Consistent Binocular Depth and Scene Flow with Chained Temporal Profiles

Chun Ho Hung · Li Xu · Jiaya Jia

Received: 5 November 2011 / Accepted: 8 August 2012 / Published online: 24 August 2012

Abstract We propose a depth and image scene flow estimation method taking the input of a binocular video. The key component is motion-depth temporal consistency preservation, making computation in long sequences reliable. We tackle a number of fundamental technical issues, including connection establishment between motion and depth, structure consistency preservation in multiple frames, and long-range temporal constraint employment for error correction. We address all of them in a unified depth and scene flow estimation framework. Our main contributions include development of motion trajectories, which robustly link frame correspondences in a voting manner, rejection of depth/motion outliers through temporal robust regression, novel edge occurrence map estimation, and introduction of anisotropic smoothing priors for proper regularization.

Keywords Video depth estimation · Consistent scene flow · Chained temporal profiles · Stereo matching

1 Introduction

In many computer vision tasks, reliable depth and motion estimation fundamentally assures high quality result production. If depth can be accurately inferred in 3D videos with necessary temporal consistency, traditionally challenging video editing to alter color, structure, and geometry, as well as the high-level scene understanding and recognition tasks can be accomplished much more easily. In addition, with the precipitate prevalence of 3D display and 3D capturing devices, the "2D-plus-depth" format becomes common and important, as it can be used to generate new views[1] for 3DTV.

Although a tremendous number of binocular videos have come into existence, with only two views, it is still very difficult to compute reliable and consistent depth in long sequences. Structure-from-motion (SFM) and multi-view stereo matching can be applied to static scenes where global constraints are established through the multi-view geometry (Snavely et al. 2006; Furukawa and Ponce 2007; Zhang et al. 2009). It is not suitable for videos that contain moving or deforming objects, which handicap correspondence establishment across multiple frames.

In optical flow estimation (Baker et al. 2011), which captures 2D apparent motion, correspondence between consecutive frames can be established. 2D optical flow and depth variation are jointly considered in Patras et al. (1996), Zhang and Kambhamettu (2001), Vedula et al. (2005), Huguet and Devernay (2007), Wedel et al. (2008, 2011), Valgaerts et al. (2010), Rabe et al. (2010), which is typically referred to as image scene flow. Given intrinsic camera parameters, 3D scene flow can be constructed (Basha et al. 2010; Wedel et al. 2011). These methods either compute motion and depth independently or resort to a four-image configuration. They do not tackle the temporal-consistency preservation problem in multiple frames.

Electronic supplementary material The online version of this article (doi:10.1007/s11263-012-0559-y) contains supplementary material, which is available to authorized users.

C.H. Hung · L. Xu · J. Jia (✉)
Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong, China
e-mail: leojia@cse.cuhk.edu.hk

C.H. Hung
e-mail: chhung@cse.cuhk.edu.hk

L. Xu
e-mail: xuli@cse.cuhk.edu.hk

[1] 2D-Plus-Depth: (2009). Stereoscopic video coding format. http://en.wikipedia.org/wiki/2D-plus-depth.

Local temporal constraints were imposed in depth estimation for video sequences, known as spatiotemporal stereo matching (Zhang et al. 2003; Richardt et al. 2010). These methods do not cope with object motion across multiple frames and thus are more suitable for static scene videos. In optical flow estimation, the methods presented in Black (1994), Bruhn and Weickert (2005) have temporal terms. They, however, may suffer from two main estimation problems. First, it is difficult or inefficient to perform long range information propagation temporally. Second, erroneous estimates caused by occasional noise, sudden luminance change, and outliers in one frame could influence later results. There is no effective way to measure and reduce estimation errors globally.

We aim at reliable depth and motion estimation from multi-frame binocular videos with appropriate temporal consistency. Our method is not based on global multi-view geometry because dynamic objects do not obey them. We also do not count on the locally established temporal constraint due to its inefficiency in information propagation.

We make several major contributions to construct the system, which can measure and reduce estimation errors in multiple frames. (1) We propose motion trajectories that link reliable corresponding points among frames. It is robust against occasional noise and abrupt luminance variation. (2) We build structure profiles by considering multi-frame edges. Through a voting-like step, only edges reliable in multiple frames are enhanced. (3) Long-range temporal constraints are advocated, based on the robust motion trajectories. Regression then corrects errors and improves estimates temporally. (4) Last but not least, we propose anisotropic smoothing to non-uniformly regularize pixels, incorporating temporal edge information and preventing unconstrained boundary degradation.

2 Related Work

Simultaneous depth and motion estimation from stereo images was studied in Zhang and Faugeras (1992). In Wedel et al. (2008), motion and depth were computed sequentially and independently, assuming that the depth in previous frames is known. To improve the results, depth and motion are jointly estimated, using two stereo pairs (Patras et al. 1996; Zhang and Kambhamettu 2001; Min and Sohn 2006; Huguet and Devernay 2007; Valgaerts et al. 2010). These approaches estimate motion fields from two calibrated cameras, where constraints are established in the four-frame configuration. Temporal consistency may still be a problem.

Recently, Wedel et al. (2011) pointed out that decoupling disparity and motion field computation is advantageous in that different optimization techniques can be applied. A semi-dense scene flow field was computed in Cech et al. (2011) through locally growing correspondence seeds. Instead of modeling image scene flow, Basha et al. (2010) imposed constraints on 3D surface motion. Calibrated multi-view sequences were used. Rabe et al. (2010) applied Kalman filtering to independently computed flow estimates and disparities. Hadfield and Bowden (2011) proposed a particle approach for scene flow estimation, making use of depth sensors and the monocular image camera in Microsoft Kinect. Vogel et al. (2011) incorporated a local rigidity constraint to regularize scene flow estimation. Our method primarily differs from these approaches in the way of enforcing temporal consistency, as multi-frame long-range structure information is made use of.

Efforts have also been put to motion/depth discontinuity handling. Zhang and Kambhamettu (2001) used segmentation and applied piecewise regularization. Xu et al. (2008) applied segmentation model to optical flow estimation. Edge-preserving regularizer (Min and Sohn 2006), image driven regularizer (Sun et al. 2008), and complementary regularizer (Zimmer et al. 2009) were used to preserve motion and depth boundaries. The color edges or segments that are used as guidance are generally hard to be consistent over time, making producing high-quality depth boundary difficult.

In optical flow, two-frame configuration is common (Brox et al. 2004; Bruhn et al. 2005; Zimmer et al. 2009; Xu et al. 2010). A taxonomy of optical flow methods, along with comparisons, is reported on the Middlebury website (Baker et al. 2011). To enforce temporal smoothness, in Brox et al. (2004), Bruhn and Weickert (2005), Bruhn et al. (2005), constraints are yielded by assuming that the flow vectors from consecutive frames at the same image location are similar. It also applies to smoothly-varying motion. Álvarez et al. (2007) enforced symmetry between the forward and backward flow to reject outliers. Assuming constant motion acceleration, Black (1994) enforced temporal smoothness by predicting motion for the next frame using current estimate. These methods only consider consecutive frames, which are not effective and may accumulate errors when propagating information among frames that are far apart.

Using multiple frames, Irani (2002) projected flow vectors onto a subspace and assumed that the resulting matrix has a low rank for noise removal. Global motion in static scenes is considered. To construct chained motion trajectories, particle samples were generated and linked in a probabilistic way (Sand and Teller 2008). In Sundaram et al. (2010), chained trajectories were constructed by bidirectional check of motion vectors based on large displacement optical flow estimation (Brox et al. 2009). The quality of trajectories depends excessively on flow estimate from consecutive frames, making these methods possibly vulnerable to noise and estimation outliers.

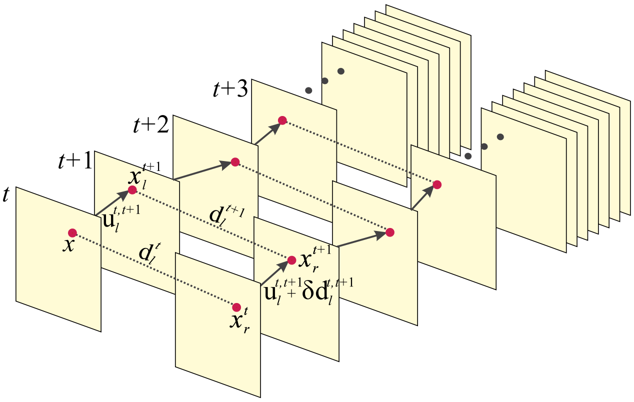


Fig. 1 x correspondences in different frames

To achieve temporal consistency in depth estimation, Zhang et al. (2003) proposed spacetime data costs that aggregate data term over a short period. Richardt et al. (2010) extended the idea by applying spatio-temporal cross-bilateral grid on the data cost, derived from the locally adaptive support aggregation window of Yoon and Kweon (2006). Temporal relationship is also considered upon nearby video frames. These methods did not deal with large motion between frames and thus are more suitable for static scenes.

Contrary to all these approaches, we propose a general binocular framework addressing the temporal consistency problem in depth and motion estimation in long sequences. Temporal information from multiple frames is incorporated with novel chained profiles.

3 Notations and Problem Introduction

Given a rectified binocular video, our method computes two-view stereo for each frame pair across the two sequences, together with dense motion in each sequence. Our framework can be readily extended to unrectified stereo videos linked with a fundamental matrix (Valgaerts et al. 2010).

We denote corresponding frames in the stereo sequences as f_l^t and f_r^t, indexed by time t where $t=\{0,1,\dots,N-1\}$. For each pixel x in frame f_l^t, we find the corresponding pixels in the neighboring frames either temporally using motion estimation or spatially with stereo matching, as shown in Fig. 1. The correspondence x_r^t in f_r^t is expressed as

$$x_r^t = x + \mathrm{d}_l^t(x),$$

where d_l^t is the view-dependent disparity. In this paper, we alternatingly use $\mathrm{d}_l^t(x)$ and d_l^t to represent the disparity value at point x. Meanwhile, optical flow correspondence in the left sequence for pixel x is

$$x_l^{t+1} = x + \mathrm{u}_l^{t,t+1},$$

based on the displacement u in frame f_l^{t+1}. The 2D vector $\mathrm{u}_l^{t,t+1}$ is written as $\mathrm{u}_l^{t,t+1} = (u_l^{t,t+1}, v_l^{t,t+1})^T$, as shown in Fig. 1. Finally, the correspondence for x in f_r^{t+1} is expressed as $x_r^{t+1} = x + \mathrm{u}_l^{t,t+1} + \mathrm{d}_l^{t+1}$, involving both motion and stereo. 3D image scene flow is, by convention, denoted as

$$s^{t,t+1} = \left(u_l^{t,t+1}, v_l^{t,t+1}, \delta \mathrm{d}_l^{t,t+1}\right)^T,$$

where $\delta \mathrm{d}_l^{t,t+1} = \mathrm{d}_l^{t+1}(x + \mathrm{u}_l^{t,t+1}) - \mathrm{d}_l^t$. This representation includes spatial shift and depth variation for correspondences in successive two frames. In this paper, both "depth" and "disparity" are used to denote the displacement of pixels in two views, although, strictly speaking, depth is proportional to the reciprocal of disparity.

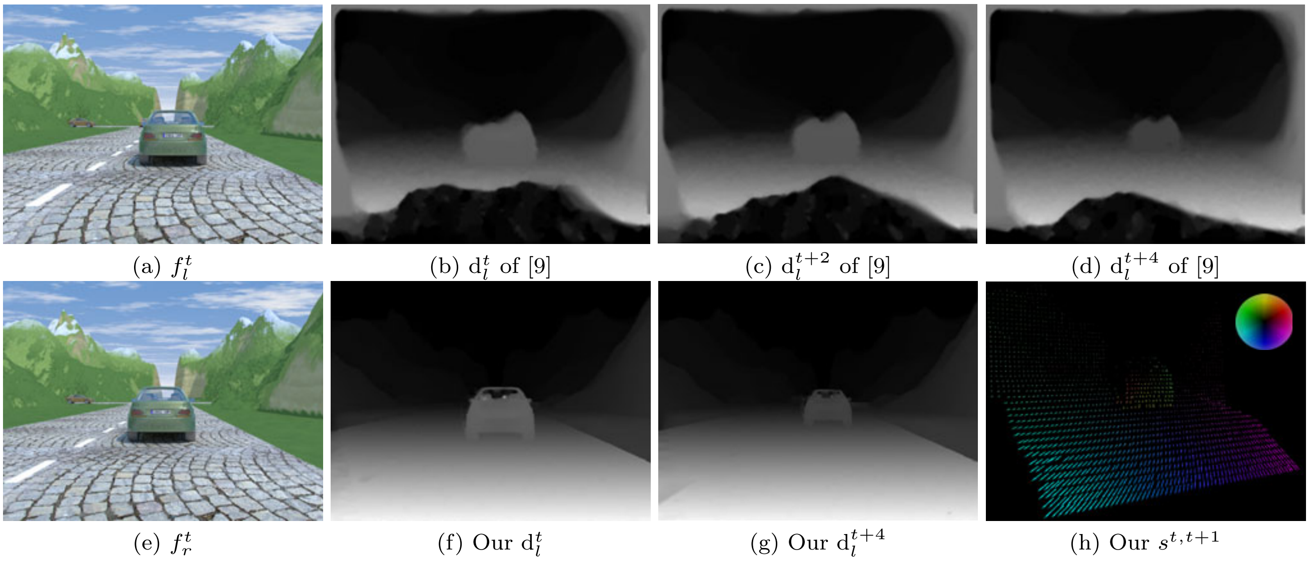
The above correspondences suggest a few fundamental constraints that are used in spatial-temporal depth estimation for every neighboring four frames (illustrated in Fig. 1). The four commonly used scene flow conditions are

$$\begin{aligned}
E_F &= \sum_k \Gamma\left(\left(f_r^{[k]t}\left(x + \mathrm{d}_l^t\right) - f_l^{[k]t}(x)\right)^2\right),\\
E_L &= \sum_k \Gamma\left(\left(f_l^{[k]t+1}\left(x + \mathrm{u}_l^{t,t+1}\right) - f_l^{[k]t}(x)\right)^2\right),\\
E_B &= \sum_k \Gamma\left(\left(f_r^{[k]t+1}\left(x + \mathrm{u}_l^{t,t+1} + \mathrm{d}_l^t + \delta \mathrm{d}_l^{t,t+1}\right)\right.\right.\\
&\qquad \left.\left. - f_l^{[k]t+1}\left(x + \mathrm{u}_l^{t,t+1}\right)\right)^2\right),\\
E_R &= \sum_k \Gamma\left(\left(f_r^{[k]t+1}\left(x + \mathrm{u}_l^{t,t+1} + \mathrm{d}_l^t + \delta \mathrm{d}_l^{t,t+1}\right)\right.\right.\\
&\qquad \left.\left. - f_r^{[k]t}\left(x + \mathrm{d}_l^t\right)\right)^2\right),
\end{aligned} \tag{1}$$

where $[k]$ indexes channels and $\Gamma(\cdot)$ is the robust Charbonnier function, i.e., the variant of L_1 regularizer, written as $\Gamma(y^2) = \sqrt{y^2 + \epsilon^2}$ to reject matching outliers. E_F and E_B are stereo constraints and E_L and E_R are motion constraints for the neighboring-four-frame set. In what follows, we omit the subscript l for all left-view unknowns for simplicity's sake.

