# EDGE-PRESERVING DISPARITY MAP ESTIMATION FROM STEREO VIDEOS FOR BOKEH SYNTHESIS

*Wei-Lun Lan, Shih-Hsuan Yao, and Shang-Hong Lai*
Dept. of Computer Science, National Tsing Hua University, Hsinchu, Taiwan
william_sky77@hotmail.com, dachshund_dog@hotmail.com, lai@cs.nthu.edu.tw

## ABSTRACT

We present a new method of estimating disparity maps from stereo videos for bokeh effect synthesis. In this work, we develop an improved total variation regularization and the robust $L^1$ norm in the data fidelity term (TV-$L^1$) [4] based method to estimate edge-preserving disparity map without stereo rectification. The proposed algorithm improves the TV-$L^1$ approach by incorporating structure edge detection, occlusion area detection, textureless region detection and applying the guided filter to alleviate the inconsistency problem between the disparity map and color image around object boundary. Furthermore, we propose a temporal filter to improve the temporal consistency of the disparity maps computed from the stereo videos. We use saliency map to focus the synthesis result on the objects which attract human attention most. Experimental comparisons on various real videos are shown to demonstrate that the proposed algorithm generates more visually pleasing bokeh video synthesis compared with those by using previous stereo matching methods.

***Index Terms***— Optical flow, stereo matching, depth-of-field synthesis

## 1. INTRODUCTION

Recently, some companies produced smartphones equipped with stereo cameras [9]. These cameras can generate deep depths-of-field and take all-in-focus stereo images. With stereo images taken from the scene, either stereo matching or optical flow estimation can be used to estimate dense disparity map, which can be used for depth-related applications, such as bokeh effect synthesis or 3D scene display.

Stereo vision has been widely researched in the past decades. Traditional steps of depth estimation include camera calibration, stereo rectification and stereo matching. Stereo matching is one of the most extensively researched topics in computer vision and there exist many state-of-the-art methods. Input stereo images for stereo matching are usually assumed to be well-rectified. This assumption needs very accurate camera calibration in advance. After camera calibration, the calibrated camera parameters are used to rectify stereo images, followed by stereo matching.

Our goal is to estimate edge-preserving disparity maps from stereo videos to synthesize bokeh videos. In consideration of fast computation, we adopt the optical flow approach to estimate approximate disparity maps from uncalibrated stereo videos. Edges in the disparity map of each frame should well align the edges in the color image because human eyes are sensitive to artifacts in the bokeh image near object boundaries. Moreover, the temporal consistency of the estimated temporal disparity maps is another important issue. Two neighboring frames may estimate two different disparity maps when the objects in the scene is almost static. The defocus effect in two consecutive frames will not be consistent, thus causing visual artifacts. Finally, we propose a bokeh synthesis method for synthesizing depth-of-field effect from stereo videos. Our experimental results show that the proposed method can generate visually pleasing depth-of-field synthesis results by using the estimated disparity maps.

## 2. PROPOSED METHOD

This section starts with the pre-processing method of our framework in section 2.1. Our proposed TV-$L^1$ optical flow method will be introduced in section 2.2 to 2.8. Some post-processing steps are described in section 2.9. A defocus rendering method is presented in section 2.10. Figure 1 illustrates the flowchart of the proposed system.

### 2.1. Pre-processing method

In our proposed framework, the stereo image pairs do not have to be rectified. However, simple alignment for stereo pairs is beneficial to the estimation of disparity map. We apply a feature correspondence based method to align the left and right images. We extract SURF feature points from the left and right images and employ SURF descriptors to match corresponding feature points with SSD distance. Then, RANSAC algorithm is used to estimate the best scaling factor and translation vector with a minimum cost to formulate the transformation matrix:

$$\begin{bmatrix} s & 0 & t_x \\ 0 & s & t_y \\ 0 & 0 & 1 \end{bmatrix}.$$

This work was partially supported by Qualcomm Technologies, Inc.

