Motion-Depth: RGB-D Depth Map Enhancement with Motion and Depth in Complement

Tak-Wai Hui and King Ngi Ngan
Department of Electronic Engineering
The Chinese University of Hong Kong
{twhui, knngan}@ee.cuhk.edu.hk

Abstract

Low-cost RGB-D imaging system such as Kinect is widely utilized for dense 3D reconstruction. However, RGB-D system generally suffers from two main problems. The spatial resolution of the depth image is low. The depth image often contains numerous holes where no depth measurements are available. This can be due to bad infra-red reflectance properties of some objects in the scene. Since the spatial resolution of the color image is generally higher than that of the depth image, this paper introduces a new method to enhance the depth images captured by a moving RGB-D system using the depth cues from the induced optical flow. We not only fill the holes in the raw depth images, but also recover fine details of the imaged scene. We address the problem of depth image enhancement by minimizing an energy functional. In order to reduce the computational complexity, we have treated the textured and homogeneous regions in the color images differently. Experimental results on several RGB-D sequences are provided to show the effectiveness of the proposed method.

1. Introduction

Depth map of a scene is an important piece of information to understand the 3D geometry of the environment. With the ability of acquiring scene depth, many applications such as autonomous navigation, 3D reconstruction, human-computer interaction and more, can be achieved.

Due to the advance in technology, active depth sensors such as imperceptible structured light sensor (e.g., Kinect [3]) and time-of-flight (ToF) sensor (e.g., SwissRanger [4]) become widely available. They are not only targeted for entertainment, but also widely utilized for research. Generally speaking, low-cost depth camera is only able to provide low-resolution depth maps for the imaged scene. Therefore, fine details of the scene are lost. Moreover, the depth maps are usually noisy and incomplete. Kinect is a kind of RGB-D imaging system that provides both color (RGB) and depth (D) images of the imaged scene. We have used a Kinect to capture a small part of our laboratory. Figure 1 illustrates the color image and the depth map in the coordinate frame of the color image. The presence of numerous holes in black color represents no available depth readings.

Figure 1. A RGB-D frame from a Kinect consists of a color image and a depth map of the scene. Light intensities in the depth map correspond to surfaces that are close to the camera, while dark intensities correspond to surfaces that are far away from the camera.

Generally, we can classify the approaches for the recovery of scene depth into three groups, namely active sensors (e.g., ToF sensor and Kinect) [14] [26], passive sensors (i.e., color camera) [10] [18] [25] [27] [28] [30] [36], and fusion of active and passive sensors [39] [37] [9] [11] [12].

Passive system generally utilizes techniques such as shape from motion (SfM) or stereo disparity to recover scene depth. Estimation of camera pose and establishment of dense correspondences are the two important steps in SfM. Optical flow which is closely related to SfM has attracted the attention of many researchers since the seminal works of Horn and Schunck [15] as well as Lucas and Kanade [24]. The major challenge in optical flow estimation lies on the fact that the flow is only partially observable in general due to the well-known aperture problem. In order to overcome the difficulty, the flow is recovered by solving a functional, usually consisting of a data term and a regularization term [7] [19] [31] [34] [40]. However, solving the functional is a computationally intensive task. This leads to implement various optical flow algorithms in a powerful computing device equipped with multi-core processor and/or graphics processing unit (GPU). On the other hand, the recovery of camera pose in different frames from feature correspondences [13] or spatial-temporal image gradients [16] [17] requires relatively lower computational time.

The complementary nature of active and passive sensors provides a great opportunity to improve the quality of the depth maps. However, the aforementioned research works for sensor fusion are all about enhancement of depth maps using stereo and active depth camera together. To the best of our knowledge, no research work has been done on the fusion of the depth maps captured by a handheld RGB-D imaging system and the depth cues from the induced optical flow. This is partially due to the fact that the computation of variational optical flow is generally not fast enough.

In this paper, we provide a novel two-step variational method to enhance the depth maps from a moving RGB-D system using two RGB-D frames. Our contribution is three-fold. First, we fill the research gap in RGB-D depth map enhancement of what we believe is the first work which utilizes the depth cues in the optical flow induced from a short image sequence to complement the depth maps. Second, we improve the computational efficiency of solving the depth functional by treating the textured and homogeneous regions in the RGB images differently. Third, we develop a sparse data term and specialize a dense regularization term for RGB-D depth map enhancement.

2. Related works

SfM is one of the most popular approaches for 3D or depth map reconstruction. Davison proposed the use of a top-down Bayesian framework for the simultaneous localization and mapping (SLAM) using a single camera via the registration of a sparse set of feature points [10]. Another pioneering work which is related to SLAM is the parallel tracking and mapping (PTAM) proposed by Klein et $al.$[18]. In PTAM, reliable image features are first selected and tracked across multiple image frames. Then, sparse 3D point clouds are mapped together using bundle adjustment. Their approach emphasizes on the tracking and mapping simultaneously by using multi-thread processing.

Recent research works on SfM focus on dense 3D reconstruction due to the advance in computing devices. Newcombe et $al.$ recovered camera pose using a sparse point cloud [25]. An approximate base mesh which is generated from the point cloud is warped to form a 3D model of the scene. Then, the mesh is continuously refined using the view-predictive optical flow. Stühmer et $al.$ proposed a TVL_1-based variational approach for computing dense depth maps from multiple images [30]. Later, Newcombe et $al.$ proposed dense tracking and mapping (DTAM) which relies not on sparse feature correspondences but information from all image points [27].