In Eq. (1), each f has 5 channels as adopted in Sand and Teller (2006), i.e., $f = (f_I, 1/4(f_G - f_R), 1/4(f_G - f_B), f_{\partial h}, f_{\partial v})$, to make the following computation slightly more robust against illumination variation, compared to only considering RGB colors. f_I is the image intensity; $f_{\partial h}$ and $f_{\partial v}$ are horizontal and vertical intensity gradients.

Key Issues Simultaneously estimating all unknowns, i.e., depth and scene flow, is computationally expensive. It takes hours for the variational method (Huguet and Devernay 2007) to compute one scene flow field for one frame. We also found that only using these constraints cannot produce


(a) f_l^t (b) d_l^t of [9] (c) d_l^{t+2} of [9] (d) d_l^{t+4} of [9]

(e) f_r^t (f) Our d_l^t (g) Our d_l^{t+4} (h) Our $s^{t,t+1}$

Fig. 2 Depth/scene flow estimation example. (**a**) and (**e**) are two corresponding frames in a binocular video. (**b**)–(**d**) show the depth estimates from the joint method of Huguet and Devernay (2007). (**f**) and (**g**) are our depth results. (**h**) visualizes the 3D scene flow

temporally consistent results. As briefed in the introduction, only connecting very close frames lacks representation ability to describe the relationship among correspondences that are far apart in the sequence. This deficiency could cause devastating failure in estimation.

We show one example in Fig. 2 where (a) and (e) are two stereo frames in the binocular sequence. (b)–(d) contain depth maps estimated using the joint method of Huguet and Devernay (2007). Even by simultaneously considering the motion and stereo terms, along with the L_1 regularization, the results are not temporally very consistent.

This is explainable: when all correspondences are *locally* established between successive frames, they are vulnerable to noise, occlusion and illumination variation. Unlike multi-view stereo, there is no global constraint to find and eliminate errors over the sequence. Even with the temporal constraints, when one depth estimate is problematic, all following computation steps can be affected, inevitably accumulating errors and eventually failing estimation. In light of this, other considerations should be taken especially for long sequences.

We propose new chained temporal constraints to make long-range depth-flow estimation less problematic. We show in Fig. 2(f)–(g) our depth results and in (h) our 3D scene flow estimate. Their quality is very high. In what follows, we use gray-scale values to visualize disparity maps and 3D arrows to visualize image scene flow. 2D optical flow is color-coded according to the wheel in Fig. 2(h), in which hue represents motion direction and intensity indicates flow magnitude. 3D scene flow vectors are similarly color coded for their first two dimensions.

Algorithm 1 Outline of Our Method

```
INPUT: a rectified stereo sequence
  1. Initialize motion fields and disparity maps.
     (Sect. 4.1)
  2. Establish temporal constraints. (Sect. 4.2)
     2.1 Build robust trajectories.
     2.2 Compute edge occurrence maps.
     2.3 Compute trajectory-based depth and motion
         using robust regression.
  3. Refine depth and scene flow with global temporal
     constraints. (Sect. 4.3)
OUTPUT: disparity maps and scene flow
```

4 Our Approach

Our method is to generate consistent depth and scene flow from only a binocular video, utilizing pixel correspondence among multiple frames. The overview of our system is given in Algorithm 1, which consists of main steps of initialization, temporal constraint establishment, and final depth and joint scene flow refinement.

4.1 Initialization

To initialize depth and motion, by convention, we apply the variational method to optical flow estimation and use discrete optimization for two-view stereo matching, the latter of which is capable of estimating large disparity.

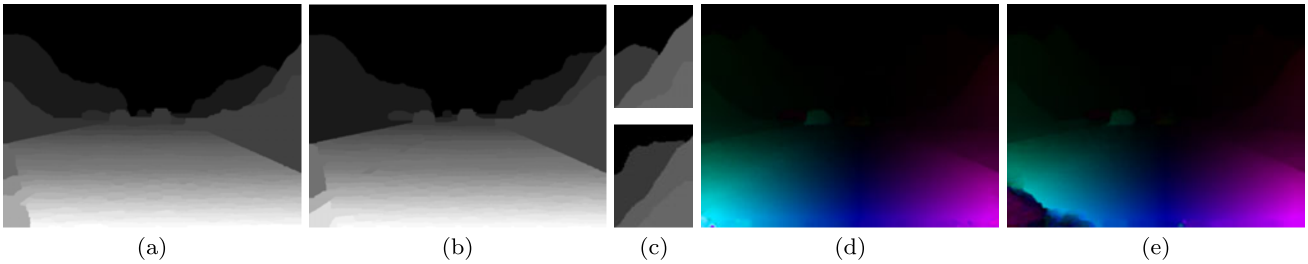


Fig. 3 Initial depth and flow. (**a**)–(**b**) Depth maps for two consecutive frames. (**c**) Close-ups of (**a**) and (**b**) in a top-down order. (**d**)–(**e**) Initial color-coded optical flow fields

Disparity Initialization We compute the disparity d_l^t for each frame t by optimizing

$$E_0(\mathrm{d}^t) = \int_{\Omega} E_F(\mathrm{d}^t) + \beta_\mathrm{d} E_{S_d}(\nabla \mathrm{d}^t)\mathrm{dx}, \qquad (2)$$

where x is defined on Ω, domain of x in the 2D image grid. $E_F(\mathrm{d}^t)$ is given in Eq. (1). $E_{S_d}(\nabla \mathrm{d}^t)$ is the truncated L_1 function to preserve discontinuity, written as $E_{S_d}(\nabla \mathrm{d}^t) := \min(|\nabla \mathrm{d}^t|, \rho)$, where $\rho = 3$. It is a regularization term to preserve edges. β_d is a weight. With the discrete energy function, we solve Eq. (2) using graph-cuts (Kolmogorov and Zabih 2004). Occlusion is further explicitly labeled using uniqueness check (Scharstein and Szeliski 2002). When two pixels in the left view are mapped to the same one in the right view, the pixel with smaller disparity is set as occlusion, with $o_\mathrm{d}(x) = 0$.

Disparity maps for two consecutive frames are shown in Fig. 3(a)–(b). Textureless regions, such as the mountain, and region boundaries have inconsistent estimates. Close-ups of the depth boundaries are shown in (c). The inconsistent depth values cause flickering. See our supplementary video for the depth sequence.[2]

Optical Flow Initialization We initialize 2D motion in a variational framework. Both the forward and backward flow vectors are computed, denoted as $\mathrm{u}_l^{t,t+1}$ and $\mathrm{u}_l^{t+1,t}$, for outlier rejection. For the following robust estimation, which is detailed in Sect. 4.2, we also compute bi-directional flow between frames f_l^t and f_l^{t+2} (denoted as $\mathrm{u}_l^{t,t+2}$ and $\mathrm{u}_l^{t+2,t}$), and between frames f_l^t and f_l^{t+3} (denoted as $\mathrm{u}_l^{t,t+3}$ and $\mathrm{u}_l^{t+3,t}$). Figure 4 illustrates these vectors. As all motion vectors are computed similarly, we only describe estimation of $\mathrm{u}_l^{t,t+1}$. It is achieved by minimizing

$$E_0(\mathrm{u}_l^{t,t+1}) = \int_{\Omega} E_L(\mathrm{u}_l^{t,t+1}) + \beta_\mathrm{u} E_{S_u}(\nabla \mathrm{u}_l^{t,t+1})\mathrm{dx}, \qquad (3)$$

where $E_{S_u}(\nabla \mathrm{u}_l^{t,t+1})$ is the total variation regularizer, expressed as $\sqrt{\|\nabla \mathrm{u}_l^{t,t+1}\|^2 + \epsilon^2}$ to preserve edges. It is a convex penalty function and is commonly used in the variational framework. β_u is a weight controlling the smoothness of the computed flow fields. Equation (3) is optimized by the efficient method of Brox et al. (2004). The initially estimated optical flow for two consecutive frames is shown in Fig. 3(d)–(e).

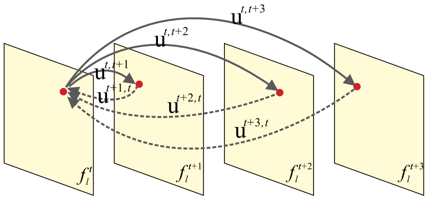


Fig. 4 Flow vector illustration

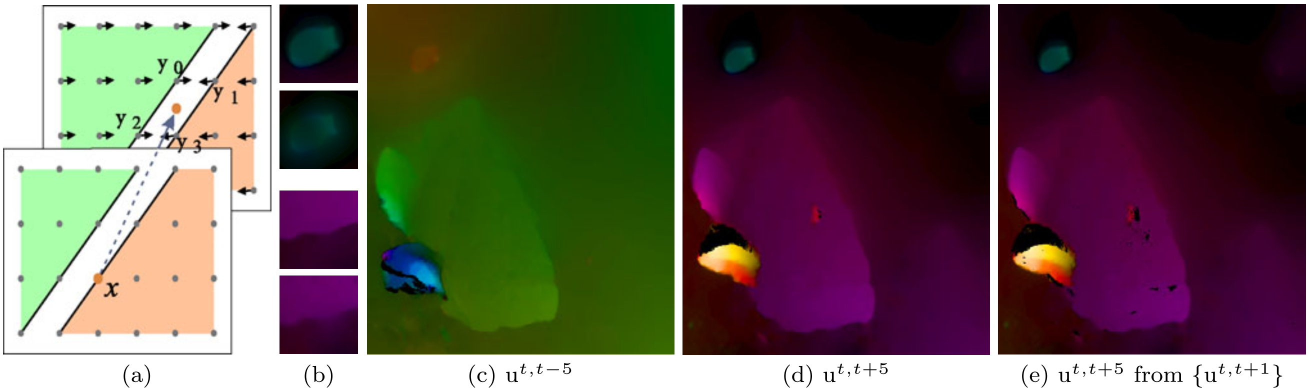
4.2 Chained Temporal Priors

A part of our contribution lies on constructing motion trajectories after initialization, which link corresponding pixels in several frames, and on proposing new structure profiles, essential in our system to form new temporal constraints.

Robust Motion Trajectory Estimation For each pixel, to find its correspondences in other frames, we build motion trajectories based on the optical flow estimate in each sequence. Note that motion vectors do not necessarily contain integers and thus may link sub-pixels. Specifically, $x + \mathrm{u}^{t,t+1}(x)$ in f^{t+1} that is mapped from x in f^t based on the motion vector $\mathrm{u}^{t,t+1}(x)$ is possibly a fractional value, locating in between four pixels, as shown in Fig. 5(a). When searching for the correspondence of $x + \mathrm{u}^{t,t+1}(x)$ in frame $t + 2$ or back in frame t, the motion vector for $x + \mathrm{u}^{t,t+1}(x)$ has to be estimated, generally by spatial interpolation.

Here we propose a simple and yet effective way to improve interpolation accuracy. We observe in our experiments that simple distance-based interpolation, e.g., bilinear or bicubic method, could produce erroneous results. Figure 5(a) shows one example that the orange and green regions undergo motion in different directions. A point projected in between pixels $\{y_0, y_1, y_2, y_3\}$, after bilinear inter-

[2] http://www.cse.cuhk.edu.hk/%7eleojia/projects/depth/.



(a) (b) (c) $\mathrm{u}^{t,t-5}$ (d) $\mathrm{u}^{t,t+5}$ (e) $\mathrm{u}^{t,t+5}$ from $\{\mathrm{u}^{t,t+1}\}$

Fig. 5 (**a**) Illustration of a flow interpolation issue. (**b**) Interpolation results on two patches. For each of them, bilateral and bilinear interpolation results are shown on top and bottom respectively. (**c**)–(**d**) $\mathrm{u}^{t,t-5}$ and $\mathrm{u}^{t,t+5}$ based on our trajectories. Occlusion is labeled as black. (**e**) $\mathrm{u}^{t,t+5}$ based on our trajectories. They are produced not considering long-range motion $\mathrm{u}_l^{t,t+2}(x)$ and $\mathrm{u}_l^{t,t+3}(x)$. Errors are larger than those in (**d**)

polation, is with near zero motion magnitude, which is obviously inappropriate. This problem is quite common for sequences containing dynamic objects. We ameliorate it by incorporating the color information to guide interpolation *bilaterally* together with the spatial distance, originated from spatial bilateral filtering (Tomasi and Manduchi 1998). The operator is written as

$$\mathrm{u}^{t+1,t'}\big(x+\mathrm{u}^{t,t+1}(x)\big)=\frac{1}{|w|}\sum_{i=0}^{3}\mathrm{u}^{t+1,t'}(y_i)\cdot e^{-(x+\mathrm{u}^{t,t+1}(x)-y_i)^2/\sigma_1-(f_I^t(x)-f_I^{t+1}(y_i))^2/\sigma_2}, \tag{4}$$

where t' can be t, $t+2$, or other frame indexes depending on the motion definition. $|w|$ is for normalization. The term $(f_I^t(x)-f_I^{t+1}(y_i))^2/\sigma_2$ considers the brightness similarity of points in different frames. σ_1 and σ_2 are set to 0.4 and 0.3 respectively. The comparison of bilateral interpolation and standard bilinear interpolation is given in Fig. 5(b). In the two motion field patches that contain dynamic object boundaries, it is clear that the bilateral method produces much sharper motion boundaries.

With this interpolation scheme, we link corresponding points among frames, which forms *motion trajectories*. Due to inevitable estimation errors in occlusion regions, object boundaries, and textureless regions, we identify and exclude outliers with bidirectional flow vectors (Huguet and Devernay 2007). In particular, we project $x+\mathrm{u}^{t,t+1}(x)$ in f^{t+1}, which is mapped from x in f^t based on the motion vector $\mathrm{u}^{t,t+1}(x)$, back to f^t. We sum the two vectors with opposite directions and define the map o_u, to mark glaring errors, as

$$o_\mathrm{u}(x)=\begin{cases}0 & |\mathrm{u}_o^{t,t+1}|\geq\tau\\ 1 & \text{otherwise}\end{cases} \tag{5}$$

where

$$\mathrm{u}_o^{t,t+1}=\mathrm{u}^{t+1,t}\big(x+\mathrm{u}^{t,t+1}(x)\big)+\mathrm{u}^{t,t+1}(x),$$

and τ is the error threshold set to 1. Satisfying the inequality $|\mathrm{u}_o^{t,t+1}|\geq\tau$ means the motion vectors that are supposedly opposite are malposed. We in this case discard $\mathrm{u}^{t,t+1}(x)$ and set $o_\mathrm{u}(x)$ to 0.

Removing a problematic flow vector shortens a motion trajectory. If it is too short, insufficient temporal information could be resulted in. With the observation that many outliers are caused by occasional noise and pixel color variation, which do not present consistently in frames for the same pixel in general, we also utilize longer-range bidirectional flow vectors, i.e., $\mathrm{u}_l^{t,t+2}(x)$ and $\mathrm{u}_l^{t+2,t}(x)$, as illustrated in Fig. 4. If they are valid after going through the same bidirectional consistency check, we reconnect the trajectory from frame t to $t+2$. Otherwise, we continue to test the pair of $\mathrm{u}_l^{t,t+3}(x)$ and $\mathrm{u}_l^{t+3,t}(x)$. Only if all these three checks fail, we break the trajectory. The algorithm to build a motion trajectory is detailed in Algorithm 2.