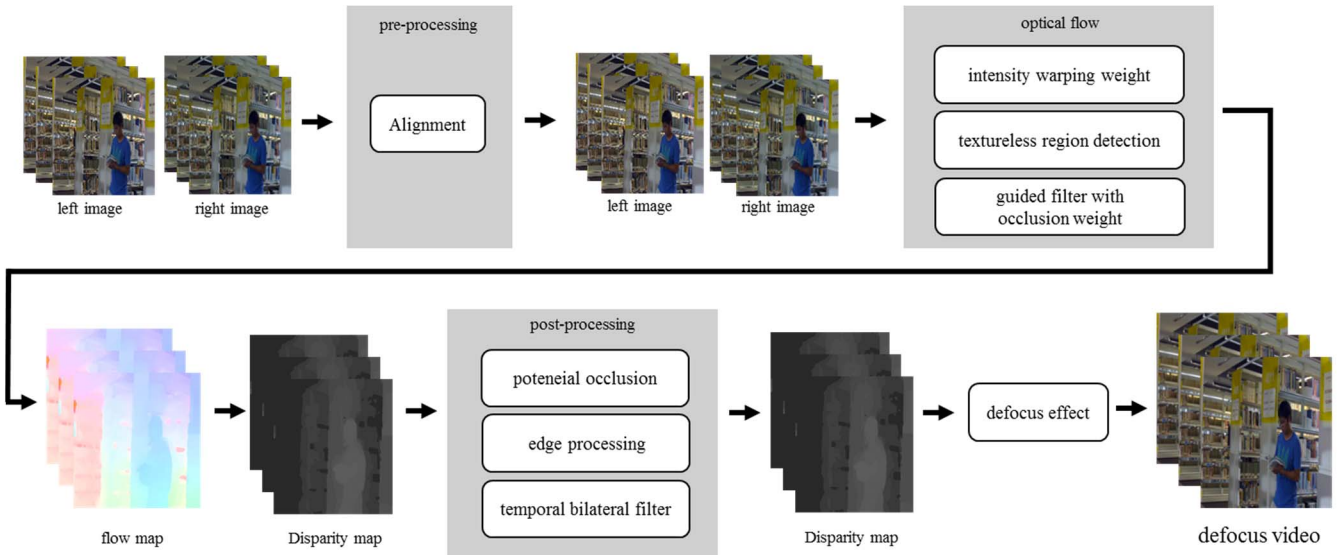


**Figure 1.** The flow chart of our proposed system

Finally, we use this transformation matrix to warp all pixels in the right image $I_R$ to obtain the transformed right image $I'_R$. Then the y components of the displacements field between the left and right images are very small and close to zero. The x components of displacements field are kept as disparity values. One thing to be mentioned, our work focus on stereo video sequences. We only estimate the transformation matrix once in the beginning of the video sequence and use the same transformation matrix to the subsequent frames for simple alignment.

### 2.2. TV-L[1] optical flow

Optical flow computation is based on brightness constancy assumption. The TV-L[1] optical flow formulation [4] involves minimizing the following energy function:

$$E_{TV-L_1} = \sum_\Omega(|\nabla u_1| + |\nabla u_2|) + \lambda|\rho(\boldsymbol{u})| \quad (1)$$

$$\rho(\boldsymbol{u}) = I_R(\boldsymbol{x}+\boldsymbol{u_0}) + (\boldsymbol{u}-\boldsymbol{u_0})\nabla I_R(\boldsymbol{x}+\boldsymbol{u_0}) - I_L(\boldsymbol{x}) \quad (2)$$

where $\boldsymbol{u}(\mathrm{x},\mathrm{y}) = \big(u_1(x,y),\, u_2(x,y)\big)^T$ and $\boldsymbol{u_0}$ is a close approximation to $\boldsymbol{u}$.