A nonlinear measure of the sub-pixel displacement was proposed by Psarakis et $al.$ for stereo correspondence [28]. However, the sub-pixel accuracy is achieved by interpolation only. A fast nonlocal cost aggregation method for stereo images using minimum spanning tree (MST) was proposed by Yang [36]. The quality of MST directly influences the matching result.

The complementary nature of ToF and stereo has attracted the attention of many researchers. Yang et $al.$ weighted the depth measurements from ToF and stereo according to the signal strength of ToF and the local image features, respectively [37]. Choi et $al.$ proposed a method which is able to disambiguate depth measurements from ToF caused by phase warping using stereo [9]. Garcia et $al.$ presented a depth map refinement procedure by separately treating regions with undesired effects such as occlusion and missing measurements differently [12]. Gandhi et $al.$ proposed a ToF-stereo fusion which utilizes the ToF measurements projected onto the stereo image pair as an initial set of correspondences [11]. High-resolution range data is then formed by propagating these correspondences with the rough depth priors using a Bayesian model. Zhu et $al.$ modeled the fusion of ToF and stereo as maximizing a posterior Markov random field (MAP-MRF) [39].

Our work is related to the TV-L_1 approaches [27] [30]. Both of us utilize motion prior to recover dense depth map bypassing the estimation of the optical flow field. Unlike their works, we do not solve the functional by introducing auxiliary function. Moreover, we improve the computational efficiency by solving the depth functional for the textured and homogeneous regions differently. For KinectFusion [26] and RGB-D Mapping [14], multiple depth measurements are integrated into a global 3D model. Color images are only utilized for rendering and depth map filtering in KinectFusion, while feature correspondences are used to facilitate point clouds alignment in RGB-D Mapping. Here, we restrict ourselves to the condition that only two consecutive color and depth images are provided. We enhance the depth images by exploring the depth cues from the induced optical flow of the moving RGB-D system.

3. Variational optical flow

Suppose we have two consecutive RGB images $I_1(\mathbf{x})$ and $I_2(\mathbf{x})$ which are captured by a moving RGB-D system. Optical flow field $\dot{\mathbf{x}}$ is often recovered by minimizing an energy functional of the form:

$$\begin{aligned} E(\dot{\mathbf{x}}) &= E_{data}(\dot{\mathbf{x}}) + E_{reg}(\nabla\dot{\mathbf{x}}) \\ &= \int e_{data}(\dot{\mathbf{x}}) + e_{reg}(\nabla\dot{\mathbf{x}}) d\mathbf{x}, \end{aligned} \tag{1}$$

where e_{data} and e_{reg} are the data and the regularization terms to the data E_{data} and the regularization E_{reg} energies, respectively. A review about some of the current variational optical flow models can be found in [40].

To minimize the functional in (1), we can apply the corresponding Euler Lagrange equation to update the flow field at iteration step τ as follows:

$$\frac{\partial \dot{\mathbf{x}}}{\partial \tau} = \dot{\mathbf{x}}^{\tau} - \dot{\mathbf{x}}^{\tau-1} = -\left(\frac{\partial e_{data}(\dot{\mathbf{x}})}{\partial \dot{\mathbf{x}}} - div\left(\frac{\partial e_{reg}(\nabla \dot{\mathbf{x}})}{\partial \nabla \dot{\mathbf{x}}} \right) \right). \tag{2}$$

3.1. Data term

The data term in (1) is related to the image brightness constancy assumption in the work of Horn and Schunck [15]. E_{data} is widely defined as the integral of the difference of the image intensities within a rectangular image domain $\Pi \subset \mathbb{R}^2$ as:

$$E_{data}(\dot{\mathbf{x}}) = \int_{\Pi} \Psi((I_2(\mathbf{W}(\mathbf{x}, \dot{\mathbf{x}})) - I_1(\mathbf{x}))^2) d\mathbf{x}, \tag{3}$$

where Ψ is a penalty function, and the warp $\mathbf{W}$ takes the image point $\mathbf{x}$ in the coordinate frame of I_1, and maps it to another image location $\mathbf{W}(\mathbf{x}, \dot{\mathbf{x}})$ in the coordinate frame of I_2.

3.2. Regularization term with convolution kernel prior

Instead of recovering optical flow from (2), it can be shown that the estimation process can be decoupled into a two-step procedure as [35]:

$$\dot{\mathbf{x}}^{\tau'} - \dot{\mathbf{x}}^{\tau-1} = -\frac{\partial e_{data}(\dot{\mathbf{x}})}{\partial \dot{\mathbf{x}}}, \tag{4}$$

$$\dot{\mathbf{x}}^{\tau} - \dot{\mathbf{x}}^{\tau'} = div\left(\frac{\partial e_{reg}(\nabla \dot{\mathbf{x}})}{\partial \nabla \dot{\mathbf{x}}} \right). \tag{5}$$

