (SIGIR)* 1245–1246
24. Nie L, Zhang L, Yang Y, Wang M, Hong R, Chua T (2015) Beyond doctors: future health prediction from multimedia and multimodal observations. *ACM Multimedia* 591–600
25. Nie L, Zhao Y, Akbari M, Shen J, Chua T (2015) Bridging the vocabulary gap between health seekers and healthcare knowledge. *IEEE Trans Knowl Data Eng* 27(2):396–409
26. Nie L, Wang M, Zhang L, Yan S, Zhang B, Chua T (2015) Disease inference from health-related questions via sparse deep learning. *IEEE Trans Knowl Data Eng* 27(8):2107–2119
27. Stoyanov D (2012) Stereoscopic scene flow for robotic assisted minimally invasive surgery. *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2012*:479–486
28. Vedula S, Baker S, Rander P, Collins R, Kanade T (1999) Three dimensional scene flow. *IEEE Int Conf Comput Vis* 2:722–729
29. Vogel C, Schindler K, Roth S (2011) 3d scene flow estimation with a rigid motion prior. *IEEE International Conference on Computer Vision (ICCV)* 1291–1298

30. Wedel A, Brox T, Vaudrey T, Rabe C, Franke U, Cremers D (2011) Stereoscopic scene flow computation for 3d motion understanding. *Int J Comput Vis* 95(1):29–51
31. Yan Y, Ricci E, Subramanian R, Lanz O, Sebe N (2013) No matter where you are: flexible graph-guided multi-task learning for multi-view head pose classification under target motion. *IEEE International Conference on Computer Vision (ICCV)* 1177–1184
32. Yan Y, Liu G, Ricci E, Sebe N (2014) Multi-task linear discriminant analysis for multi-view action recognition. *IEEE Trans Image Process (TIP)* 23(12):5599–5611
33. Yan Y, Yang Y, Meng D, Liu G, Tong W (2015) Event oriented dictionary learning for complex event detection. *IEEE Trans Image Process (TIP)* 24(6):1867–1878
34. Yan Y, Ricci E, Liu G, Sebe N (2015) Egocentric daily activity recognition via multitask clustering. *IEEE Trans Image Process (TIP)* 24(10):2984–2995
35. Yang Z, Xiong Z, Zhang Y, Wang J, Wu F (2013) Depth acquisition from density modulated binary patterns. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 25–32
36. Zhang Z (2012) Microsoft kinect sensor and its effect. *IEEE MultiMedia* 19(2):4–10



Degui Xiao received the B.E. degree in industrial automation from the Wuhan University of Textile, Wuhan, P. R. China in 1994, and the Ph.D. degree in computer science and technology from Huazhong University of Science and Technology, Wuhan, P. R. China, in 2003. From 2003 to 2005, he was an Assistant Professor with the College of Computer and Communication, Hunan University, Changsha, P. R. China. Since 2005, he has been an Associate Professor with the College of Computer Science and Electronic Engineering, Hunan University. From 2006 to 2007, he was a Visiting Scholar for seismic data process and visualization with the Rock Physics Lab, Department of Earth and Atmospheric Sciences, University of Houston, Houston, Texas, USA. His current research interests include image and video processing, computer vision, big data computing.



Qiuwei Yang is an assistant professor in the Department of Computer Science and Electrical Engineering (CSEE) at University of Hunan, China (April 2008 - Present), and as a post doctor present. Before this, he got PHD in Huazhong University of Technology. Currently, His interests are multimedia security, network security and privacy protection.



Bing Yang is an associate professor in the School of Education at University of Hubei, China. He got PHD in Huazhong University of Technology. Currently, His interests are multimedia, wireless sensor network, and mobile learning.



Wei Wei received the Ph.D. degree from the Huazhong University of Science and Technology, Wuhan, China, in 2012. He is currently an assistant professor with the School of Computer of Science and Technology, Huazhong University of Science and Technology. He was a Research Fellow with Nanyang Technological University, Singapore, and Singapore Management University, Singapore. His current research interests include information retrieval, data mining, social computing, and natural language processing.