### 2.3. Intensity warping weight

In the data term, every pixel shares the same parameter λ, which means each pixel is imposed with the same data constraint. Here is an intensity warping weight,

$$W_I = e^{-\frac{(I_1(x+u)-I_0(x))^2}{2*\sigma_c^2}}. \quad (3)$$

where $(I_R(x+u)-I_L(x))^2$ is the squared difference between the corresponding intensities. We combine this term into (1), and the energy term can be modified as:

$$E'_{TVL_1} = \sum_\Omega(|\nabla u_1| + |\nabla u_2|) + \lambda W_I|\rho(\boldsymbol{u})|. \quad (4)$$

We find that when the absolute difference of intensity is large, it often occurs in occlusion regions and small regions near large objects. These regions usually correspond to miscalculated flow vectors. After adding $W_I$, the miscalculated regions become smoother and avoid outliers in flow field and disparity map.

### 2.4. Textureless region detection

Computing optical flow field in textureless regions is doomed to larger errors, since pixels in these regions may correspond to wrong pixels with similar intensities and gradients. We first detect the regions where the gradient magnitude averaged over a square window $N_w$ of a given size is below a given threshold. Then, we define the textureless weight as follows:

$$\begin{cases} W_{tex\_less} = 1 - e^{\frac{-|g_{N_w}|*|g_{N_w}|}{2*\sigma^2_{tex\_less}}}, & if\ |g_{N_w}| < threshold^2_{tex\_less} \\ W_{tex\_less} = 1, & if\ |g_{N_w}| \geq threshold^2_{tex\_less}. \end{cases} \quad (6)$$

where

$$|g_{N_w}| = \frac{\sum_{i\in N_w}\sqrt{g_{ix}^2+g_{iy}^2}}{\|N_w\|}. \quad (5)$$

The textureless weight means the pixels with small image gradients should lower the influence of their data term to avoid propagating miscalculated flow field. We combine this term into (4), and the energy term can be modified as:

$$E''_{TVL_1} = \sum_\Omega\big(|\nabla u_1| + |\nabla u_2|\big) + \lambda W_I W_{tex\_less}|\rho(\boldsymbol{u})|. \quad (7)$$

### 2.5. Structure edge weight

To alleviate the problem of optical flow smoothness constraint across edge boundary, we impose additional structure edge weight to the smoothness term of our energy

function of optical flow computation to avoid propagating over object boundaries. For the structure edge weight, we use the structure edge detector (SED) proposed by Dollár et al. [12] and compute the weight of SED model, which is denoted by $D_{SED}$. Thus, the energy function becomes

$$E'''_{TVL_1} = \sum_\Omega D_{SED}(|\nabla u_1| + |\nabla u_2|) + \lambda W_I W_{tex\_less}|\rho(\boldsymbol{u})|. \quad (8)$$

### 2.6. Numerical Optimization

Minimizing the energy function in eq. (8) can be efficiently accomplished by using the following convex approximation given in eq. (9) with an auxiliary variable $\boldsymbol{v}$:

$$E_\theta = \sum_\Omega D_{SED}(|\nabla u_1| + |\nabla u_2|) + \frac{1}{2\theta}|\boldsymbol{u} - \boldsymbol{v}|^2 + \lambda W_I W_{tex\_less}|\rho(\boldsymbol{u})|, \quad (9)$$

where $\theta$ is a small constant, such that $\boldsymbol{v}$ is a close approximation of $\boldsymbol{u}$. This minimization problem of convex function $E_\theta$ can be optimized by fixing one of $\boldsymbol{u}$ or $\boldsymbol{v}$ and solving for the other one alternatively in every iteration.