In particular, the regularization of the intermediate flow field $\dot{\mathbf{x}}^{\tau'}$ resulted from the data term is equivalent to applying a convolution with a 2D oriented Gaussian kernel G to the intermediate flow field as [33]:

$$\dot{\mathbf{x}}^{\tau} = G * \dot{\mathbf{x}}^{\tau'}. \tag{6}$$

4. Variational depth

The recovery of optical flow in homogeneous regions of an image is ill-posed if only visual cues are utilized. Regularizer is often required to diffuse optical flows from textured regions to homogeneous regions. This fact has been known since the seminal work of Horn and Schunck [15]. With this insight, minimization of the data term $E_{data}(Z)$ (or $E_{data}(\dot{\mathbf{x}})$ for the case of optical flow) is only meaningful at the image positions which are textured. The depth values (or optical flows) at homogeneous regions are indeed diffused from nearby textured regions. Due to the recent research work about the two-step variational functional [35], we are able to perform minimization of $E_{data}(Z)$ sparsely at the selected image positions. Using this strategy, we can significantly reduce the computational time of solving the functional by focusing more on the image positions that provide more useful information to the task.

```
Inputs: A pair of color images ($I_1$, $I_2$) and the
    associated pair of raw depth images ($Z_{D1}$, $Z_{D2}$)
1: Recover camera motion ($\mathbf{t}$, $\mathbf{w}$).
2: Construct image pyramids for the color and depth
   images, and set the initial level $l = 0$ and $Z_1$ = $Z_{D1}$.
3: Propagate $Z_1$ to level $l + 1$.
4: Fuse the depth images $Z_{E1}$ and $Z_{D1}$ (Sect. 4.5).
5: Variational depth calculation
   5.1: Minimize data energy at image locations
        with enough texture (Sect. 4.1).
   5.2: Remove correspondence ambiguity (Sect. 4.2).
   5.3: Regularize depth map (Sect. 4.3).
6: Occlusion-aware refinement (Sect. 4.4).
7: If $l \neq N - 1$, where $N$ is the total number of levels,
   $l = l + 1$, and go to Step 3.
Output: Refined depth image $Z_1$
```

Table 1. Method Overview

Our overall algorithm which is outlined in Table 1 is based on iterative coarse-to-fine processing. The steps are detailed further below.

4.1. Nonlocal data term with motion prior

Consider the RGB-D system undergoes a general camera motion with translation $\mathbf{t} = (t_x, t_y, t_z)^T$ and rotation $\mathbf{w} = (w_x, w_y, w_z)^T$ with respect to the camera center C. Optical flow $\dot{\mathbf{x}}$ at image position $\mathbf{x}$ can be expressed in terms of the camera motion $(\mathbf{t}, \mathbf{w})$ as [22]:

$$\begin{aligned} \dot{\mathbf{x}} = & \begin{pmatrix} -f & 0 & x \\ 0 & -f & y \end{pmatrix} \frac{\mathbf{t}}{Z(\mathbf{x})} \\ & + \begin{pmatrix} \frac{xy}{f} & -(\frac{x^2}{f} + f) & y \\ \frac{y^2}{f} + f & -\frac{xy}{f} & -x \end{pmatrix} \mathbf{w}, \end{aligned} \tag{7}$$

where f is the focal length of the RGB camera, and $Z(\mathbf{x})$ is the scene depth at $\mathbf{x}$.

The transformation $H(\mathbf{t}, \mathbf{w})$ between the two RGB-D frames can be recovered by aligning the two associated 3D point clouds using iterative-closest-point (ICP) matching [6]. Since the performance of ICP matching may be affected if the two point clouds are not close enough, we initialize an estimate of the transformation H_0 by using the feature correspondences in the RGB images. Distinct feature points are matched across the two RGB images by using scale-invariant feature transform (SIFT) [23], and then H_0 is computed using 8-point RANSAC [13]. If a binocular imaging system is available, a stereo image pair resembles the two consecutive images. A predefined transformation H between the two camera frames can be easily obtained through camera calibration.

We can express the warp $\mathbf{W}$ in (3) as $\mathbf{x}+\dot{\mathbf{x}}$ for the image frame I_2 towards the frame I_1 using the parametric form in (7). Since the camera motion parameters $(\mathbf{t}, \mathbf{w})$ have been recovered, we can reduce the 2D variational problem for optical flow $\dot{\mathbf{x}}$ to the 1D variational problem for depth Z. Our data term at image position $\mathbf{x}$ is defined as follows:

$$\begin{aligned} &E_{data}\left(Z(\mathbf{x})\right) \\ &= \int_{\Omega} \omega(\mathbf{x}, \mathbf{x}') \bigg(\Psi\left(\left(I_1(\mathbf{x}') - I_2\left(\mathbf{W}(\mathbf{x}', Z(\mathbf{x}))\right)\right)^2 \right) \\ &\quad + \alpha \Psi\left(\left(\nabla I_1(\mathbf{x}') - \nabla I_2\left(\mathbf{W}\left(\mathbf{x}', Z(\mathbf{x})\right)\right)\right)^2 \right) \bigg) d\mathbf{x}', \end{aligned} \tag{8}$$

with some regularization parameter $\alpha > 0$. The above formulation is related to the local method proposed by Lucas and Kanade [24], but we include the contributions of image gradient constancy and nonlocal weight $w(\mathbf{x}, \mathbf{x}')$ to it. Unlike the works by Werlberger *et al.* [34] and Krähenbühl *et al.* [19], we utilize the coherence of neighboring pixels over a window Ω on the data term but not on the regularization term. Moreover, we only perform minimization of E_{data} on the image positions that have enough texture. These positions are detected from the magnitude of spatial intensity gradient (or the eigenvalues of the local image structure tensor). It should be noted that we have only considered a sparse depth image up to this moment.