With these trajectories built in the forward and backward directions, we can find a series of correspondences in other frames for each pixel x^t in frame t. The motion vector w.r.t. x^t and the correspondence in frame $t\pm i$ is expressed as $\mathrm{u}^{t,t\pm i}$, which is the sum of all consecutive motion vectors in the trajectory from frame t to $t\pm i$ for x^t.

Figure 5(c)–(d) show respectively motion fields $\mathrm{u}^{t,t-5}$ and $\mathrm{u}^{t,t+5}$ produced by Algorithm 2. Unreliable matching is marked as black (NaN in Algorithm 2). They are typically caused by motion occlusion, leading to break of motion trajectories. This type of disconnection is however desirable because occlusion shortens trajectories by nature. Results in (c)–(d) also demonstrate that one pixel is occluded at most along one direction, but not both.

Algorithm 2 Robust Trajectory Building

```
INPUT: $\{\mathrm{u}^{t,t\pm1}\}$, $\{\mathrm{u}^{t,t\pm2}\}$, $\{\mathrm{u}^{t,t\pm3}\}$
for $i = 2$ to $n$ do
  for coordinate $x$ do
    if $o_{\mathrm{u}}(\mathrm{u}^{t\pm(i-1),t\pm i}, \mathrm{u}^{t\pm i,t\pm(i-1)}, x)$ then
      $\mathrm{u}^{t,t\pm i}(x) = \mathrm{u}^{t\pm(i-1),t\pm i}(x + \mathrm{u}^{t,t\pm(i-1)}(x)) + \mathrm{u}^{t,t\pm(i-1)}(x)$.
    else if $o_{\mathrm{u}}(\mathrm{u}^{t\pm(i-2),t\pm i}, \mathrm{u}^{t\pm i,t\pm(i-2)}, x)$ then
      $\mathrm{u}^{t,t\pm i}(x) = \mathrm{u}^{t\pm(i-2),t\pm i}(x + \mathrm{u}^{t,t\pm(i-2)}(x)) + \mathrm{u}^{t,t\pm(i-2)}(x)$.
    else if $o_{\mathrm{u}}(\mathrm{u}^{t\pm(i-3),t\pm i}, \mathrm{u}^{t\pm i,t\pm(i-3)}, x)$ and $i > 2$ then
      $\mathrm{u}^{t,t\pm i}(x) = \mathrm{u}^{t\pm(i-3),t\pm i}(x + \mathrm{u}^{t,t\pm(i-3)}(x)) + \mathrm{u}^{t,t\pm(i-3)}(x)$.
    else
      $\mathrm{u}^{t,t\pm i}(x) = \mathrm{NaN}$.
    end if
  end for
end for
OUTPUT: $\{\mathrm{u}^{t,t\pm l}\}$, $l \in \{1, 2, \ldots, n\}$
```

Note: NaN represents invalid motion vectors.

To demonstrate the effectiveness of longer-range bidirectional flow $\mathrm{u}_l^{t,t+2}(x)$ and $\mathrm{u}_l^{t,t+3}(x)$ in generating trajectories, we show a comparison in Fig. 5(d) and (e), where (e) is produced with trajectory construction not using long-range flow and only relying on $\mathrm{u}_l^{t,t+1}$. Less black pixels are in the flow field (d) due to higher robustness against occasional noise and estimation outliers.

Trajectory-Based Structure Profile To build a practical system with constraints applied across multiple frames, besides trajectories, we also base our computation on a central observation—that is, motion and disparity boundaries are mostly in line with image structure boundaries. Salient edge maps, in this regard, are useful in shaping motion and disparity.

Unfortunately, depth and flow edges are very sensitive to noise, blurriness, illumination variance, and other kinds of image degradation. When taking sequences into consideration, boundaries of dynamic objects vary over time and are usually composed of different sets of pixels. It is common knowledge that only using image gradient information for frames separately can hardly infer reliable and consistent edges, as illustrated in Fig. 6(a).

Our goal in this part is to compute a series of salient structures that are temporally consistent and spatially conspicuous, regardless whether they are on dynamic objects or along illuminance variation boundaries. To this end, we first calculate edge magnitude maps separately for each frame after bilateral smoothing to remove a small degree of noise. The magnitude operator is $\sqrt{f_{\partial h}^2 + f_{\partial v}^2}$. Then we compute an *edge occurrence* map C_f^t by simply setting pixels with their magnitudes smaller than a threshold (generally set to 0.01) to zeros. It actually indicates the occurrence of significant structure edges. All maps in the input sequence, together with the computed dense motion trajectories, are used to establish a structure profile map for each frame.

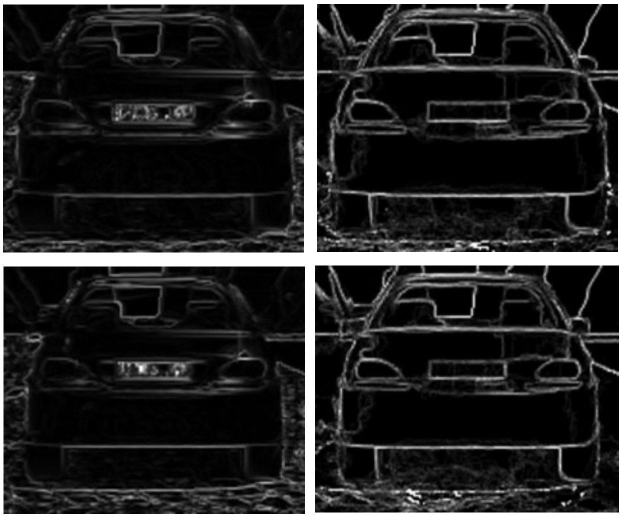

(a) $\sqrt{f_{\partial h}^2 + f_{\partial v}^2}$ (b) $\tilde{C}_f^t$

Fig. 6 Illustration of structure profiles. (**a**) Single-image edge extraction. The edges are weak and inconsistent in the two frames. (**b**) Our structure profile that is temporally more consistent

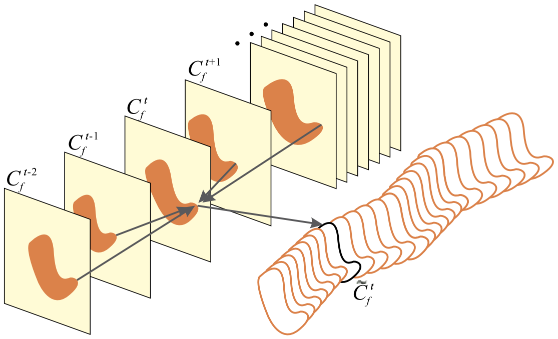


Fig. 7 Trajectory-based structure profile construction

For each pixel x in frame t, we project all *edge occurrence* values in other frames, according to the correspondences along the trajectories $\{\mathrm{u}^{t,t+i}\}$ and $\{\mathrm{u}^{t,t-i}\}$, to it, as illustrated in Fig. 7. In this process, we can find consistent edge occurrence values where errors, after a voting-like process, can be quickly suppressed.

The corresponding point of x in frame $t + i$ is $x + \mathrm{u}^{t,t+i}(x)$ after chain projection, where $\mathrm{u}^{t,t+i}(x)$ is the overall motion vector. The average of the occurrence value in the trajectory is expressed as

$$\tilde{C}_f^t(x) = \frac{1}{n}\sum_i C_f^{t+i}\big(x + \mathrm{u}^{t,t+i}(x)\big), \tag{6}$$

where n is the number of corresponding pixels along the trajectory. The structure profiles $\tilde{C}^t_f(x)$ embody statistics of the occurrence of strong edges over multiple frames. Its value reveals the chance that the current pixel is on a consistent edge.

We do not introduce weights in Eq. (6) because true edges can typically exist in a large number of frames consistently while a false one caused by noise and estimation errors does not. By projecting all correspondences to the current pixel and adding their occurrence values, a true edge point can gather a large confidence value. Occasional outliers cannot receive consistent support temporally, and therefore only have small confidence. In this voting-like process, originally weak but consistent edges can be properly enhanced.

The resulting edge occurrence maps are $\{\tilde{C}^t_f\}$, which are used to define edge priors. Figure 6(b) shows two edge occurrence maps, where inconsistent edges are notably weakened and consistent ones are enhanced. This profile construction process is robust against noise and sudden illumination change.

Trajectory-Based Depth/Motion Profile Another set of important profiles are constructed based on the fact that $\tilde{C}_f$ does not contain scene flow information. Spatial-temporal constraints were proposed in Black (1994), Bruhn and Weickert (2005), Bruhn et al. (2005) to enforce temporal consistency of motion vectors. Locally constant speed (Bruhn and Weickert 2005; Bruhn et al. 2005) and acceleration (Black 1994) are typical assumptions. We use a temporal-linear model to fit motion and depth, in order to reject outliers while allowing for depth and motion variation.

Here, we describe our depth profile estimation procedure. Motion profile can be computed similarly. For each pixel, we adopt a linear parametric model to fit depth after projecting values from other frames to t temporally based on our trajectories. The linear model is controlled by two parameters w_0 and w_1, representing depth offset and slope. Regression needs to minimize the energy

$$\sum_i \gamma_i(x)\left(w_1(x)i + w_0(x) - \frac{1}{\mathrm{d}^{t+i}}\right)^2, \tag{7}$$

where i indexes frames in the trajectory and $\gamma_i(x)$ is the weight for the $(t+i)$th frame and $1/\mathrm{d}^{t+i}$ is the corresponding depth for x in frame $t+i$. With sub-pixel point position, we interpolate depth using bilateral weights described in Eq. (4). $\gamma_i(x)$ plays an important role and is defined as

$$\gamma_i(x) = e^{-i^2/\sigma_t} \cdot o_{\mathrm{d}}(x),$$

which embodies two parts. They are respectively temporal weight e^{-i^2/σ_t} in a Gaussian window to reduce the influence of frames far away from the current frame t with $\sigma_t = 10$, and depth occlusion $o_{\mathrm{d}}(x)$, which is labeled using the uniqueness check (Scharstein and Szeliski 2002). Zero $o_{\mathrm{d}}(x)$ indicates occlusion, which correspondingly decreases the weight $\gamma_i(x)$ to zero.

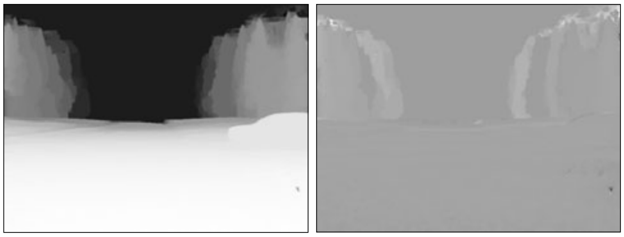

(a) w_0 (b) w_1

Fig. 8 Computed w_0 and w_1 maps for one frame

Before regression in Eq. (7), we check the sum $\sum_i \gamma_i$. If $\sum_i \gamma_i < 3$, we skip regression for the robustness' sake. Equation (7) provides an effective way to gather statistical disparity information from multiple frames without much occlusion influence. As the initial disparity values are optimized in a global fashion in each frame, the robust regression process actually has incorporated neighboring disparity information for each pixel.

In minimization, taking derivatives w.r.t. the parameters $w_0(x)$ and $w_1(x)$ and setting them to zeros yield two equations. After a few simple algebraic operations, closed-form solutions are obtained as

$$w_0(x) = \frac{\sum_i \gamma_i(x)i \cdot \sum_i \frac{\gamma_i(x)i}{\mathrm{d}^{t+i}} - \sum_i \gamma_i(x)i^2 \sum_i \frac{\gamma_i(x)}{\mathrm{d}^{t+i}}}{(\sum_i \gamma_i(x)i)^2 - (\sum_i \gamma_i(x)i^2 \cdot \sum_i \gamma_i(x))}, \tag{8}$$

$$w_1(x) = \frac{\sum_i \gamma_i(x)i \cdot \sum_i \frac{\gamma_i(x)}{\mathrm{d}^{t+i}} - \sum_i \gamma_i(x) \sum_i \frac{\gamma_i(x)i}{\mathrm{d}^{t+i}}}{(\sum_i \gamma_i(x)i)^2 - (\sum_i \gamma_i(x)i^2 \cdot \sum_i \gamma_i(x))}. \tag{9}$$

Finally, given the estimated linear parameters, the depth profile $\tilde{\mathrm{d}}^t(x)$ for x in frame t is written as

$$\tilde{\mathrm{d}}^t(x) = \frac{1}{w_0(x)}. \tag{10}$$

We show the w_0 and w_1 results in Fig. 8. w_0 corresponds to scene depth. It is made temporally more consistent after optimization thanks to the regression to reject random outliers and preserve boundaries. The average magnitude of the w_1 map is much smaller than that of w_0, manifesting that depth does not undergo abrupt change over time.

For motion profile computation, we similarly apply the linear model, yielding

$$\tilde{\mathrm{u}}^t(x) = \frac{\sum_i \gamma_i(x)i \sum \gamma_i(x)i\mathrm{u}^{t+i} - \sum_i \gamma_i(x)i^2 \sum_i \gamma_i(x)\mathrm{u}^{t+i}}{(\sum_i \gamma_i(x)i)^2 - (\sum_i \gamma_i(x)i^2 \cdot \sum_i \gamma_i(x))}, \tag{11}$$

where $\tilde{\mathrm{u}}^t(x)$ is the motion profile for pixel x in frame t and

$$\gamma_i(x) = e^{-i^2/\sigma_t} \cdot o_{\mathrm{u}}(x).$$

Algorithm 3 Depth and Scene Flow Computation

```
INPUT: depth profile $\{\tilde{u}^{t,t+1}\}$, motion profile $\{\tilde{d}^t\}$, sequences $\{f^0_{l,r}, f^1_{l,r}, \cdot, f^{N-1}_{l,r}\}$
Update depth $\{\tilde{d}^t\}$ to $\{d^t\}$ for all frames with the temporal constraint (Sect. 4.3.1).
for frame $i = 0$ to $N-1$ do
    Optimize scene flow $s^{i,i+1}$ based on $(d^i, d^{i+1}, \tilde{u}^{i,i+1})$ (Sect. 4.3.2).
end for
OUTPUT: temporally consistent disparity maps $\{d^t\}$ and scene flow $\{s^{t,t+1}\}$
```

o_u is motion occlusion estimated through bidirectional check. As shown in Fig. 5(c)–(d), occlusion generally arises along one direction—that is, either forward or backward—but not both. So the situation that $o_u(x) = 0$ for all correspondences of one pixel in the trajectory seldom occurs, making it always possible to find points to gather statistics and refine motion during regression. We adopt a small σ_t (set to 3) due to the fact that motion variation is typically larger than the change of scene depth.

4.3 Temporally-Constrained Depth and Scene Flow

In this section, we describe the central steps to estimate depth and image scene flow given the temporal constraints. We find that estimating depth and scene flow in the same pass is computationally expensive (Wedel et al. 2011) and unstable due to the fact that depth and scene flow are completely different variables by nature. Disparity can have very large values (up to tens or hundreds in the pixel scale) while scene flow captures object position variation and thus has much smaller scales. Putting them together makes variational optimization difficult to perform satisfyingly and be easily stuck in local optima. As reported in Huguet and Devernay (2007), Wedel et al. (2011), a full joint procedure to estimate depth and scene flow takes pretty long time.