1. Fixed $\boldsymbol{v}$, solve
$$\min_{\boldsymbol{u}} \sum_\Omega D_{SED}(|\nabla u_1| + |\nabla u_2|) + \frac{1}{2\theta}|\boldsymbol{u} - \boldsymbol{v}|^2. \quad (10)$$
2. Fixed $\boldsymbol{u}$, solve
$$\min_{\boldsymbol{v}} \sum_\Omega \frac{1}{2\theta}|\boldsymbol{u} - \boldsymbol{v}|^2 + \lambda W_I W_{tex\_less}|\rho(\boldsymbol{v})|. \quad (11)$$

The first sub-problem is the total variation based image denoising model of [10] and it can be efficiently solved by using the optimization steps proposed in [11]. The first solution of (10) is given by

$$\boldsymbol{u}^{k+1} = \boldsymbol{v}^{k+1} + \theta \text{ div } (D_{SED}\boldsymbol{p}^k), \quad (12)$$

where $\boldsymbol{p} = (p_1, p_2)$ is a dual vector which can be obtained by computing the fixed point of the following iteration,

$$\boldsymbol{p}^{k+1} = \frac{\boldsymbol{p}^k + \tau/\theta \nabla(\theta div(D_{SED}\boldsymbol{p}^k) + \boldsymbol{v}^{k+1})}{1 + \tau/\theta |\nabla(\theta div(D_{SED}\boldsymbol{p}^k) + \boldsymbol{v}^{k+1})|}, \quad (13)$$

where $k$ is the iteration number, $\boldsymbol{p}^0 = 0$, and $\tau$ is the time step. The second sub-problem can be solved by thresholding in a point-wise manner because it does not rely on spatial derivatives of $\boldsymbol{v}$ and it can be minimized by

$$\boldsymbol{v}^{k+1} = \boldsymbol{u}^{k+1} + \boldsymbol{TH}(\boldsymbol{u}^{k+1}, \boldsymbol{u}^0), \quad (14)$$

$$\boldsymbol{TH}(\boldsymbol{u}^{k+1}, \boldsymbol{u}^0) = \begin{cases} \lambda\theta W_I W_{tex_{less}} \nabla I_1(\boldsymbol{x} + \boldsymbol{u}^0) & , if\ \rho(\boldsymbol{u}^{k+1}, \boldsymbol{u}^0) < -\lambda\theta W_I W_{tex\_less}|\nabla I_1(\boldsymbol{x} + \boldsymbol{u}^0)|^2 \\ -\lambda\theta W_I W_{tex_{less}} \nabla I_1(\boldsymbol{x} + \boldsymbol{u}^0) & , if\ \rho(\boldsymbol{u}^{k+1}, \boldsymbol{u}^0) > \lambda\theta W_I W_{tex\_less}|\nabla I_1(\boldsymbol{x} + \boldsymbol{u}^0)|^2 \\ -\rho(\boldsymbol{u}^{k+1}, \boldsymbol{u}^0) W_I W_{tex_{less}} \frac{\nabla I_1(\boldsymbol{x}+\boldsymbol{u}^0)}{|\nabla I_1(\boldsymbol{x}+\boldsymbol{u}^0)|^2} & , if\ |\rho(\boldsymbol{u}^{k+1}, \boldsymbol{u}^0)| \le \lambda\theta W_I W_{tex\_less}|\nabla I_1(\boldsymbol{x} + \boldsymbol{u}^0)|^2 \end{cases} \quad (15)$$

### 2.7. Edge-preserving filter

Guided filter [13] solves the problem of the gradient reversal artifacts and it is more efficient than the bilateral filter. It involves using a kernel filter:

$$W_{ij}^{gf}(I) = \frac{1}{|\omega|^2}\sum_{k:(i,j)\in\omega_k}\left(1 + \frac{(I_i - \mu_k)(I_j - \mu_k)}{\sigma_k^2 + \epsilon}\right), \quad (16)$$

where $\mu_k$ and $\sigma_k^2$ are the mean and variance of guided image $I$ in window $\omega_k$, $|\omega|$ is the number of pixels in $\omega_k$ and $j$ is the neighbor pixel of pixel $i$ in window $\omega_k$.