The value of the nonlocal weight $\omega(\mathbf{x}, \mathbf{x}')$ depends on of the spatial distance, intensity difference, and intensity gradient difference between the pixels at $\mathbf{x}$ and $\mathbf{x}'$ as follows:

$$\begin{aligned} \omega(\mathbf{x}, \mathbf{x}') = \exp\bigg(&-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{2\sigma_s^2} - \frac{\|I_1(\mathbf{x}) - I_1(\mathbf{x}')\|^2}{2\sigma_c^2} \\ &- \frac{\|\nabla I_1(\mathbf{x}) - \nabla I_1(\mathbf{x}')\|^2}{2\sigma_g^2} \bigg), \end{aligned} \tag{9}$$

where σ_s, σ_c, and σ_g are the standard deviations of the Gaussian kernels for the spatial distance, intensity difference, and intensity gradient difference, respectively. Color images are encoded in CIELAB color space. The weight ω is related to the conventional bilateral filter proposed by Tomasi *et al.* [32], but we emphasize more on the use of visual cues from the textured regions in the image by including a term related to the image gradient. Figure 2 visualizes a nonlocal weight image from the Reindeer sequence of the Middlebury stereo dataset [1]. The figure illustrates the visualization of intermediate weights having areas of size 65 × 65. All the standard deviations σ were set to 11.0. Image pixels which have similar color and intensity gradient as the central pixel lead to higher weights.

In order to make the data term more robust to outliers and convenient to the computation later, we use the Charbonnier norm $\Psi(s^2) = \sqrt{(s^2 + \varepsilon^2)}$ (where ε = 0.001) to approximate the L_1 norm for the penalizer Ψ. To simplify the deviation of the data term, we denote the inverse depth as ρ. In the latter part of the derivation, we will need to compute several gradient terms such as $\nabla I_2(\mathbf{W}(\mathbf{x}, \rho))$. The computation of $\nabla I_2(\mathbf{W}(\mathbf{x}, \rho))$ requires dense flow field due to the approximation of derivatives using discrete filter, and therefore this is not suitable for minimizing data term using sparse depth image (or flow field). Instead of computing $\nabla I_2(\mathbf{W}(\mathbf{x}, \rho))$, we exchange the role of I_1 and I_2 using the inverse additive image alignment as Baker *et al.* [5]. This not only makes the sparse depth image computation possible, but also reduces the computational complexity.

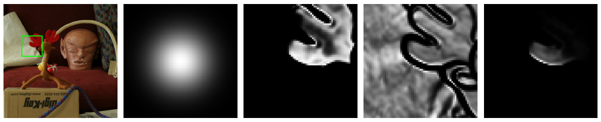

Figure 2. Visualization of nonlocal weight. From left to right: (a) Marked region in the Reindeer dataset, (b) Proximal weight, (c) Color-similarity weight, (d) Gradient-similarity weight, and (e) Combined weight for the marked region.

The derivative of E_{data} in (8) can be expressed as follows:

$$\sum_{\Omega} \omega \left(\Psi'(I_z^2) I_z I_\rho + \alpha \Psi'(I_{xz}^2 + I_{yz}^2)(I_{xz} I_{\rho x} + I_{yz} I_{\rho y}) \right), \tag{10}$$

where

$$\begin{aligned} I_x &= \partial_x I_1(\mathbf{x}), \\ I_y &= \partial_y I_1(\mathbf{x}), \\ I_z &= I_1(\mathbf{x}) - I_2(\mathbf{W}(\mathbf{x}, \rho)), \\ I_{xz} &= \partial_x I_1(\mathbf{x}), \\ I_{yz} &= \partial_y I_1(\mathbf{x}), \\ I_\rho &= \nabla I_1^T \left(\frac{\partial \mathbf{W}}{\partial \mathbf{x}} \right)^{-1} \frac{\partial \mathbf{W}}{\partial \rho}, \\ I_{\rho x} &= \nabla I_x^T \left(\frac{\partial \mathbf{W}}{\partial \mathbf{x}} \right)^{-1} \frac{\partial \mathbf{W}}{\partial \rho}, \\ I_{\rho y} &= \nabla I_y^T \left(\frac{\partial \mathbf{W}}{\partial \mathbf{x}} \right)^{-1} \frac{\partial \mathbf{W}}{\partial \rho}. \end{aligned} \tag{11}$$

For the simplicity of presentation, the variables $\mathbf{x}$ and $\mathbf{x}'$ for some of the above terms are dropped.

With iteration variable ρ^k instead of ρ, ρ^{k+1} can be obtained as the solution of:

$$\begin{aligned} &\sum_{\Omega} \omega (\Psi'((I_z^{k+1})^2) I_z^{k+1} I_\rho^k \\ &\quad + \alpha \Psi'((I_{xz}^{k+1})^2 + (I_{yz}^{k+1})^2)(I_{xz}^{k+1} I_{\rho x}^k + I_{yz}^{k+1} I_{\rho y}^k)) = 0. \end{aligned} \tag{12}$$

In order to remove the nonlinearity in I_*^{k+1}, we approximate I_*^{k+1} by using the first order Talyor expansion as:

$$\begin{aligned} I_z^{k+1} &\approx I_z^k + I_\rho d_\rho^k, \\ I_{xz}^{k+1} &\approx I_{xz}^k + I_{\rho x} d_\rho^k, \\ I_{yz}^{k+1} &\approx I_{yz}^k + I_{\rho y} d_\rho^k, \end{aligned} \tag{13}$$

where $d\rho_{k+1} = \rho_k - d\rho_k$. Here ρ_k is the solution from the previous iteration, and $d\rho_k$ is the unknown increment.