It is also notable that the initial disparity values estimated frame-by-frame are lack of sub-pixel accuracy, unsuitable for $\delta d^{t,t+1}$ estimation in scene flow. With these concerns, we decouple depth and scene flow, and optimize the disparity sequence with the long-range temporal constraint. Scene flow is then updated. The algorithm is outlined in Algorithm 3. We describe below the spatio-temporal functions to constrain depth and scene flow.

4.3.1 Consistent Depth Estimation

To refine depth, we minimize

$$E_f(d^t) = \int_\Omega \hat{o}_d(x) E_F(d^t) + \alpha_d E_T(d^t) + \beta_d E_S(\nabla d^t)\,dx, \tag{12}$$

where α_d and β_d are two weights. $E_F(d^t)$ is the data cost defined in Eq. (1), which relates two views at time t. The occlusion variable $\hat{o}_d(x)$ helps reduce the adverse influence of occasional occlusion. We define $\hat{o}_d(x) = \max(o_d(x), 0.01)$, for the sake of numerical stability. $E_T(d^t)$ is the *temporal* depth data term, defined as

$$E_T(d^t) = (d^t - \tilde{d}^t)^2, \tag{13}$$

where $\tilde{d}^t$ is the fitted depth profile. This seemingly simple term is essential in our method because it incorporates long-range temporal information from multiple frames. E_F, on the contrary, is only a local frame-wise data term.

The structure profile is incorporated in depth regularization to enforce structure consistency among frames. We only impose smoothness for regions with small edge-occurrence values in $\tilde{C}^t_f$ and allow depth discontinuity to take place when $\tilde{C}^t_f(x)$ is large. On account of possible subpixel errors in averaging the edge-occurrence maps, edges in $\tilde{C}^t_f$ could be slightly wider than what they should be, as shown in Fig. 6. It is inappropriate to naively enforce no or small smoothness for these pixels because, without necessary regularization, edge cannot be well preserved.

We turn to an anisotropic smoothness method (Xiao et al. 2006; Sun et al. 2008; Zimmer et al. 2009) to provide critical constraints for edge-preserving regularization. We decompose depth gradient ∇d into $\{\nabla d^\parallel, \nabla d^\perp\}$ according to the image gradient, where

$$\nabla d^\parallel = \langle \nabla d, \nabla f_I^\parallel \rangle \cdot \nabla f_I^\parallel, \nabla d^\perp = \nabla d - \nabla d^\parallel.$$

where $\nabla f_I^\parallel$ is the unit vector parallel to the frame intensity gradient ∇f_I, i.e. $\nabla f_I^\parallel = \frac{\nabla f_I}{\|\nabla f_I\|}$. We propose the following function for anisotropic smoothness regularization:

$$E_S(\nabla d) = \Gamma\big((\nabla d^\perp)^2 + (1 - \tilde{C}_f)(\nabla d^\parallel)^2\big), \tag{14}$$

where $\Gamma(\cdot)$ is the robust Charbonnier function, defined in Eq. (1). In Eq. (14), for all pixels, smoothness is enforced along the isophote direction while discontinuity along gradient is allowed only for reliable strong edges, which corresponds to large $\tilde{C}_f$. Hence, Eq. (14) provides necessary constraints in different directions. We note here that strong edges caused by occasional outliers in one frame would not affect the anisotropic regularizer. On the other hand, compared with C_f, temporally consistent edges have been enhanced in $\tilde{C}_f$, which boost the anisotropic properties accordingly.

With a few algebraic operations, Eq. (14) can be written in a form of diffusion tensor

$$E_S(\nabla d) = \Gamma\big(\nabla d^T D(\nabla f_I) \nabla d\big), \tag{15}$$

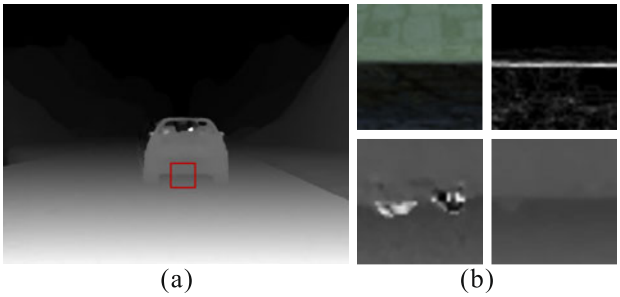


Fig. 9 Effectiveness of anisotropic regularization. (**a**) Our depth estimate $\{\mathrm{d}^t\}$. (**b**) Top to bottom and left to right: patch of input image f^t, $\tilde{C}^t_f$, and depth results using the isotropic and anisotropic regularization terms, both guided by $\tilde{C}^t_f$

where $D(\nabla f_I)$ is the diffusion tensor defined as

$$D(\nabla f_I) = \left((\nabla f_I^{\perp})(\nabla f_I^{\perp})^T + (1-\tilde{C}_f)(\nabla f_I^{\|})(\nabla f_I^{\|})^T\right),$$

where $\nabla f_I^{\|}$ and $\nabla f_I^{\perp}$ are two unit vectors parallel and perpendicular to ∇f_I, respectively. Equation (12) can be efficiently minimized using variational solvers to enable subpixel accuracy. Note that the inherent difficulty to solve for large displacements in the variational framework is greatly reduced with our initial estimate $\tilde{\mathrm{d}}^t$, obtained by robust regression. Our energy minimization is discussed in Sect. 4.3.3.

A depth map result is shown in Fig. 9(a), with the comparison in (b). The bottom left subfigure of (b) is the depth map obtained by enforcing smoothness uniformly in all directions with strength $(1-\tilde{C}^t_f)$. When $\tilde{C}^t_f$ is large near the boundaries, the depth estimation is ill-posed, resulting in a problematic map. Our result with anisotropic regularization preserves much better edges.

4.3.2 Scene Flow Estimation

We now estimate 2D motion and depth variation with necessary temporal constraints for image scene flow.

Data Fidelity Term Our new data cost is given by

$$\begin{aligned} &E_D(\mathrm{u}^{t,t+1}, \delta\mathrm{d}^t) \\ &\quad = \hat{o}_d\hat{o}_u E_B(\mathrm{u}^{t,t+1}, \delta\mathrm{d}^{t,t+1}) \\ &\qquad + \left(\hat{o}_u E_L(\mathrm{u}^{t,t+1}) + \hat{o}_d\hat{o}_u E_R(\mathrm{u}^{t,t+1}, \delta\mathrm{d}^{t,t+1})\right) \\ &\qquad + \alpha_{\mathrm{u}}\hat{o}_u E_{Td}(\delta\mathrm{d}^t_l) + \alpha_{\mathrm{u}} E_{Tu}(\mathrm{u}^{t,t+1}), \end{aligned} \tag{16}$$

where E_B, E_L and E_R are the traditional scene flow constraints, defined in Eq. (1). $\{\mathrm{d}^t\}$ is the disparity computed in the above estimation step (described in Sect. 4.3.1). $\hat{o}_u$, where $\hat{o}_u = \max(o_u, 0.01)$, and $\hat{o}_d$ mask out unreliable depth and flow primarily caused by occlusion. α_{u} is the weight for the long-range temporal constraint.

E_{Td} and E_{Tu} incorporate our new temporal profiles. E_{Td} is defined as

$$E_{Td}(\delta\mathrm{d}^{t,t+1}) = \left(\delta\mathrm{d}^{t,t+1} + \mathrm{d}^t - \mathrm{d}'^{t+1}\right)^2, \tag{17}$$

where d^t is a shorthand for $\mathrm{d}^t(x)$ and $\mathrm{d}'^{t+1} := \mathrm{d}^{t+1}(x + \mathrm{u}^{t,t+1}(x))$. They are disparities of x in the t-th and $(t+1)$-th frames, respectively. Our depth refinement makes $\delta\mathrm{d}^{t,t+1}$ a sub-pixel value.

Similarly, the flow constraint is defined as

$$E_{Tu}(\mathrm{u}^{t,t+1}) = \left(\mathrm{u}^{t,t+1} - \tilde{\mathrm{u}}^{t,t+1}\right)^2. \tag{18}$$

Smoothness Term In regularization, we penalize sudden and significant change of the 2D motion $\nabla\mathrm{u}^{t,t+1}$ and of the disparity $\nabla\delta\mathrm{d}^{t,t+1}$ w.r.t. the frame diffusion tensor, yielding an anisotropic smoothing effect with the edge maps as guidance. We define

$$\begin{aligned} &E_S(\nabla\mathrm{u}^{t,t+1}, \nabla\delta\mathrm{d}^{t,t+1}) \\ &\quad = \Gamma\left(\nabla\mathrm{u}^{t,t+1\,T} D(\nabla f_I)\nabla\mathrm{u}^{t,t+1}\right) \\ &\qquad + \kappa\Gamma\left(\nabla\delta\mathrm{d}^{t,t+1\,T} D(\nabla f_I)\nabla\delta\mathrm{d}^{t,t+1}\right). \end{aligned} \tag{19}$$

Here κ is a weight set to 0.5. The final objective function for scene flow estimation with temporal constraints is given by

$$\begin{aligned} &E_f(\mathrm{u}^{t,t+1}, \delta\mathrm{d}^{t,t+1}) \\ &\quad = \int_\Omega E_D(\mathrm{u}^{t,t+1}, \delta\mathrm{d}^{t,t+1}) \\ &\qquad + \beta_{\mathrm{u}} E_S(\nabla\mathrm{u}^{t,t+1}, \nabla\delta\mathrm{d}^{t,t+1})\mathrm{d}x. \end{aligned} \tag{20}$$

We minimize it using the variational method detailed below.

4.3.3 Energy Minimization

Equations (12) and (20) can be solved in a coarse-to-fine framework using variational solvers. However, this procedure is found not necessary. It is because in our system, variables are well initialized and estimated before optimization in the two stages. To compute depth using Eq. (12), the temporal depth profile is available. While jointly optimizing elements in scene flow using Eq. (20), the updated depth and motion profiles provide good initialization. Moreover, energy minimization with proper initialization ameliorates the inherent estimation problems for large displacements (Brox et al. 2009; Xu et al. 2010), even in the original image resolution.

With this consideration, we perform Taylor series expansion on the data term and solve for the increments

$$\Delta\mathrm{d} = \mathrm{d} - \mathrm{d}^{(0)}, \qquad \Delta s = s - s^{(0)}.$$

$\mathrm{d}^{(0)}$ is set to $\tilde{\mathrm{d}}$ in the first place. After $\Delta\mathrm{d}$ is computed, $\mathrm{d}^{(0)}$ is accordingly updated to $\mathrm{d}^{(0)} + \Delta\mathrm{d}$. Then we estimate $\Delta\mathrm{d}$ again. We repeat this procedure for 3 times. The final depth map is computed by $\mathrm{d} = \mathrm{d}^{(0)} + \Delta\mathrm{d}$. Given the refined depth map, we set $s^{(0)} = (\tilde{u}, \tilde{v}, \mathrm{d}^{t+1}(x + \tilde{u}) - \mathrm{d}^t)$ and iteratively solve for the increment Δs.

This scheme is referred to as the warping strategy in Brox et al. (2004). In our system, to estimate $\Delta\mathrm{d}$ and Δs, the overall energy functions are minimized by solving the their corresponding Euler-Lagrange equations. To make description easy, we denote

$$\begin{aligned}
f_{dh} &:= \partial_h f_r^t\big(x + \mathrm{d}^{(0)}\big),\\
f_{dz} &:= f_r^t\big(x + \mathrm{d}^{(0)}\big) - f_l^t(x),\\
\varepsilon_d &:= f_{dh} \cdot \Delta\mathrm{d} + f_{dz}.
\end{aligned}$$

In addition, $\Gamma'(y^2) := 1/\sqrt{y^2 + \epsilon^2}$, representing derivative of the Γ function. The Euler-Lagrange equation for Eq. (12) is given by

$$\begin{aligned}
0 = {}& \hat{o}_d \sum_k \Gamma'\big(\big(\varepsilon_d^{[k]}\big)^2\big) \cdot \varepsilon_d^{[k]} f_{dh}^{[k]} + 2\alpha_{\mathrm{d}} \Delta\mathrm{d}\\
&-\beta_{\mathrm{d}} \mathrm{div}\big(\Gamma'\big(\nabla\mathrm{d}^T D(\nabla f_I)\nabla\mathrm{d}\big) \cdot D(\nabla\mathrm{d})\nabla\mathrm{d}\big), \qquad (21)
\end{aligned}$$

where $\mathrm{div}(\cdot)$ is the divergence operator. After discretizing the equation, nonlinearity is only held in Γ'. A fixed-point loop similar to that in Brox et al. (2004), Bruhn and Weickert (2005) is applied, which removes the nonlinearity of Γ' by using values obtained from the previous iteration. This step yields linear equations, which can be quickly solved by standard linear solvers. More details are included in Appendix A.

Similarly, for final scene flow computation, we denote

$$\begin{aligned}
f_{lh} &:= \partial_h f_l^{t+1}\big(x + \mathrm{u}^{(0)}(x)\big)\\
f_{lv} &:= \partial_v f_l^{t+1}\big(x + \mathrm{u}^{(0)}(x)\big)\\
f_{lz} &:= f_l^{t+1}\big(x + \mathrm{u}^{(0)}(x)\big) - f_l^t(x)\\
\varepsilon_L &:= f_{lh} \cdot \Delta u + f_{lv} \cdot \Delta v + f_{lz}\\
f_{rh} &:= \partial_h f^{t+1}\big(x + \mathrm{d} + \mathrm{u}^{(0)} + \delta\mathrm{d}^{(0)}\big)\\
f_{rv} &:= \partial_v f^{t+1}\big(x + \mathrm{d} + \mathrm{u}^{(0)} + \delta\mathrm{d}^{(0)}\big)\\
f_{rz} &:= f^{t+1}\big(x + \mathrm{d} + \mathrm{u}^{(0)} + \delta\mathrm{d}^{(0)}\big) - f_r^t(x + \mathrm{d})\\
\varepsilon_R &:= f_{rh} \cdot (\Delta u + \Delta\delta\mathrm{d}) + f_{rv} \cdot \Delta v + f_{rz}.
\end{aligned}$$

The Euler-Lagrange equations for Eq. (20) are given by

$$\begin{aligned}
&\hat{o}_d \hat{o}_u \sum_k \Gamma'\big(\big(\varepsilon_R^{[k]} - \varepsilon_L^{[k]}\big)^2\big) \cdot \big(\varepsilon_R^{[k]} - \varepsilon_L^{[k]}\big)(f_{rh} - f_{lh})\\
&\quad + \hat{o}_u \Gamma'\big(\big(\varepsilon_L^{[k]}\big)^2\big)\varepsilon_L^{[k]} f_{lh} + \hat{o}_d \hat{o}_u \Gamma'\big(\big(\varepsilon_R^{[k]}\big)^2\big)\varepsilon_R^{[k]} f_{rh}\\
&\quad + 2\alpha_{\mathrm{u}} \Delta u - \beta_{\mathrm{u}} \mathrm{div}\big(\Gamma'_{su} \cdot D(\nabla f_I)\nabla u\big) = 0 \qquad (22)\\
&\hat{o}_d \hat{o}_u \sum_k \Gamma'\big(\big(\varepsilon_R^{[k]} - \varepsilon_L^{[k]}\big)^2\big) \cdot \big(\varepsilon_R^{[k]} - \varepsilon_L^{[k]}\big)(f_{rv} - f_{lv})\\
&\quad + \hat{o}_u \Gamma'\big(\big(\varepsilon_L^{[k]}\big)^2\big)\varepsilon_L^{[k]} f_{lv} + \hat{o}_d \hat{o}_u \Gamma'\big(\big(\varepsilon_R^{[k]}\big)^2\big)\varepsilon_R^{[k]} f_{rv}\\
&\quad + 2\alpha_{\mathrm{u}} \Delta v - \beta_{\mathrm{u}} \mathrm{div}\big(\Gamma'_{su} \cdot D(\nabla f_I)\nabla v\big) = 0 \qquad (23)\\
&\hat{o}_d \hat{o}_u \sum_k \Gamma'\big(\big(\varepsilon_R^{[k]} - \varepsilon_L^{[k]}\big)^2\big) \cdot \big(\varepsilon_R^{[k]} - \varepsilon_L^{[k]}\big)(f_{rh})\\
&\quad + \hat{o}_d \hat{o}_u \Gamma'\big(\big(\varepsilon_R^{[k]}\big)^2\big)\varepsilon_R^{[k]} f_{rh}\\
&\quad + 2\hat{o}_u \alpha_{\mathrm{u}} \Delta\delta\mathrm{d} - \beta_{\mathrm{u}} \mathrm{div}\big(\Gamma'_{sd} \cdot D(\nabla f_I)\nabla\delta\mathrm{d}\big) = 0 \qquad (24)
\end{aligned}$$

where

$$\Gamma'_{su} := \Gamma'\big(\nabla\mathrm{u}^T D(\nabla f_I)\nabla\mathrm{u} + \nabla\mathrm{u}^T D(\nabla f_I)\nabla\mathrm{u}\big), \qquad (25)$$

representing derivatives of the smoothness penalty on 2D motion. Γ'_{sd} is expressed as $\Gamma'(\nabla\delta\mathrm{d}^T D(\nabla f_I)\nabla\delta\mathrm{d})$, a penalty on depth variation. Applying a fixed-point conversion, the system becomes linear w.r.t. Δs and thus can be solved easily. Implementation details are in Appendix B.