### 2.8. Occlusion weight

The previous work [14] developed an occlusion region detection method to handle the correspondence problem in these areas. It provides two cues: the divergence of the motion field is negative for occluded boundaries and the intensity difference of pixel warping. The occlusion weight is defined by combining the modified divergence and intensity difference by using a zero-mean Gaussian distribution leads to:

$$r(x, y) = N(d(x, y); \sigma_d) \cdot N(e(x, y); \sigma_e). \quad (17)$$

where $d(x, y)$ is negative divergence, and $e(x, y)$ is the intensity difference. We normalize the occlusion weight $r$ such that its sum is 1, i.e.

$$W_{ij}^{occ}(I) = \frac{r_j}{\sum_j r_j}. \quad (18)$$

Here, $W_{ij}^{occ}$ is the normalized occlusion weight. We combine this term into the guided filter kernel as follows:

$$q_i = \frac{\sum_j\left((W_{ij}^{gf}(I) W_{ij}^{occ}(I)) p_j\right)}{\sum_j\left(W_{ij}^{gf}(I) W_{ij}^{occ}(I)\right)}. \quad (19)$$

where $q_i$ is the output of guided filter, and $p_j$ is the input motion field. Equation (19) is normalized again to ensure the sum of weights is 1.

### 2.9. Post-processing procedure

The x component of optical flow field can be used as the disparity map. We only handle the disparity map to make it aligned with object boundaries for defocus rendering

#### *2.9.1. Potential occlusion*

Occlusion is a common problem in estimating disparity map. Although occlusion weight is added in our TV-$L^1$ optical flow method, there are some occlusion regions remained to be handled. Instead of using the typical L/R check method, we employ another method to find the potential occlusion regions and avoid calculating the optical flow twice.

Occlusion normally locates near the edge, exactly when the disparity difference between both sides of the vertical edge is large. We detect potential occlusion regions where the difference of average disparities for the two sides is larger than a threshold. Once the occlusion regions are detected, cross-region voting [8] is used to update the wrong disparity first. For a pixel $p$ in the potential occlusion region, a histogram is built by collecting the remaining reliable disparities in its cross-region window with $d_{\max + 1}$ bins. The total number of the reliable pixels is denoted as $S_p = \sum_{d=0}^{d=d_{max}} H_p(d)$. The disparity with the most votes is denoted

| | TVL1[4] | EPPM[6] | Drules[5] | Ours | Ours* | LIBELAS[1] | SSCA[3] | SPS-StFl[2] |
|---|---|---|---|---|---|---|---|---|
| 041 | 2 | 0 | 10 | 5 | **21** | 0 | 0 | 0 |
| 042 | 9 | 1 | 10 | **13** | 5 | 0 | 0 | 0 |
| 044 | 10 | 0 | 1 | **17** | 8 | 1 | 0 | 1 |
| 0102 | 3 | 0 | 4 | 1 | **26** | 3 | 0 | 1 |
| 0104 | 3 | 2 | 4 | **19** | 10 | 0 | 0 | 0 |
| 0109 | 0 | 0 | 1 | 2 | **18** | 12 | 0 | 5 |
| Mean $\pm$ std | $4.5 \pm 4.0$ | $0.5 \pm 0.8$ | $5.0 \pm 4.1$ | $9.3 \pm 8.0$ | $16.0 \pm 8.5$ | $2.7 \pm 4.7$ | $0.0 \pm 0.0$ | $1.2 \pm 1.9$ |

**Table 1.** The results of our user study for video sequences. Ours* is our optical flow method with post processing. The last row is the mean and standard deviation of the vote received by each method for the 6 videos.

by $d_p^*$. The disparity of pixel $p$ is then updated with $d_p^*$ if the ratio of most-vote disparity and total number of reliable disparities is over a threshold $\tau_H$ and the number of most votes disparity is larger than the threshold $\tau_S$.

*2.9.2. Edge processing*

In this step, we further improve the remaining disparity edges which do not align well with object boundaries, especially the edges with very large gradient magnitudes. We modify the post-processing method called RADAR [7], and use joint bilateral filter for refinement. Edges of disparity map are detected by Canny edge detection with threshold to keep strong edges and SED model for detecting color edges. Then mismatched edges are found by checking whether the edges of disparity map coincide with the boundaries of objects in the scene. For the mismatched edges, we adapt the inconsistent region detection method in [7] to detect the inconsistent regions, and we apply the joint bilateral filter to refine these regions.