For the better readability, we define

$$\begin{aligned} (\Psi')_c^k &= (\Psi')^k \left((I_z^k + I_\rho d_\rho^k)^2 \right), \\ (\Psi')_g^k &= (\Psi')^k \left((I_{xz}^k + I_{\rho x} d_\rho^k)^2 + (I_{yz}^k + I_{\rho y} d_\rho^k)^2 \right). \end{aligned} \tag{14}$$

We further remove the nonlinearity in $(\Psi')_*$ by performing an inner iteration loop. We initialize $d\rho^{k,0} = 0$, and denote $d\rho^{k,l}$ the iteration variable at step l. The solution for $d\rho^{k,l+1}$ is given by:

$$\begin{aligned} & d\rho^{k,l+1} \\ & = \frac{-\sum \omega \left((\Psi')_c^{k,l} I_\rho^k I_z^k + \alpha (\Psi')_g^{k,l} (I_{\rho x}^k I_{xz}^k + I_{\rho y}^k I_{yz}^k) \right)}{\sum \omega \left((\Psi')_c^{k,l} (I_\rho^k)^2 + \alpha (\Psi')_g^{k,l} \left((I_{\rho x}^k)^2 + (I_{\rho y}^k)^2 \right) \right)}. \end{aligned} \tag{15}$$

4.2. Ambiguity compensated data term

The constraints which are used in the data term E_{data} in (8) are the brightness and intensity gradient constancies of the same feature point across I_1 and I_2. However, there is a degenerate case which is ill-posed to the system in (8) or other state-of-the-art methods without considering regularization. If the truth optical flow is indeed along the iso-brightness contour, ambiguity of matching of the corresponding image points can occur. No matter what the amount of the magnitude of the flow is, the associated data energy is the same. Fortunately, this can be well detected by applying the following detector:

$$\cos^{-1}\left(\hat{\dot{\mathbf{x}}} \cdot \mathbf{n} \right) \in [\frac{\pi}{2} \pm \eta], \tag{16}$$

where $\mathbf{n}$ is the unit vector of the intensity gradient, $\hat{\dot{\mathbf{x}}}$ is the unit vector of $\dot{\mathbf{x}}$, and η is a small threshold. Depth values at such image positions are replaced by the associated depth readings from the RGB-D system.

4.3. Regularization for holes filling

Edge-preserving property is the fundamental concern of the appropriate selection of the regularizer in E_{reg}. As presented in Section 3.2, the diffusion tensor can be replaced by a convolution-based diffusion filter. In the literature, bilateral filter [32] is widely used as the diffusion filter for optical flow [35] [21] [34] [19], scene flow [38], and depth map [26] filtering. Due to the space limit, only the bilateral (BL) filter for depth image is presented here:

$$Z_{BL} = \frac{\int_\Pi Z(\mathbf{x}') g_s(\mathbf{x} - \mathbf{x}') g_c(\|I(\mathbf{x}) - I(\mathbf{x}')\|^2) d\mathbf{x}'}{\int_\Pi g_s(\mathbf{x} - \mathbf{x}') g_c(\|I(\mathbf{x}) - I(\mathbf{x}')\|^2) d\mathbf{x}'}, \tag{17}$$

where g_s and g_c are the two Gaussian functions for spatial and intensity domains, respectively.

The raw depth images from the depth sensor generally contains numerous holes which are assigned with zero depth values. If we filter the depth images using the BL filter in (17), then any image points which are located near to these holes will have smaller depth values than usual. The normalization is biased because the Gaussian kernel $G_s(\sigma_s)G_c(\sigma_c)$ in the denominator does not take into account of the zero depth values in in the numerator. A remedy to it is to multiply a binary map g_h which indicates the location of image points having zero depth values to the kernel in the denominator. In this way, the Gaussian kernel changes to $G_h G_s(\sigma_s) G_c(\sigma_c)$, and the BL filter becomes:

$$Z_{BL} = \frac{\int_\Pi Z(\mathbf{x}') g_s(\mathbf{x} - \mathbf{x}') g_c(\|I(\mathbf{x}) - I(\mathbf{x}')\|^2) d\mathbf{x}'}{\int_\Pi g_h(Z(\mathbf{x}')) g_s(\mathbf{x} - \mathbf{x}') g_c(\|I(\mathbf{x}) - I(\mathbf{x}')\|^2) d\mathbf{x}'}. \tag{18}$$

Exact implementation of (17) or (18) requires $O(N)$ operations per pixel, where N is the kernel size. The bottle neck of the running time for solving the functional is due to the regularization. Instead of using the exact BL filter, we implement (18) using the framework from the fast $O(1)$ BL filter [8]. We follow the regularization as [31] by performing a 2D median filtering before applying the BL filter.

4.4. Occlusion handling

Different kinds of occlusion often occur in optical flow and also depth map recovery. One common consequence is that no corresponding pixel at I_2 can match the occluded pixel at I_1. Distortion and dragging are often resulted in the warped image $I_2(\mathbf{W}(\mathbf{x}, \rho))$ if no special measure is applied to minimize the data energy E_{data}. As a result, estimation of depth map can also be affected accordingly.