5 Experiments

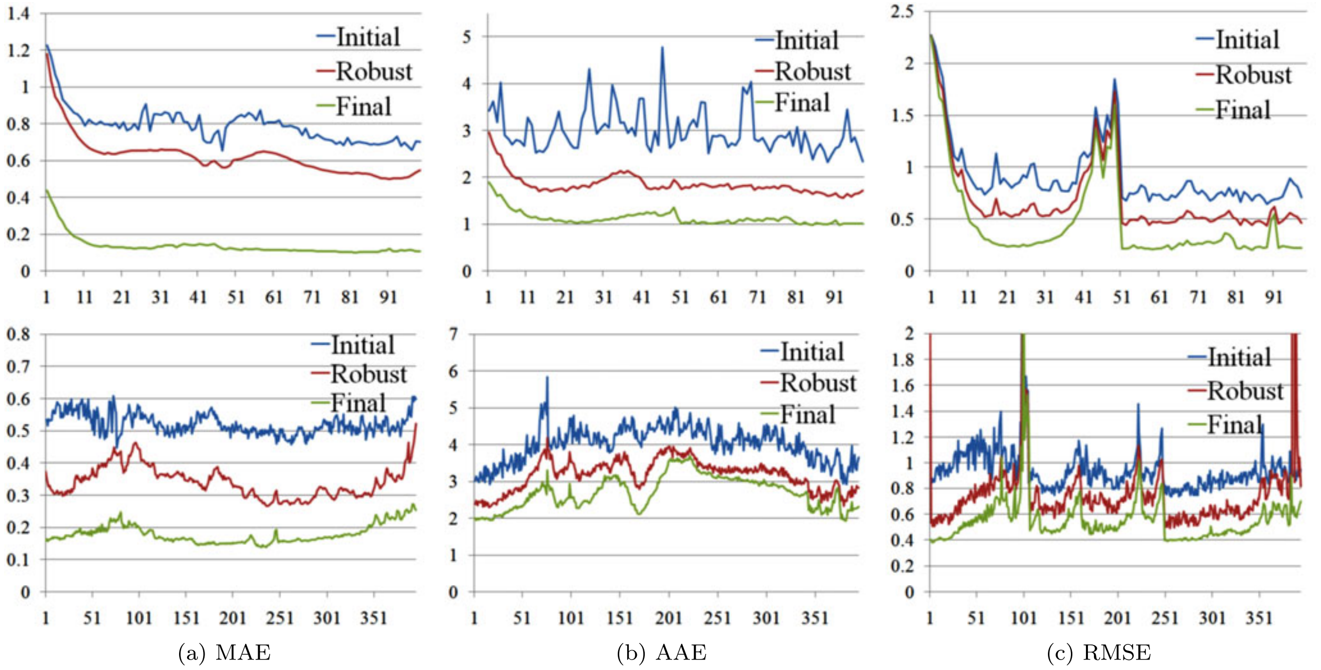
We extensively evaluate our method. We first perform quantitative evaluation on the POV-Ray-rendered sequences (Vaudrey et al. 2008; University of Auckland 2008), where ground truth depth and scene flow fields are available. It is notable that very few methods in the literature reported error statistics on video sequences because most of them used the four-frame configuration without leveraging long-range temporal information. In contrast, we provide both per-sequence statistics and detailed per-frame errors for each step of the proposed method. Complete results are presented in the supplementary video. To evaluate the long-range temporal constraints, we also conduct experiments to separately test important components, including the structure, long-range depth and motion profiles. Finally, we show results on challenging sequences containing large dynamic objects.

In dealing with synthetic sequences, β_{d} and β_{u} are set to 10 and 15, respectively. α_{d} and α_{u} are both with value 10. For natural sequences, we use $\beta_{\mathrm{d}} = 15$, $\beta_{\mathrm{u}} = 20$, $\alpha_{\mathrm{d}} = 15$, and $\alpha_{\mathrm{u}} = 5$. All other parameters are fixed as specified in the paper.

5.1 Quantitative Evaluation

We employ the mean absolute error (MAE) to measure the errors between our estimate d and the ground truth disparity d*:

$$\mathrm{MAE}_{\mathrm{d}} = \frac{1}{|\Omega|} \sum_{\Omega} \big|\mathrm{d} - \mathrm{d}^*\big|, \qquad (26)$$



(a) MAE (b) AAE (c) RMSE

Fig. 10 Per-frame error statistics for sequences *Traffic Scene 1* (*1st row*) and *Traffic Scene 2* (*2nd row*)

where $|\Omega|$ is the number of pixels in image Ω.

For scene flow, we adopt the average angular error (AAE) and the root mean square error (RMSE) (Wedel et al. 2011) to evaluate the angular and end-point errors:

$$\mathrm{AAE}_{3D} = \frac{1}{|\Omega|}\sum_{\Omega}\cos^{-1} \times \left(\frac{uu^* + vv^* + \delta d\delta d^* + 1}{\sqrt{(u^2+v^2+\delta d^2+1)}\sqrt{(u^{*2}+v^{*2}+\delta d^{*2}+1)}}\right), \tag{27}$$

$$\mathrm{RMSE}_{3D} = \left[\frac{1}{|\Omega|}\sum_{\Omega}\left\|(u,v,\delta d)^{\top} - (u^*,v^*,\delta d^*)^{\top}\right\|^2\right]^{\frac{1}{2}}. \tag{28}$$

The RMSE measures pixel-level errors while the AAE is a measure in degree. For fair comparison, we exclude image borders, occluded regions, and the infinite background, as suggested in Wedel et al. (2011).

Traffic Scene 1 is a sequence with 100 frames (University of Auckland 2008), which contains rapidly moving cars. Two frames are shown in Fig. 11(a)–(b). The whole sequence is provided in the supplementary video. The initial disparity map (d) is erroneous at occluded regions. In the robust regression result (e), we can notice that depth at occlusion is refined and the estimation outliers are reduced owing to the use of temporal information. Our final variational refinement further increases sub-pixel accuracy and lowers MAE down to 0.1 pixel. The absolute error is visualized in (g)–(i). The gray pixels with value 128 are with no error and brighter pixels indicate larger positive errors, following the representation in Wedel et al. (2011). The 3D scene flow fields in different stages are shown in (j)–(l), with the angular error maps coded in (m)–(p). Note that the rear window of the car and the dark shadow regions are consistently with large errors, due to transparency and large color variation.

Table 1 Average AAE, RMSE, and MAE obtained at different stages, including initialization ("Initial"), robust regression ("Robust"), and final refinement ("Final")

Stage	Scene flow		Disparity
	AAE	RMSE	MAE
(a) Errors for *Traffic Scene 1*			
Initial	2.989	0.925	0.789
Robust	1.848	0.697	0.624
Final	1.130	0.468	0.136
(b) Errors for *Traffic Scene 2*			
Initial	4.021	0.946	0.522
Robust	3.180	0.806	0.341
Final	2.701	0.557	0.179

The corresponding statistics for the whole sequence are listed in Table 1(a), where the errors produced in different stages are shown. They indicate that the two main steps make a great improvement. Detailed per-frame error plots

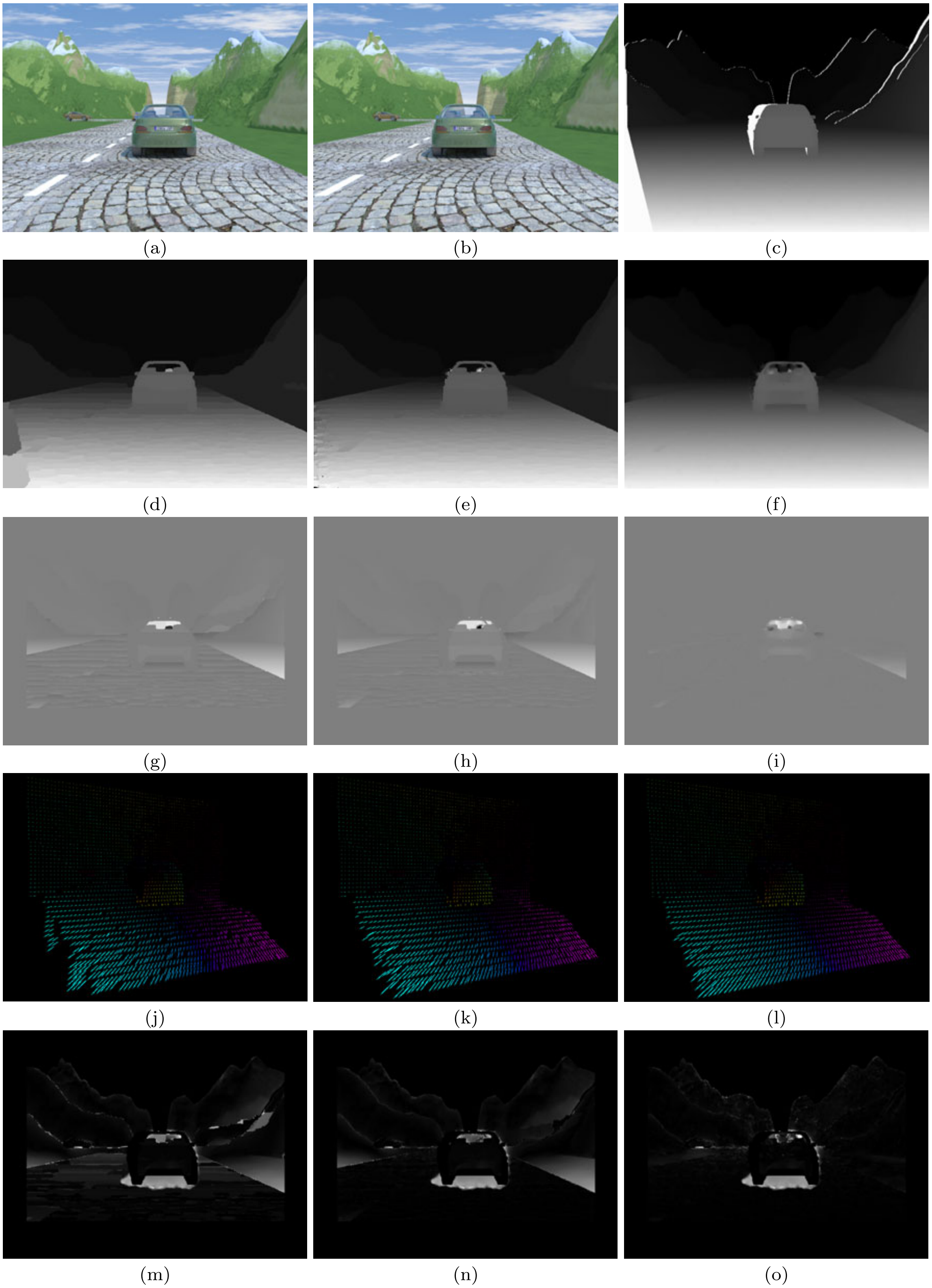


Fig. 11 *Traffic Scene 1* sequence. (**a**) and (**b**) are the left and right views respectively in time t. (**c**) is the ground truth disparity map, where white pixels are occlusion. (**d**)–(**f**) are disparity maps obtained in initialization, robust regression, and in the final refinement, respectively. (**g**)–(**i**) are the error images for (**d**)–(**f**) respectively, where dark to bright pixels are with negative to positive errors. (**j**)–(**l**) visualize the scene flow fields corresponding to (**g**)–(**i**). (**m**)–(**o**) show the angular error maps of scene flow

Table 2 Multi-pass depth and scene flow estimation errors

Pass	Scene flow		Disparity
	AAE	RMSE	MAE
1 pass	1.130	0.468	0.136
2 passes	0.922	0.422	0.101
3 passes	0.905	0.399	0.102

Table 3 Error comparison with different structure profiles

Strategies	Scene flow		Disparity
	AAE	RMSE	MAE
Without structure map	2.306	0.721	0.405
Single-frame structure map	1.310	0.519	0.273
Temporal structure map	1.130	0.468	0.136

are in the first row of Fig. 10. Our results get consistent improvement for all frames in each stage.

The results of *Traffic Scene 2* are shown in Fig. 12. Sample frames and the ground truth depth are shown in (a)–(c). Our results at different stages are shown in (d)–(f) with errors coded in (g)–(i). The 3D scene flow vectors and errors are shown in (j)–(l). The corresponding errors are listed in Table 1(b). The per-frame statistics are plotted in the second row of Fig. 10. These visual and quantitative results verify the effectiveness of our method.

5.2 Multi-Pass Depth/Scene Flow Estimation

After obtaining scene flow and depth fields in our framework, we can use them to reconstruct motion trajectories and further refine the estimates. We have conducted experiments by running our system several passes, each taking the result from the previous pass as initialization. In this process, the scene flow fields are computed bidirectionally to construct new motion trajectories. The error statistics for *Traffic Scene 1* are provided in Table 2. It is notable that multi-pass estimation can only improve the result marginally because the temporal information has already been well incorporated in the first-pass estimation. For the sake of efficiency, we run our algorithm only once for all examples.

5.3 Structure Profile Evaluation

We in this section evaluate the usefulness of our structure profile $\tilde{C}^t_f$. Table 3 contains the statistics on *Traffic Scene 1*. The first-row errors are obtained by turning off the structure profile and simply setting $\tilde{C}^t_f(x) = 0$. The second-row statistics are with the edge map $C^t_f(x)$ constructed in each single frame as the structure prior. The last-row figures are obtained with the temporal structure profile $\tilde{C}^t_f(x)$. It is clear that employing our structure profiles yields the least errors.