*2.9.3. Temporal bilateral filter*

We smooth the disparity map by the color and disparity similarity. If the temporal smoothing is not performed, flickering artifact is obvious. The temporal bilateral filter (TBF) is given by:

$$\bar{d}_t = \frac{\sum_{k=0}^{num} W_{d_{t-k}} * W_{color_{t-k}} * d_{t-k}(x_{t-k}, y_{t-k})}{\sum_{k=0}^{num} W_{d_{t-k}} * W_{color_{t-k}}}, \quad (20)$$

$$W_{d_{t-k}} = e^{\left(\frac{-\|d_{t-k}-\hat{d}\|_2}{2\sigma_{td}^2}\right)}, W_{color_{t-k}} = e^{\left(\frac{-\|color_{t-k}-\widehat{color}\|_2}{2\sigma_{tr}^2}\right)}. \quad (21)$$

where $d$ is the disparity, $t$ is the index of the frame and $num$ is the numbers of frame for TBF.

**2.10. Defocus rendering**

Our own rendering algorithm to generate depth-of-field video involves defocus synthesis. Disparity map is divided into N layers. We use [16] to obtain the saliency map from our disparity map, and choose the disparity which has the largest quality of saliency pixels as the 1st layer's median disparity. Remaining N-1 layers' median disparities are chosen equidistantly according to the number of layers and the range of disparity map in each frame. We use each layer's median disparity to determine the range of disparity in each layer. We generate one clear color image and N-1 different blurred color images by using the guided filter, and the level of blur is related to the median disparity difference to the 1st layer. The pixels with disparity values belong to the 1st layer (foreground region) are displayed with clear color image. The remaining pixels are computed by using linear interpolation with N-1 different blurred color images and one clear image. The blur applied to the remaining pixels is proportional to the difference between the disparities of the pixel and their mean disparity values of the two nearest layers.

## 3. EXPERIMENTAL RESULTS

The evaluation of the proposed algorithm is performed in three ways. First, the defocus effect synthesis results were evaluated by user study which involved asking users to choose the most visually pleasing and comfortable defocus synthesis results generated from eight different disparity estimation methods. Second, the average of error pixels for different methods were compared. Third, the SSIM values [15] between the estimated disparity maps and ground truth were evaluated. We depict some example results by using the proposed algorithm in Figure 2.

The dataset used in our experiments was collected by acquiring stereo videos with a stereo camcorder. There are 6 videos with almost 300 frames in each sequence, with 20 frames uniformly distributed for each video labeled with ground truth disparity maps. One region's ground truth disparity value is the horizontal distance between left image and right image when the region appears on two images. We have developed a program to label ground truth conveniently.

In the stereo matching approach, LIBELAS [1], SPS-StFl [2] and SSCA [3] are included into the experimental comparison. In the optical flow approach, TV-L[1] [4], correlation flow [5] and EPPM [6] which emphasizes edge-preserving properties are considered. Our method with and

| | TVL1 [4] | EPPM [6] | Drules [5] | Ours | Ours$^{P}$ | Ours$^{P+E}$ | Ours$^{P+E+T}$ | LIBELAS [1] | SSCA [3] | SPS-StFl [2] |
|---|---|---|---|---|---|---|---|---|---|---|
| 041 | 0.162 | 0.198 | 0.056 | 0.058 | **0.047** | 0.049 | 0.057 | 0.093 | 0.216 | 0.270 |
| 042 | 0.112 | 0.292 | **0.077** | 0.111 | 0.109 | 0.109 | 0.106 | 0.280 | 0.482 | 0.621 |
| 044 | 0.118 | 0.156 | 0.104 | 0.096 | **0.088** | 0.089 | 0.094 | 0.208 | 0.165 | 0.153 |
| 0102 | 0.087 | 0.074 | 0.086 | 0.083 | **0.073** | 0.078 | 0.112 | 0.094 | 0.107 | 0.117 |
| 0104 | 0.102 | 0.109 | 0.081 | 0.073 | **0.063** | 0.066 | 0.108 | 0.184 | 0.280 | 0.384 |
| 0109 | 0.163 | 0.119 | 0.116 | 0.104 | 0.097 | 0.099 | 0.135 | 0.088 | 0.096 | **0.080** |
| mean | 0.124 | 0.158 | 0.087 | 0.088 | **0.080** | 0.082 | 0.102 | 0.158 | 0.224 | 0.271 |