We exclude the occluded pixels from the minimization of E_{data} by introducing an occlusion-aware penalty function to e_{data}. Our algorithm uses a Gaussian function which is related to the combination of flow divergence and pixel projection difference as Sand $et\ al.$ [29].

4.5. Fusion and update framework

At the beginning of solving the variational depth at each pyramid level (from coarse to fine manner), we not only have the depth estimation (Z_E) interpolated from the previous coarser level, but also the depth image (Z_D) from the depth sensor as well. We fuse the two depths together using a confident-weighted sum as follows:

$$Z = w_c Z_D + (1 - w_c) Z_E, \tag{19}$$

where $w_c = \frac{E_E}{E_E+E_D}$, and the simplified energy functional E_* (i.e. E_D or E_E) is defined as:

$$\begin{aligned} E_*(Z_*) &= E_c(Z_*) + E_g(Z_*) \\ &= \Psi\left(\left(I_1(\mathbf{x}) - I_2(\mathbf{W}(\mathbf{x}, Z_*))\right)^2\right) \\ &\quad + \alpha\Psi\left(\left(\nabla I_1(\mathbf{x}) - \nabla I_2(\mathbf{W}(\mathbf{x}, Z_*))\right)^2\right). \end{aligned} \tag{20}$$

5. Experiments

We evaluated the experimental results for the following methods using several real RGB-D sequences:

1. **TVL$_1$+NL**: Optical flow using TV-L_1 with non-local terms by Sun *et al.* [31].
2. **LDOF**: Large-displacement optical flow using descriptor matching by Brox *et al.* [7].
3. **ENCC**: Enhanced correlation-based stereo correspondence with subpixel accuracy by Psaraki *et al.* [28].
4. **Motion-Depth (M-D)**: Our RGB-D depth map enhancement with motion and depth in complement.

It should be noted that depth map can be recovered from **TVL$_1$+NL** or **LDOF** by solving Z from (7) as the camera motion parameters $(\mathtt{t}, \mathbf{w})$ have been recovered.

In the evaluation, all programs were written in MATLAB. An exception is **LDOF** [7] which is a MEX-function converted from C/C++ by the author. For a better comparison, we also included the running time of a MATLAB's **LDOF** [2]. All the programs ran on a Win7 PC with Core 2 CPU and 4GB RAM. We used the same number of image pyramids and inner-loop iterations as **TVL$_1$+NL** but without using the graduated non-convexity (GNU) scheme. For the image pyramid, a downsampling factor of 0.5 was used. The other parameter settings were: $\alpha = 0.8$, $\sigma_s = 4.0$, $\sigma_c = \sigma_g = 11.0$. Gaussian kernels were truncated at 3σ.

In the first experiment, we used two benchmark datasets (`Reindeer` and `Moebius`, with resolution 463 × 370) from the Middlebury stereo dataset [1] to evaluate the performance of several methods, namely **ENCC**, **LDOF**, and **M-D**. Figures 3 shows the color images of the two datasets. In order to simulate the depth maps captured by a low-cost RGB-D system, we downsampled the ground-truth depth maps by one-tenth of the original resolution, and then resize them to the original resolution using bilinear interpolation. We also synthesized an occlusion map for each of the datasets by projecting the ground-truth 3D scene points in the coordinate of the right image frame on the left image plane. All the image pixels that cannot be matched are defined as the occluded pixels. We denote the input depth maps for the experiment with the original and reduced resolution as Z_i and $Z_{i\downarrow}$, respectively. Figure 4 shows the resulted depth maps. Our method **M-D** provides better depth discontinuities. Quantitative comparison which was performed using the metric – root mean square error (RMSE), is summarized in Table 2. The computational time for the various methods is given in Table 3. **M-D** performs better than the others.

Figure 3. Two stereo image pairs from the Middlebury stereo dataset.

	ENCC	**LDOF**	$Z_{i\downarrow}$, **M-D**	Z_i, **M-D**
Reindeer	1.995	1.855	1.977, 0.5048	1.938, 0.4342
Moebius	1.649	1.831	1.715, 0.3321	1.694, 0.2715

Table 2. Quality of the input depth maps for testing and the refined depth maps in terms of RMSE.

	ENCC	**LDOF**	**M-D**
Reindeer	491.6	94.82 [2] (17.70 [7])	31.92
Moebius	531.0	102.7 [2] (18.94 [7])	33.93

Table 3. Computational time (in seconds) for the first experiment.

In the second experiment, we evaluated the performance of the proposed method against three RGB-D datasets (`Kitchen`, `Table`, and `Laboratory`, with resolution 640 × 480) without available ground truths. The `Kitchen` and `Table` sequences were obtained from the multi-view RGB-D object dataset [20]. The `Laboratory` sequence was captured by a Kinect for XBOX in our laboratory. Figure 5 shows the color images of the datasets. Figure 6 shows the depth maps resulted from various methods, namely **TVL$_1$+NL**, **LDOF**, and **M-D**. It can be shown that our method (**M-D**) not only recovers depth maps with sharper depth discontinuities, but also provides more fine scene details. More specifically, we can reconstruct the cupboard door handles for the `Kitchen` sequence. We can also reconstruct the computer cables and the office chair in the `Table` sequence. The cable of the computer mouse in the `Laboratory` sequence can also be reconstructed as well. Other compared schemes cannot provide such fine details of the scene. Table 4 summarizes the computational time for the compared methods (excluding the running time for estimating camera motion as this is the common step). Our method requires lower computational time than the others.