Table 4 Error statistics with and without the long-range temporal constraints

Strategies	Scene flow		Disparity
	AAE	RMSE	MAE
Without temporal constraint	2.036	0.787	0.695
With temporal constraint	1.130	0.468	0.136

In Fig. 13, we compare the disparity maps produced using the single-frame edge prior and our temporal structure profile respectively. Temporally more consistent edges are preserved using our complete framework. In the upper image in (b), the rightmost car boundary is less erroneous. Problematic edges are also weakened, as compared in the bottom images. The corresponding edge maps are visualized in Fig. 6.

5.4 Long-Range Temporal Constraint Evaluation

We evaluate another important constraint in our method—that is, the long-range temporal depth and motion profiles. We conduct experiments with and without the robust temporal depth and motion profiles. Other components and parameters remain the same for fairness' sake.

When the temporal depth/motion constraint is not used, we alternatively perform variational minimization by setting α_u and α_d to zeros in Eqs. (12) and (16). One comparison is given in Fig. 14 where (a) shows three frames computed without the temporal constraints and (b) shows our results. Obvious problems on the moving car can be noticed in (a). The estimates with the temporal constraints are much better, especially for pixels in highlight and occlusion. Table 4 lists the corresponding errors generated with and without the long-range temporal constraints.

5.5 Challenging Natural Video Examples

We also apply our method to several natural video sequences containing multiple dynamic objects with subtle motion and deforming surfaces. Figures 15 and 16 show intermediate and final results of the *Balloons* and *Fish* sequences respectively. In each figure, the input images, initial depth maps, trajectory-based edge profiles that faithfully enhance boundaries, our depth maps obtained with robust regression, final depth results after sub-pixel refinement, and the final 3D scene flow fields are shown in rows in a top-down order. Figure 17 shows another dynamic scene with a moving person. Our depth maps and scene flow fields are in the second and third rows, respectively. We note that consistent depth and scene flow estimation for sequences containing large dynamic objects is very challenging due to noise, illumination variation, and occlusions. Our final results are with properly maintained temporal consistency.

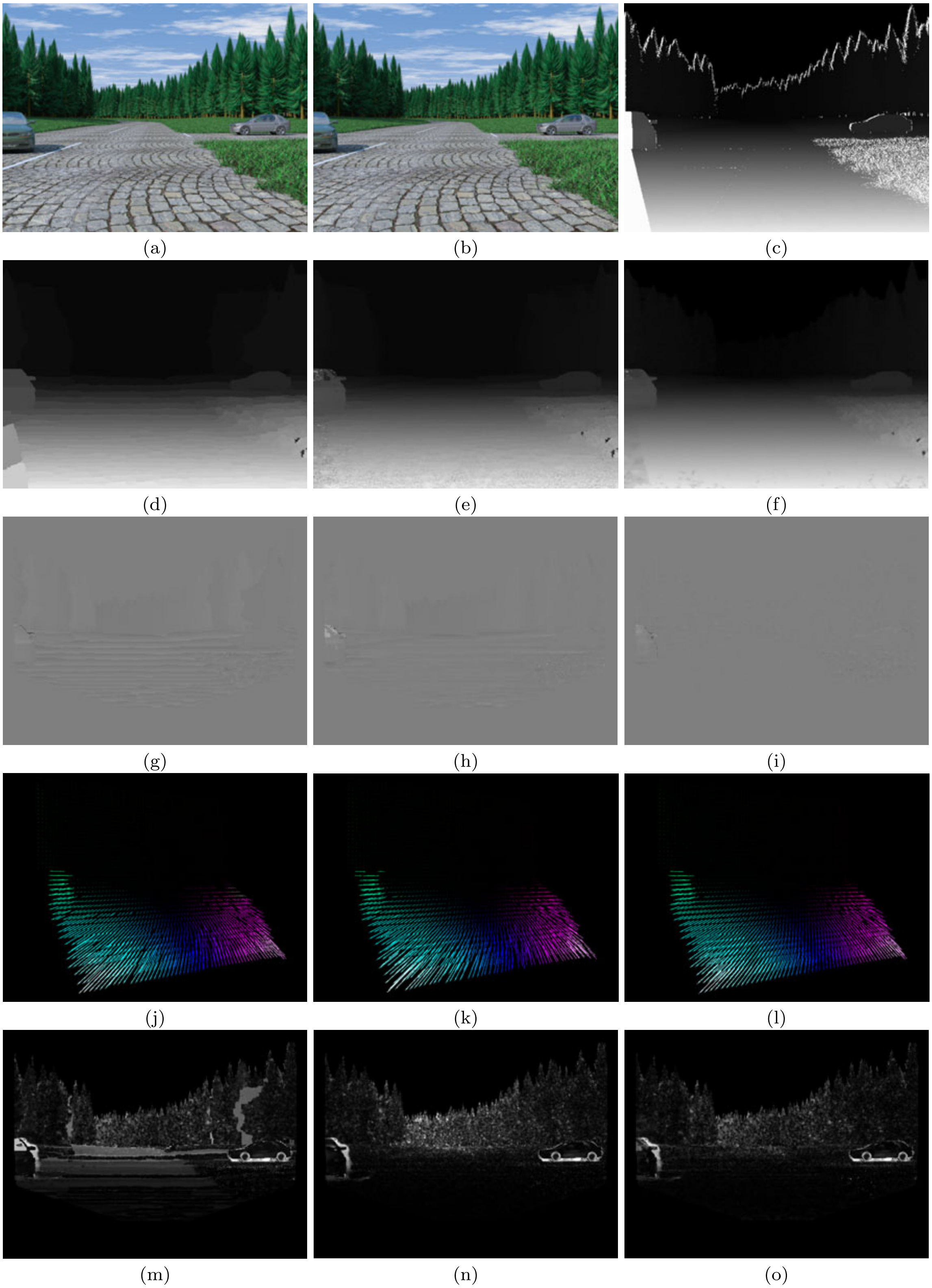


Fig. 12 *Traffic Scene 2* sequence. (**a**) and (**b**) are the left and right views respectively in time t. (**c**) is the ground truth disparity map, where white pixels are occlusion. (**d**)–(**f**) are disparity maps obtained in initialization, robust regression, and final refinement, respectively. (**g**)–(**i**) are the error images for (**d**)–(**f**) respectively, where dark to bright pixels are with negative to positive errors. (**j**)–(**l**) visualize the scene flow fields corresponding to (**g**)–(**i**). (**m**)–(**o**) show the angular error maps of scene flow

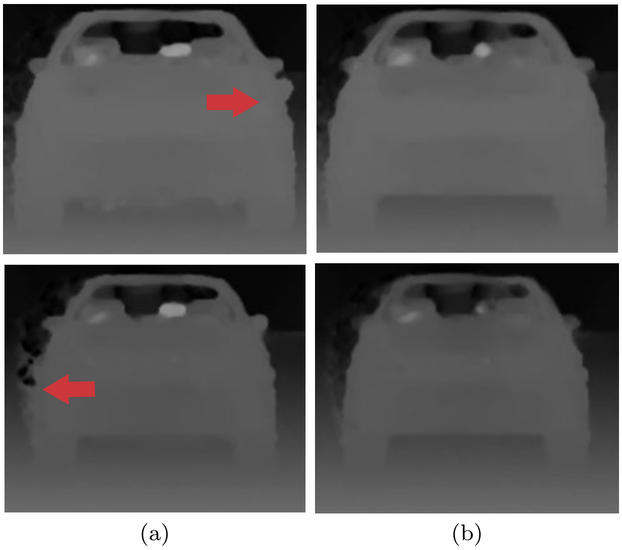


Fig. 13 Results obtained with (**a**) the single-frame edge prior and (**b**) our temporal structure profile

As the 2D-plus-depth solution for 3DTV utilizes depth information to synthesize new views, our method is capable of converting standard two-view 3D videos into the 2D-plus-depth format. To demonstrate the accuracy and consistency of our resulting depth maps, we synthesize new view sequences in Figs. 18 and 19. New views produced from the initial depth are also shown for comparison, which contain obvious visual artifacts. In our final results, continuous variation of depth both spatially and temporally is preserved, together with discontinuous object boundaries, thanks to the effective regularization, multi-frame profile construction, and robust optimization. The full novel-view-synthesis results are in the supplementary video.

5.6 Computational Cost

Although our method employs long-range temporal constraints, the computational cost does not increase exponentially compared with single-frame methods. Despite constructing trajectory-based priors, the optimization procedure between different frames is independent. It is also possible to use parallel computing techniques, such as OpenMP (OpenMP ARB 2012), for acceleration.

For the binocular 100-frame video *Traffic Scene 1* with resolution 640 × 480, using 24 cores of Intel Xeon @2.67 GHz, each frame only takes 1 min on average. We list in two rows of Table 5 the running time of depth-and-motion initialization and of the whole system when using a single CPU core.

5.7 More Discussion

Temporal consistency is known as very important in estimating depth and scene flow maps in video sequences. However, finding good constraints in multiple frames is still an open problem. Previous joint estimation methods are generally insufficient for long range information propagation. Recent developments in scene flow estimation (Basha et al. 2010; Vogel et al. 2011; Wedel et al. 2011) did not tackle the problem from the temporal consistency point of view either. Our approach explicitly models temporal constraints using chained priors, which makes temporal propagation practical and efficient. This marks the major differences between the proposed approach and others. Our chained temporal priors described in Sect. 4.2 are also very general, which not only benefit depth and scene flow estimation for binocular sequences, but also work for cameras with active range sensors, such as Microsoft Kinect.

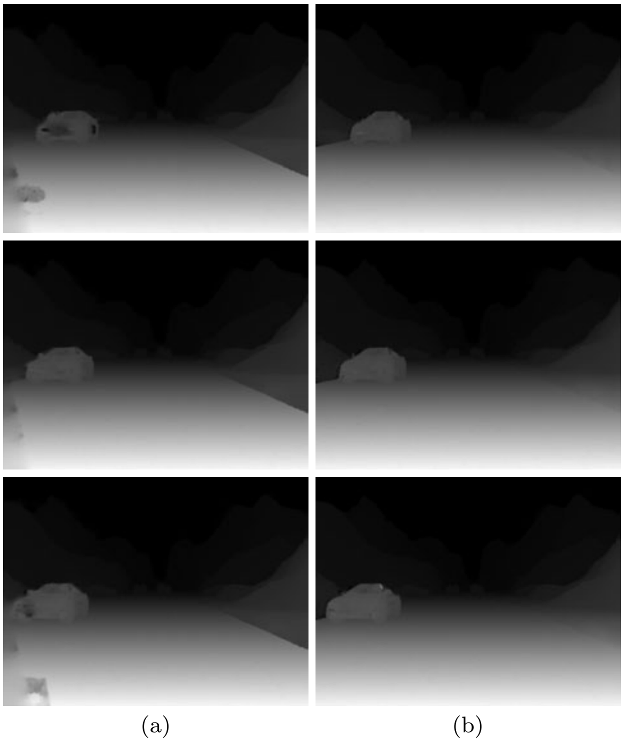


Fig. 14 Disparity result comparison (**a**) without the temporal depth profile and (**b**) with the temporal depth profile

Table 5 Time (in minutes) spent for data initialization, the whole process on a single core, and for the whole system running on 24 cores

	Total running time	Per frame running time
Initial	1287	13
Final	2772	28
Final+OMP	138	1.4

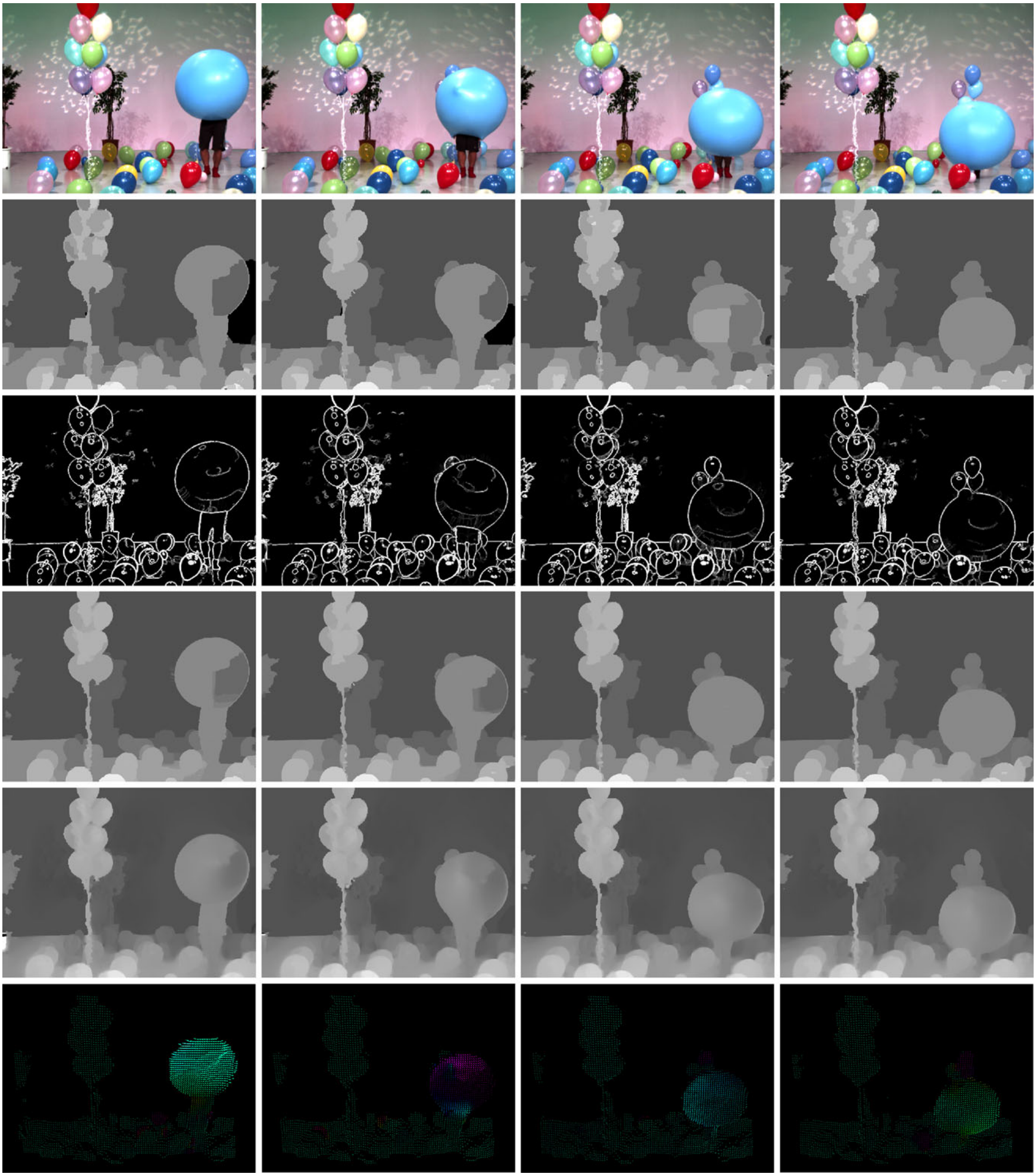


Fig. 15 The *Balloons* sequence results. The *1st row*: input images. The *2nd row*: initial depth maps. The *3rd row*: trajectory-based structure profiles. The *4th row*: depth maps after temporal refinement. The *5th row*: depth maps after sub-pixel refinement. The *6th row*: final color-coded scene flow fields