**Table 2.** The average error rate for different methods on 6 stereo videos. Ours denotes the proposed method without all post-processing steps. Ours$^{P}$ stands for the proposed method with potential occlusion. Ours$^{P+E}$ stands for the proposed method with edge-processing. Ours$^{LR+E+T}$ is Ours$^{P+E}$ with temporal bilateral filter.

| | TVL1 [4] | EPPM [6] | Drules [5] | Ours | Ours$^{P}$ | Ours$^{P+E}$ | Ours$^{P+E+T}$ | LIBELAS [1] | SSCA [3] | SPS-StFl [2] |
|---|---|---|---|---|---|---|---|---|---|---|
| 041 | 0.9427 | 0.8838 | 0.9757 | 0.9778 | 0.9844 | 0.9859 | **0.9861** | 0.9275 | 0.9075 | 0.917 |
| 042 | 0.9825 | 0.8729 | **0.9905** | 0.9856 | 0.9864 | 0.9868 | 0.9870 | 0.8785 | 0.8159 | 0.8366 |
| 044 | 0.9709 | 0.9658 | 0.9725 | 0.9752 | 0.9782 | **0.9803** | **0.9803** | 0.9322 | 0.9547 | 0.9711 |
| 0102 | 0.9623 | 0.9765 | 0.9693 | 0.9687 | 0.9747 | **0.9776** | 0.9723 | 0.9598 | 0.9452 | 0.9685 |
| 0104 | 0.9607 | 0.9654 | 0.9708 | 0.9742 | 0.9799 | **0.9823** | 0.9747 | 0.8847 | 0.8713 | 0.8451 |
| 0109 | 0.9270 | 0.9646 | 0.9509 | 0.9452 | 0.9513 | 0.9592 | 0.9544 | 0.9487 | 0.9486 | **0.9729** |
| mean | 0.9577 | 0.9382 | 0.9716 | 0.9711 | 0.9758 | **0.9787** | 0.9758 | 0.9219 | 0.9072 | 0.9185 |

**Table 3**. The average SSIM values for different methods on 6 stereo videos. The meanings of the superscripts of our method are the same as those in Table 2.

without post-processing and the six methods mentioned above are compared with the accuracy of disparity estimation, SSIM values of disparity maps and the bokeh effect synthesis by user study.

### 3.1. User study of defocus rendering results

We invited 38 people who have not been involved in our work to participate in the user study. They were asked to choose the best defocus synthesis results by using different disparity estimation methods for each video. To avoid bias, we randomized the order of 8 algorithms in each presentation for users to choose. Table 1 shows the user study results for different video sequences. The preference is almost the same in first order. Our method with post processing takes the 1st place from the voting by 38 users. The defocus synthesis videos are shown in the supplemental material.

One observation is that our method performs not very well in video 042. Note that three methods have high votes in this case. This is because the objects in this video are steady and the camera motion is very small. On the other hand, we have good performance on video 0102, which contains tiny structures and large displacements. As a result, our method overcomes the two problems of optical flow computation and has better performance than all the others.

### 3.2. Average error rate comparison

We define the bad pixel as those with the deviations between the estimated and ground-truth disparity values larger than 3 pixels. The error rate is defined as the average of bad pixel ratios over 20 frames in each video. The results of error rate comparison for different disparity estimation methods are shown in Table 2.