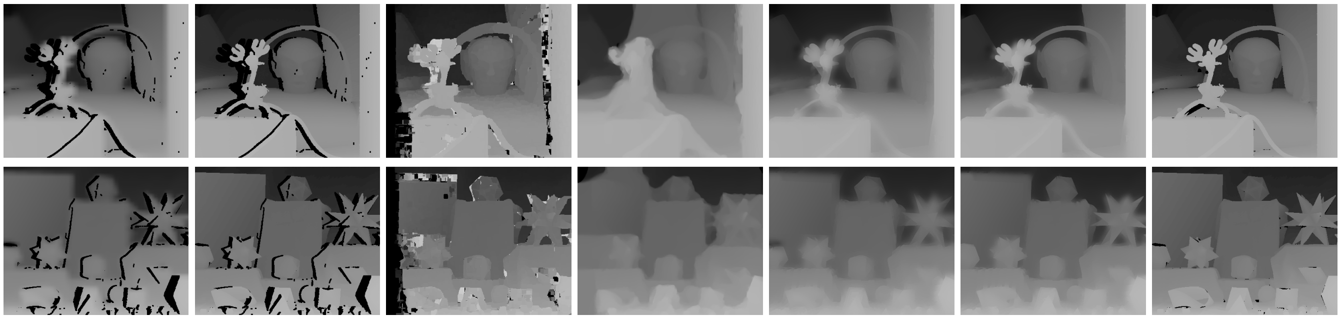

Figure 4. Experimental results on the two stereo pairs from the Middlebury dataset. From left to right: Input depth maps having (a) Reduced and (b) Original resolution. Depth maps resulted from (c) ENCC [28], (d) LDOF [7], (e) M-D using (a) as input, and (f) M-D using (b) as input. (g) Ground truth. For the best visual evaluation, it is recommended to enlarge the figure in the electronic version of this article.

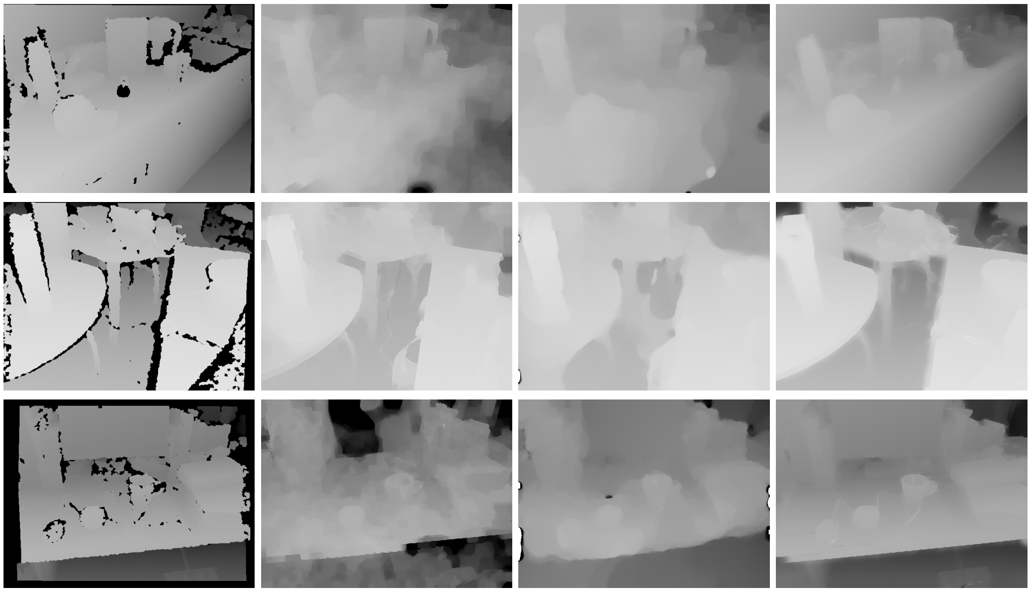

Figure 6. Experimental results on the three RGB-D sequences captured by Kinects. From left to right: Depth maps from (a) Kinect, (b) TVL$_1$+NL [31], (c) LDOF [7], and (d) Proposed motion-depth (M-D) method. For the best visual evaluation, it is recommended to enlarge the figure in the electronic version of this article.

	TVL$_1$+NL	LDOF	M-D
Kitchen	187.6	204.4 [2] (40.56 [7])	43.58
Table	153.8	192.7 [2] (38.34 [7])	48.61
Laboratory	198.0	187.4 [2] (38.12 [7])	55.71

Table 4. Computational time (in seconds) for the second experiment.

6. Conclusions

We have proposed a variational-based depth map enhancement which naturally fuses the depth maps from the active sensor of a moving RGB-D system and the depth cues from the induced optical flow. Instead of computing the flow field, we recover the depth map directly. We have improved the computational efficiency by treating the textured and homogeneous regions differently. The overall result is that our proposed method has a fast computational time and the ability to recover fine details of the imaged scene.

References

[1] http://vision.middlebury.edu/stereo/. 2013.

[2] http://www.seas.upenn.edu/ katef/ldof. 2013.