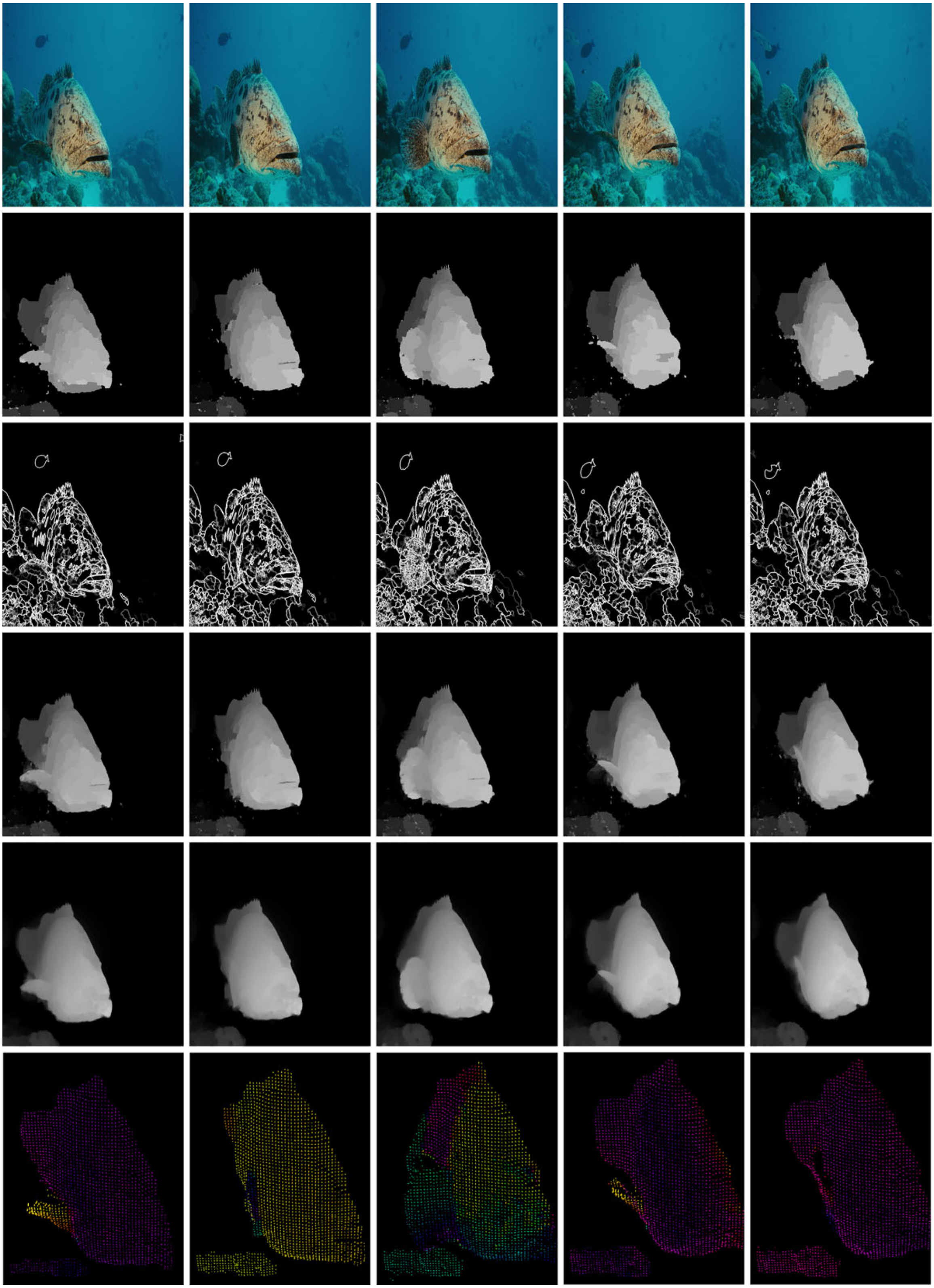


Fig. 16 The *Fish* sequence results. The *1st row*: input images; the *2nd row*: initial depth maps; the *3rd row*: trajectory-based structure profiles; the *4th row*: depth maps after temporal refinement; the *5th row*: depth maps after sub-pixel continuous refinement; the *6th row*: final color-coded scene flow fields

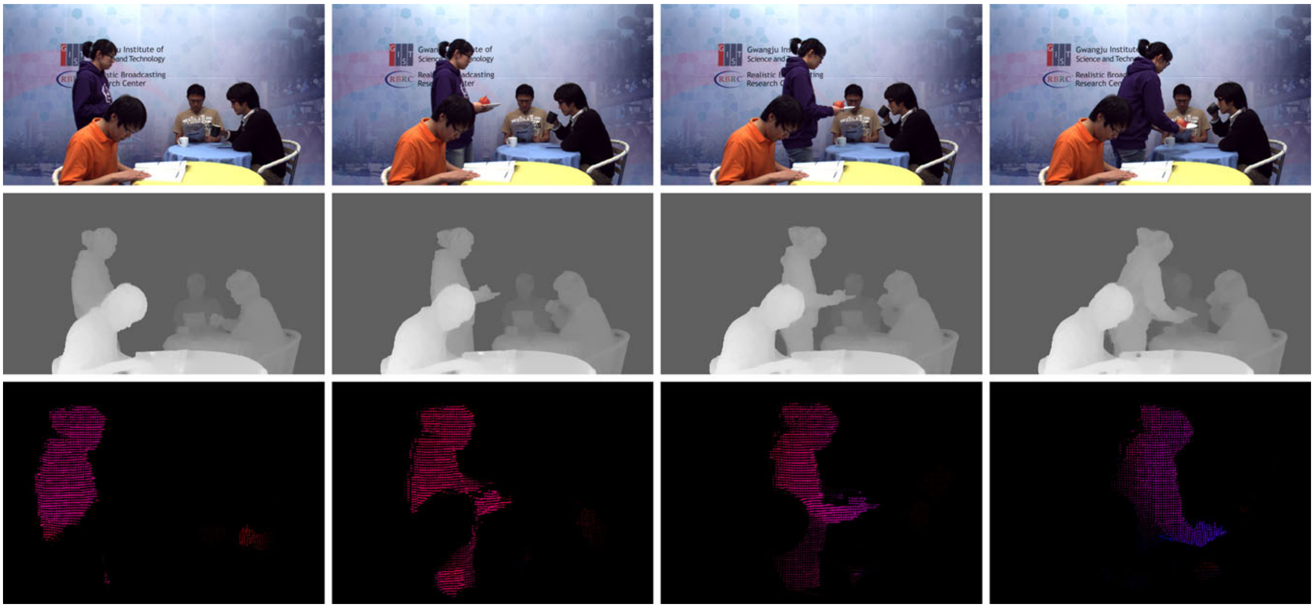


Fig. 17 The *Cafe* sequence results. From top to bottom: input images, our depth maps, and our scene flow fields

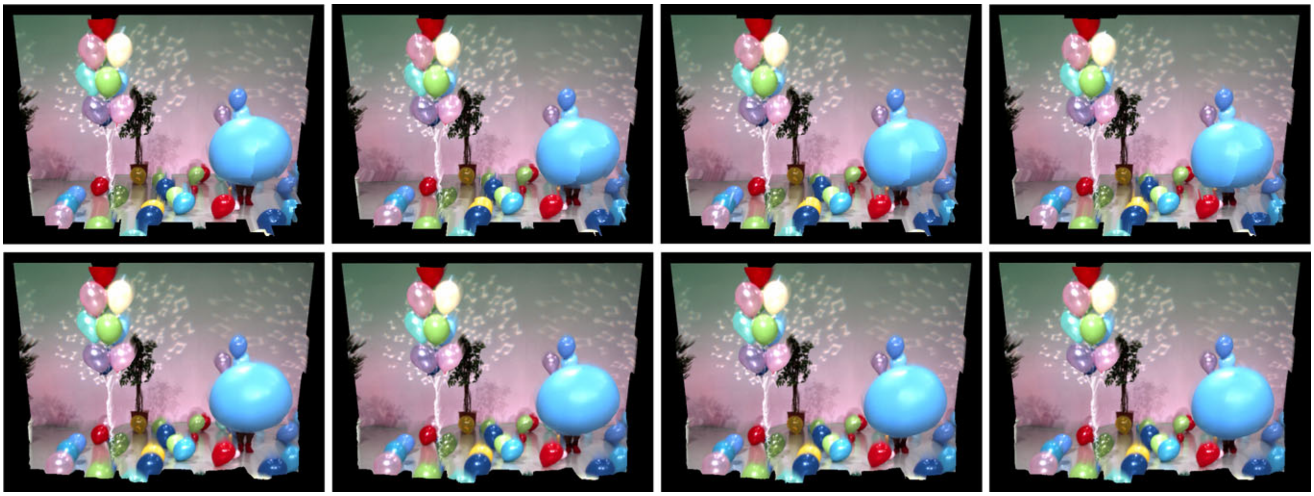


Fig. 18 Novel-view synthesis for frames 16–19 of the *Balloons* sequence. The *1st row*: initial depth; the *2nd row*: depth after sub-pixel refinement. Visual artifacts are suppressed in the final results, while boundaries are faithfully preserved

6 Conclusion

In summary of our method, the major novelty lies on the long-range temporal constraints, which significantly improve the depth and scene flow consistency both visually and quantitatively. In building the robust estimation system, our method contributes in the following ways. Firstly, the motion trajectory construction can find reliable estimates consecutively and break links when occlusion consistently arises in multiple frames. Occasional noise in one or two frames, on the contrary, can be robustly ignored. Secondly, the novel edge occurrence maps are constructed incorporating structural information from multiple frames. The voting-like average scheme greatly suppresses errors that cannot be coherent in multiple frames and enhances credible estimates. Thirdly, we propose the anisotropic smoothing scheme to provide proper regularization for all pixels based on the structure profiles.

Limitation and Future Work Our method needs reasonable depth initialization. If, for one region, its initial depth estimation is consistently wrong for all frames, the following refinement would not improve it much. Our future work includes the extension to unrectified videos and acceleration using GPU.

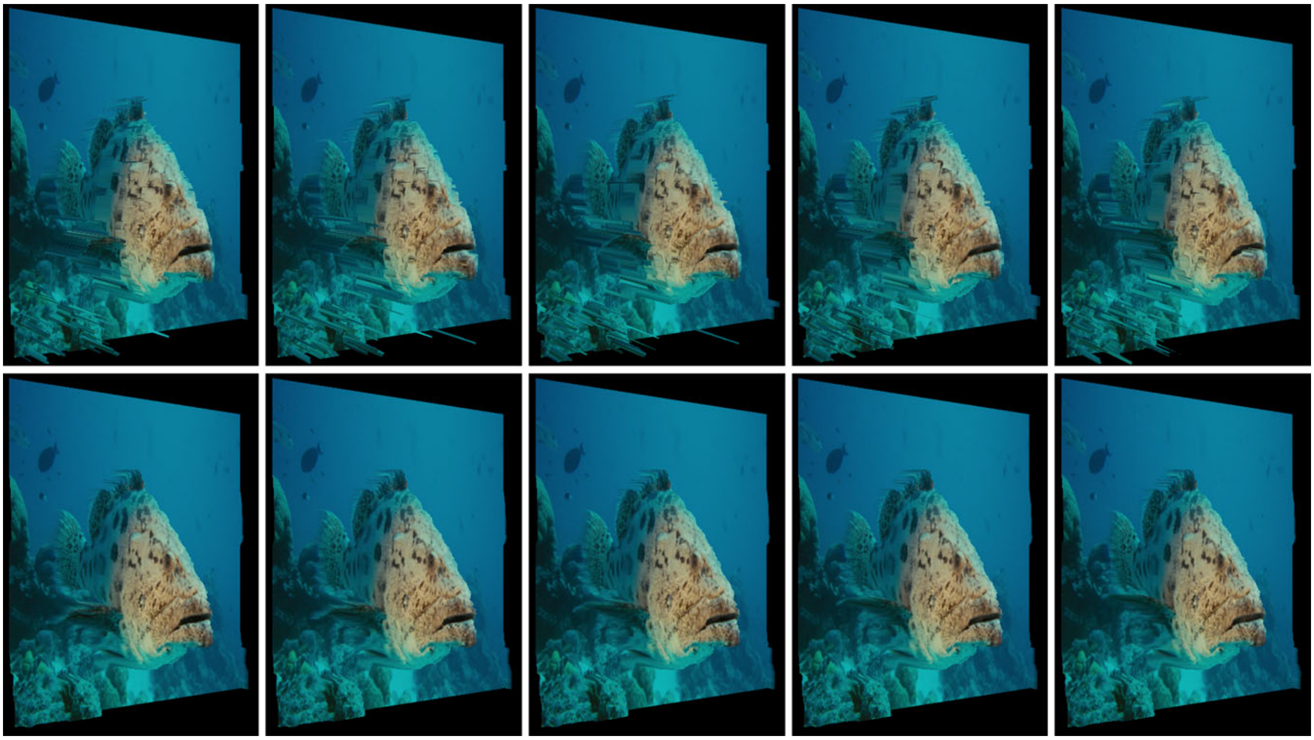


Fig. 19 Novel-view synthesis for frames 76–80 of the *Fish* sequence. The *1st row*: initial depth. The *2nd row*: depth after sub-pixel continuous optimization

Acknowledgements The authors would like to thank the associate editor and all the anonymous reviewers for their time and effort. This work is supported by a grant from the Research Grants Council of the Hong Kong SAR (Project No. 413110).