After performing temporal bilateral filter, the error rate is slightly higher than Ours$^{P+E}$ and Ours$^{P}$. Temporal bilateral filter is applied to preserve the disparity constancy. If the miscalculated regions in the disparity map also appear in the previous frame, the current frame is influenced. However, temporal bilateral filter makes the defocused result more stable and provides better viewing experience.

### 3.3. SSIM comparison

SSIM is a measure for evaluating the similarity between two images from their edge structure. We evaluated average SSIM values for the estimated disparity maps with the ground truth at the 20 frames for each video. The results of the average SSIM values for different disparity estimation methods are shown in Table 3. There is a relationship


**Figure 2**. Example of our disparity results with and without post-processing, and the bokeh syntheses. From left to right: video 0102, video 041, and video 044. The meaning of our and our* is same as Table 1.

between SSIM value and voting of defocus synthesis results. The defocus synthesis results from the estimated disparity map with the highest SSIM value usually has the highest voting. Ours$^{P+E+T}$ has the highest SSIM value in almost all the cases. We think SSIM value is a more suitable measure for evaluating the edge-preserving property in the estimated disparity maps for defocus synthesis than the measure of traditional error rate.

## 4. CONCLUSION

In this work, we propose an improved TV-$L^1$ algorithm for estimating edge-preserving disparity maps for depth-of-field synthesis from unrectified stereo videos. We demonstrate that the proposed algorithm can improve the consistency between disparity discontinuities and object boundaries, and temporal filter can avoid flickering effect in the synthesized bokeh videos. Our experimental comparison shows that the proposed algorithm provides more visually pleasing bokeh videos compared to those generated by using other disparity estimation methods.

## 5. REFERENCES

[1] A. Geiger, M. Roser, and R. Urtasun, "Efficient large-scale stereo matching," *ACCV*, 2010

[2] K. Yamaguchi, D. A. McAllester, and R. Urtasun, "Efficient joint segmentation, occlusion labeling, stereo and flow estimation," *ECCV*, 2014.

[3] K. Zhang, Y. Fang, "Cross-scale cost aggregation for stereo matching," *CVPR*, 2014.

[4] C. Zach, T. Pock, and H. Bischof, "A duality based approach for realtime TV-$L^1$ optical flow," *29th DAGM Symposium on Pattern Recognition*, 2007.

[5] M. Drulea and S. Nedevschi, "Motion estimation using the correlation transform," *IEEE Trans. Image Processing*, 2013.

[6] L. Bao, Q. Yang, and H. Jin, "Fast edge-preserving patchmatch for large displacement optical flow," *IEEE Trans. Image Processing*, 2014.

[7] J. Jiao, R. Wang, and W. Wang, "Local stereo matching with improved matching cost and disparity refinement," *IEEE Int. J. Multimedia*, 2014.

[8] X. Mei et al., "On building an accurate stereo matching system on graphics hardware," *Proc. IEEE Int'l Conf. Computer Vision Workshops*, 2011.

[9] HTC. Htc one-m8. www.htc.com/us/smartphones/htc-one-m8/camera

[10] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, 60:259–268, November 1992.

[11] A. Chambolle, "An algorithm for total variation minimization and applications," *Journal of Mathematical Imaging and Vision*, 20(1-2):89–97, 2004.

[12] P. Dollár and C. Zitnick, "Structured forests for fast edge detection," *ICCV*, 2013

[13] K. He, J. Sun, and X. Tang, "Guided Image Filtering," *ECCV*, 2010.

[14] P. Sand and S. Teller, "Particle video: long-range motion estimation using point trajectories," *IJCV*, 2008

[15] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. on Image Processing*, 2004.

[16] R. Ju, L. Ge, W. Geng, T. Ren, and G. Wu. "Depth saliency based on anisotropic center-surround difference," *IEEE International Conf. on Image Processing*, Paris, France, 2014.