[3] Kinect, http://www.microsoft.com/en-us/kinectforwindows. 2013.

[4] SwissRanger, http://www.mesa-imaging.ch. 2013.

[5] S. Baker and I. Matthews. Equivalence and efficiency of image alignment algorithms. *CVPR*, pages 1090–1097, 2001.

[6] P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. *PAMI*, 14(2):239–256, 1992.

Figure 5. Three image sets captured by Kinects.

[7] T. Brox and J. Malik. Large displacement optical flow: Descriptor matching in variational motion estimation. *PAMI*, pages 500–513, 2011.

[8] K. N. Chaudhury. Acceleration of the shiftable O(1) algorithm for bilateral filtering and nonlocal means. *TIP*, 22(4):1291–1300, 2013.

[9] O. Choi and S. Lee. Fusion of time-of-flight and stereo for disambiguation of depth measurements. *ACCV 2012*, pages 640–653, 2013.

[10] A. J. Davison. Real-time simultaneous localisation and mapping with a single camera. *ICCV*, pages 1406–1410, 2003.

[11] V. Gandhi, J. Cech, and R. Horaud. High-resolution depth maps based on TOF-stereo fusion. *ICRA*, pages 4742–4749, 2012.

[12] F. Garcia, D. Aouada, H. K. Abdella, T. Solignac, B. Mirbach, and B. Ottersten. Depth enhancement by fusion for passive and active sensing. *ECCV Workshops*, pages 506–515, 2012.

[13] R. Hartley and A. Zisserman. Multiple view geometry in computer vision. *Cambridge University Press*, 2003.

[14] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox. RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments. *IJRR*, 31(5):647–663, 2012.

[15] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Arifical Intelligence*, 17:185–203, 1981.

[16] T.-W. Hui and R. Chung. Determining spatial motion directly from normal flow field: A comprehensive treatment. *ACCV 2010 Workshops*, pages 23–32, 2011.

[17] T.-W. Hui and R. Chung. Determining motion directly from normal flows upon the use of a spherical eye platform. *CVPR*, pages 2267–2274, 2013.

[18] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. *ISMAR*, pages 225–234, 2007.

[19] P. Krähenbühl and V. Koltun. Efficient nonlocal regularization for optical flow. *ECCV*, pages 2464–2471, 2012.

[20] K. Lai, L. Bo, X. Ren, and D. Fox. A large-scale hierarchical multi-view RGB-D object dataset. *ICRA*, pages 1817–1824, 2011.

[21] K. J. Lee, I. D. Y. D. Kwon, and S. U. Lee. Optical flow estimation with adaptive convolution kernel prior on discrete framework. *CVPR*, pages 2504–2511, 2010.

[22] H. C. Longuet-Higgins and K. Prazdny. The interpretation of a moving retinal image. *Proc. Royal Society London B*, 208(1173):385–397, 1980.

[23] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[24] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. *IJCAI*, pages 674–679, 1981.

[25] R. A. Newcombe and A. J. Davison. Live dense reconstruction with a single moving camera. *CVPR*, pages 1498–1505, 2010.

[26] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. *ISMAR*, pages 91–97, 2011.

[27] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. DTAM: Dense tracking and mapping in real-time. *ICCV*, pages 2320–2327, 2011.

[28] E. Z. Psarakis and G. D. Evangelidis. An enhanced correlation-based method for stereo correspondence with sub-pixel accuracy. *ICCV*, pages 907–912, 2005.

[29] P. Sand and S. Teller. Particle video: Long-range motion estimation using point trajectories. *IJCV*, 8(1):72–91, 2008.

[30] J. Stühmer, S. Gumhold, and D. Cremers. Real-time dense geometry from a handheld camera. *DAGM Sym. on Pattern Recognitiom*, pages 11–20, 2010.

[31] D. Sun, S. Roth, and M. J. Black. Secrets of optical flow estimation and their principles. *CVPR*, pages 2432–2439, 2010.

[32] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. *ICCV*, pages 839–846, 1998.

[33] D. Tschumperle and R. Deriche. Vector-valued image regularization with PDE's: A common framework for different applications. *CVPR*, pages 651–656, 2003.

[34] M. Werlberger, T. Pock, and H. Bischof. Motion estimation with non-local total variation regularization. *CVPR*, pages 2464–2471, 2010.

[35] J. Xiao, H. Cheng, H. Sawhney, C. Rao, and M. Isnardi. Bilateral filtering-based optical flow estimation with occlusion detection. *ECCV*, pages 211–224, 2006.

[36] Q. Yang. A non-local cost aggregation method for stereo matching. *CVPR*, pages 1402–1409, 2012.

[37] Q. Yang, K.-H. Tan, B. Culbertson, and J. Apostolopoulo. Fusion of active and passive sensors for fast 3D capture. *MMSP*, pages 69–74, 2010.

[38] X. Zhang, D. Chen, Z. Yuan, and N. Zheng. Dense scene flow based on depth and multi-channel bilateral filter. *ACCV 2012*, pages 140–151, 2013.

[39] J. Zhu, L. Wang, R. Yang, and J. Davis. Fusion of time-of-flight depth and stereo for high accuracy depth maps. *CVPR*, 2008.

[40] H. Zimmer, A. Bruhn, and J. Weickert. Optic flow in harmony. *IJCV*, 93:368–388, 2011.