Appendix A

We give the details of solving the Euler-Lagrange equation (21):

$$0 = \hat{o}_d \sum_k \Gamma'\big((\varepsilon_d^{[k]})^2\big) \cdot \varepsilon_d^{[k]} f_{dh}^{[k]} + 2\alpha_{\mathrm{d}} \Delta \mathrm{d} \\ - \beta_{\mathrm{d}} \mathrm{div}\big(\Gamma'\big(\nabla \mathrm{d}^T D(\nabla f_I) \nabla \mathrm{d}\big) \cdot D(\nabla \mathrm{d}) \nabla \mathrm{d}\big).$$

With the applied anisotropic diffusion tensor, the smoothness term involves d_{hh}, d_{vv}, and d_{hv}, which relate several neighboring points. We use the indices in Fig. 20 to represent the 2D coordinates: $\mathrm{d}_1 = \mathrm{d}(i+1, j+1)$. q is used to index the current point (i, j). We apply central difference in the second order derivative computation. Specifically, we introduce function $\zeta(\cdot)$ expressed as

$$(\zeta \mathrm{d}_h)_h := \left(\frac{\zeta_2 + \zeta_p}{2} \frac{\mathrm{d}_2 - \mathrm{d}_q}{h_h} - \frac{\zeta_6 + \zeta_p}{2} \frac{\mathrm{d}_q - \mathrm{d}_6}{h_h} \right) \Big/ h_h,$$

$$(\zeta \mathrm{d}_v)_h := \left(\frac{\zeta_2 + \zeta_p}{2 h_h h_v} \frac{\mathrm{d}_0 + \mathrm{d}_1 - \mathrm{d}_3 - \mathrm{d}_4}{4} \right. \\ \left. - \frac{\zeta_6 + \zeta_p}{2 h_h h_v} \frac{\mathrm{d}_0 + \mathrm{d}_7 - \mathrm{d}_4 - \mathrm{d}_5}{4} \right).$$

Fig. 20 Indices for the 2D coordinates

7 (i-1,j+1)	0 (i,j+1)	1 (i+1,j+1)
6 (i-1,j)	q (i,j)	2 (i+1,j)
5 (i-1,j-1)	4 (i,j-1)	3 (i+1,j-1)

$(\zeta \mathrm{d}_v)_v$ and $(\zeta \mathrm{d}_h)_v$ are defined similarly. Then we discretize a grid with size $h_h \times h_v$ to apply Gauss-Seidel relaxation. By defining

$$r_h = \big(f_{\partial v}^2 + (1 - \tilde{C}) f_{\partial h}^2\big) / \big(\|\nabla f_I\|^2 + \epsilon\big),$$

$$r_v = \big(f_{\partial h}^2 + (1 - \tilde{C}) f_{\partial v}^2\big)\big(\|\nabla f_I\|^2 + \epsilon\big),$$

$$r_{h1} = r_h(i+1, j+1),$$

$$r_c = \frac{-\tilde{C} f_{\partial h} f_{\partial v}}{\|\nabla f_I\|^2 + \epsilon},$$

we represent the anisotropic factors in simpler forms. The increment $\Delta \mathrm{d}$ can be computed using the following iterations:

$$\Delta \mathrm{d}_p = b/a,$$

$$a = \hat{o}_d \sum_k \Gamma'\big(\big(\varepsilon_d^{[k]}\big)^2\big) \cdot (f_h)^2 + 2\alpha_{\mathrm{d}} + \beta_{\mathrm{d}} g_1(\mathrm{d}),$$

where $\mathcal{N}$ is the set of neighboring pixels, $\mathcal{N}_h(q) = \{2, 6\}$, and $\mathcal{N}_v(q) = \{0, 4\}$. Further, g_1 is defined as

$$g_1(\mathrm{d}) = \sum_{\diamond\in\{h,v\}} \sum_{p\in\mathcal{N}_\diamond(q)} \frac{\Gamma'_{s\mathrm{d}q} r_{\diamond q} + \Gamma'_{s\mathrm{d}p} r_{\diamond p}}{2h_\diamond^2},$$

and b can be derived as

$$b = -\hat{o}_d \sum_k \Gamma'\big(\big(\varepsilon_d^{[k]}\big)^2\big) \cdot f_h f_z + \beta_{\mathrm{d}} g_2(\mathrm{d}),$$

where

$$\begin{aligned} g_2(\mathrm{d}) = {} & \sum_{\diamond\in\{h,v\}} \sum_{p\in\mathcal{N}_\diamond(q)} \frac{\Gamma'_{sq} r_{\diamond q} + \Gamma'_{sp} r_{\diamond p}}{2h_\diamond^2} \big(\mathrm{d}_p - \mathrm{d}_q^{(0)}\big) \\ & + \sum_{p\in\{0,2,4,6\}} \frac{\Gamma'_{sr} r_{cp} + \Gamma'_{sq} r_{cq}}{2h_h h_v} \\ & \times \frac{(\mathrm{d}_{\overline{p-2}} + \mathrm{d}_{\overline{p-1}} - \mathrm{d}_{\overline{p+1}} - \mathrm{d}_{\overline{p+2}})}{4}. \end{aligned}$$

$\overline{p} = p \mod 8$. To facilitate computation, we adopt a standard non-linear multi-grid numerical scheme (Bruhn and Weickert 2005) to accelerate convergence. The Gauss-Seidel relaxation works as the pre- and post-smoother, which is applied twice in each level.

Appendix B

After discretization, the linear equations to approximate Eq. (20) can be easily derived. Δu, Δv, and $\Delta\delta\mathrm{d}$ are iteratively refined, by fixing the other two variables during update. It leads to the Gauss-Seidel relaxation, written as

$$\Delta u = b_u / a_u,\ \Delta v = b_v / a_v,\ \Delta\delta\mathrm{d} = b_{\delta\mathrm{d}} / a_{\delta\mathrm{d}}$$

where

$$\begin{aligned} a_u &= C_u + 2\alpha_{\mathrm{u}} + \beta_{\mathrm{u}} g_1(u) \\ b_u &= -D_{-u} + +\beta_{\mathrm{u}} g_2(u) \\ a_v &= C_v + 2\alpha_{\mathrm{u}} + \beta_{\mathrm{u}} g_1(v) \\ b_v &= -D_{-v} + \beta_{\mathrm{u}} g_2(v) \\ a_{\delta\mathrm{d}} &= C_{\delta\mathrm{d}} + 2\alpha_{\mathrm{u}} \hat{o}_u + \beta_{\mathrm{u}} g_1(\delta\mathrm{d}) \\ b_{\delta\mathrm{d}} &= -D_{-\delta\mathrm{d}} + \beta_{\mathrm{u}} g_2(\delta\mathrm{d}) \\ C_u &= \hat{o}_d \hat{o}_u \sum_k \Gamma'\big(\big(\varepsilon_R^{[k]} - \varepsilon_L^{[k]}\big)^2\big) \cdot (f_{rh} - f_{lh})^2 \\ &\quad + \hat{o}_u \Gamma'\big(\big(\varepsilon_L^{[k]}\big)^2\big) f_{lh}^2 + \hat{o}_d \hat{o}_u \Gamma'\big(\big(\varepsilon_R^{[k]}\big)^2\big) f_{rh}^2 \end{aligned}$$

$$\begin{aligned} D_{-u} &= \hat{o}_d \hat{o}_u \sum_k \Gamma'\big(\big(\varepsilon_R^{[k]} - \varepsilon_L^{[k]}\big)^2\big) \cdot \big(\varepsilon_R^{[k]} - \varepsilon_L^{[k]}\big)(f_{rh} - f_{lh}) \\ &\quad + \hat{o}_u \Gamma'\big(\big(\varepsilon_L^{[k]}\big)^2\big) \varepsilon_L^{[k]} f_{lh} + \hat{o}_d \hat{o}_u \Gamma'\big(\big(\varepsilon_R^{[k]}\big)^2\big) \varepsilon_R^{[k]} f_{rh} \\ &\quad - C_u \Delta u \\ C_v &= \hat{o}_d \hat{o}_u \sum_k \Gamma'\big(\big(\varepsilon_R^{[k]} - \varepsilon_L^{[k]}\big)^2\big) \cdot (f_{rv} - f_{lv})^2 \\ &\quad + \hat{o}_u \Gamma'\big(\big(\varepsilon_L^{[k]}\big)^2\big) f_{lv}^2 + \hat{o}_d \hat{o}_u \Gamma'\big(\big(\varepsilon_R^{[k]}\big)^2\big) f_{rv}^2 \\ D_{-v} &= \hat{o}_d \hat{o}_u \sum_k \Gamma'\big(\big(\varepsilon_R^{[k]} - \varepsilon_L^{[k]}\big)^2\big) \cdot \big(\varepsilon_R^{[k]} - \varepsilon_L^{[k]}\big)(f_{rv} - f_{lv}) \\ &\quad + \hat{o}_u \Gamma'\big(\big(\varepsilon_L^{[k]}\big)^2\big) \varepsilon_L^{[k]} f_{lv} + \hat{o}_d \hat{o}_u \Gamma'\big(\big(\varepsilon_R^{[k]}\big)^2\big) \varepsilon_R^{[k]} f_{rv} \\ &\quad - C_v \Delta v \\ C_{\delta\mathrm{d}} &= \hat{o}_d \hat{o}_u \sum_k \Gamma'\big(\big(\varepsilon_R^{[k]} - \varepsilon_L^{[k]}\big)^2\big) \cdot f_{rh}^2 + \hat{o}_d \hat{o}_u \Gamma'\big(\big(\varepsilon_R^{[k]}\big)^2\big) f_{rh}^2 \\ D_{-\delta\mathrm{d}} &= \hat{o}_d \hat{o}_u \sum_k \Gamma'\big(\big(\varepsilon_R^{[k]} - \varepsilon_L^{[k]}\big)^2\big) \cdot \big(\varepsilon_R^{[k]} - \varepsilon_L^{[k]}\big)(f_{rh}) \\ &\quad + \hat{o}_d \hat{o}_u \Gamma'\big(\big(\varepsilon_R^{[k]}\big)^2\big) \varepsilon_R^{[k]} f_{rh} - C_{\delta\mathrm{d}} \Delta\delta\mathrm{d}. \end{aligned}$$

g_1,g_2 are functions defined in Appendix A. The Gauss-Seidel iteration is accelerated by a non-linear Multi-grid numerical scheme similar to the one to compute disparities in Appendix A.

References

Álvarez, L., Deriche, R., Papadopoulo, T., & Sánchez, J. (2007). Symmetrical dense optical flow estimation with occlusions detection. *International Journal of Computer Vision*, *75*, 371–385.

Baker, S., Scharstein, D., Lewis, J. P., Roth, S., Black, M. J., & Szeliski, R. (2011). A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, *92*, 1–31.

Basha, T., Moses, Y., & Kiryati, N. (2010). Multi-view scene flow estimation: a view centered variational approach. In *CVPR* (pp. 1506–1513).

Black, M. J. (1994). Recursive non-linear estimation of discontinuous flow fields. In *ECCV* (Vol. 1, pp. 138–145).

Brox, T., Bruhn, A., Papenberg, N., & Weickert, J. (2004). High accuracy optical flow estimation based on a theory for warping. In *ECCV* (Vol. 4, pp. 25–36).

Brox, T., Bregler, C., & Malik, J. (2009). Large displacement optical flow. In *CVPR* (pp. 41–48).

Bruhn, A., & Weickert, J. (2005). Towards ultimate motion estimation: combining highest accuracy with real-time performance. In *ICCV* (pp. 749–755).

Bruhn, A., Weickert, J., & Schnörr, C. (2005). Lucas/Kanade meets horn/Schunck: combining local and global optic flow methods. *International Journal of Computer Vision*, *61*, 211–231.

Cech, J., Sanchez-Riera, J., & Horaud, R. (2011). Scene flow estimation by growing correspondence seeds. In *CVPR* (pp. 3129–3136).

Furukawa, Y., & Ponce, J. (2007). Accurate, dense, and robust multi-view stereopsis. In *CVPR* (pp. 1362–1376).

Hadfield, S., & Bowden, R. (2011). Kinecting the dots: particle based scene flow from depth sensors. In *ICCV* (pp. 2290–2295).

Huguet, F., & Devernay, F. (2007). A variational method for scene flow estimation from stereo sequences. In *ICCV* (pp. 1–7).

Irani, M. (2002). Multi-frame correspondence estimation using subspace constraints. *International Journal of Computer Vision*, *48*, 173–194.

Kolmogorov, V., & Zabih, R. (2004). What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*, 147–159.

Min, D. B., & Sohn, K. (2006). Edge-preserving simultaneous joint motion-disparity estimation. In *ICPR* (Vol. 2, pp. 74–77).

OpenMP ARB (2012). Open multi-processing. http://openmp.org/.

Patras, I., Alvertos, N., & Tziritas, G. (1996). Joint disparity and motion field estimation in stereoscopic image sequences. In *International conference on pattern recognition* (Vol. 1, pp. 359–363).

Rabe, C., Müller, T., Wedel, A., & Franke, U. (2010). Dense, robust, and accurate motion field estimation from stereo image sequences in real-time. In *ECCV* (Vol. 4, pp. 582–595).

Richardt, C., Orr, D., Davies, I., Criminisi, A., & Dodgson, N. A. (2010). Real-time spatiotemporal stereo matching using the dual-cross-bilateral grid. In *ECCV* (Vol. 3, pp. 510–523).

Sand, P., & Teller, S. J. (2006). Particle video: long-range motion estimation using point trajectories. In *CVPR* (Vol. 2, pp. 2195–2202).

Sand, P., & Teller, S. J. (2008). Particle video: long-range motion estimation using point trajectories. *International Journal of Computer Vision*, *80*, 72–91.

Scharstein, D., & Szeliski, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, *47*, 7–42.

Snavely, N., Seitz, S. M., & Szeliski, R. (2006). Photo tourism: exploring photo collections in 3d. *ACM Transactions on Graphics*, *25*, 835–846.

Sun, D., Roth, S., Lewis, J. P., & Black, M. J. (2008). Learning optical flow. In *ECCV* (Vol. 3, pp. 83–97).

Sundaram, N., Brox, T., & Keutzer, K. (2010). Dense point trajectories by gpu-accelerated large displacement optical flow. In *ECCV* (Vol. 1, pp. 438–451).

Tomasi, C., & Manduchi, R. (1998). Bilateral filtering for gray and color images. In *ICCV* (pp. 839–846).

University of Auckland (2008). Enpeda. Image sequence analysis test site (eisats). http://www.mi.auckland.ac.nz/EISATS/.

Valgaerts, L., Bruhn, A., Zimmer, H., Weickert, J., Stoll, C., & Theobalt, C. (2010). Joint estimation of motion, structure and geometry from stereo sequences. In *ECCV* (Vol. 4, pp. 568–581).

Vaudrey, T., Rabe, C., Klette, R., & Milburn, J. (2008). Differences between stereo and motion behavior on synthetic and real-world stereo sequences. In *International conference of image and vision computing New Zealand (IVCNZ)* (pp. 1–6).

Vedula, S., Baker, S., Rander, P., Collins, R. T., & Kanade, T. (2005). Three-dimensional scene flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*, 475–480.

Vogel, C., Schindler, K., & Roth, S. (2011). 3d scene flow estimation with a rigid motion prior. In *ICCV* (pp. 1291–1298).

Wedel, A., Rabe, C., Vaudrey, T., Brox, T., Franke, U., & Cremers, D. (2008). Efficient dense scene flow from sparse or dense stereo data. In *ECCV* (Vol. 1, pp. 739–751).

Wedel, A., Brox, T., Vaudrey, T., Rabe, C., Franke, U., & Cremers, D. (2011). Stereoscopic scene flow computation for 3d motion understanding. *International Journal of Computer Vision*, *95*, 29–51.

Xiao, J., Cheng, H., Sawhney, H. S., Rao, C., & Isnardi, M. A. (2006). Bilateral filtering-based optical flow estimation with occlusion detection. In *ECCV* (Vol. 1, pp. 211–224).

Xu, L., Chen, J., & Jia, J. (2008). A segmentation based variational model for accurate optical flow estimation. In *ECCV* (Vol. 1, pp. 671–684).

Xu, L., Jia, J., & Matsushita, Y. (2010). Motion detail preserving optical flow estimation. In *CVPR* (pp. 1293–1300).

Yoon, K. J., & Kweon, I. S. (2006). Adaptive support-weight approach for correspondence search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*, 650–656.

Zhang, Z., & Faugeras, O. D. (1992). Estimation of displacements from two 3-d frames obtained from stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *14*, 1141–1156.

Zhang, Y., & Kambhamettu, C. (2001). On 3d scene flow and structure estimation. In *CVPR* (Vol. 2, pp. 778–785).

Zhang, L., Curless, B., & Seitz, S. M. (2003). Spacetime stereo: shape recovery for dynamic scenes. In *CVPR* (Vol. 2, pp. 367–374).

Zhang, G., Jia, J., Wong, T. T., & Bao, H. (2009). Consistent depth maps recovery from a video sequence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *31*, 974–988.

Zimmer, H., Bruhn, A., Weickert, J., Valgaerts, L., Salgado, B. R. A., & Seidel, H. P. (2009). Complementary optic flow. In *EMMCVPR* (pp. 207–220).