A Semi-Automatic Depth Estimation Method for FTV

(FTV のための半自動奥行き推定方式)

Meindert Onno Wildeboer†, Norishige Fukushima (member)††, Tomohiro Yendo (member)†,
Mehrdad Panahpour Tehrani (member)†, Toshiaki Fujii (member)†††, and Masayuki Tanimoto (member)†

Abstract In this paper, we propose a semi-automatic depth estimation algorithm for Free-viewpoint TV (FTV). The proposed method is an extension of an automatic depth estimation method whereby additional manually created data is input for one or multiple frames. Automatic depth estimation methods generally have difficulty obtaining good depth results around object edges and in areas with low texture. The goal of our method is to improve the depth in these areas and reduce view synthesis artifacts in Depth Image Based Rendering. High-quality view synthesis is very important in applications such as FTV and 3DTV. We define three types of manual input data providing disparity initialization, object segmentation information, and motion information. This data is input as images, which we refer to as manual disparity map, manual edge map, and manual static map, respectively. For evaluation, we used MPEG multi-view videos to demonstrate that our algorithm can significantly improve the depth maps and, as a result, reduce view synthesis artifacts.

Key words: Semi-automatic depth estimation, Depth map, Free-viewpoint Television (FTV), 3DTV, Graph Cuts

1. Introduction

Free-viewpoint view synthesis has gained increasing research interest over the last decade. Multi view video signals are typically captured by an array of synchronized and calibrated cameras, which capture a 3D scene from multiple viewpoints. Virtual viewpoint images can be generated using the multiple views and associated depth data through Depth Image Based Rendering (DIBR)[1]. This enables applications such as Free-viewpoint TV (FTV)[3)~5)] or 3DTV[2], where the user can freely change their viewpoint, and perceive depth. Due to increased popularity of 3D cinema the interest in 3D video applications is growing rapidly. Most current 3D cinema systems are based on stereo images requiring glasses to make the viewer see a different view with each eye[6]. The recent development of auto-stereoscopic displays enables multiple viewers to experience a 3D depth impression without glasses. These type of 3DTV and FTV applications require dense depth maps for photo-realistic image rendering. The goal of our method is to generate depth data accurate enough to enable view-synthesis with no visual artifacts for FTV-type applications. In these applications, the depth is generally generated offline, and the required manual work for generating the manual input data can be part of the production work.

2. Related works

Disparity estimation or stereo matching has been an active research area for many years, and many algorithms are evaluated in[7]. Generally these algorithms can be divided into local (window-based) and global methods[7]. Most of the best performing offline depth estimation algorithms are global methods based on an energy minimization framework[7]. In this framework, the disparity matching is approached as a labeling problem formulated in terms of energy minimization. The energy function contains a data-term and smoothing term as in:

$$E(f_p) = E_{data}(f_p) + E_{smooth}(f_p, f_q) \qquad (1)$$

The data-term E_{data} is a matching cost, indicating how well the label f_p fits pixel p, and is normally derived from the intensity or color differences between the points to be matched. Generally, the disparity of neighboring pixels are piecewise smooth within ob-

Received March 31, 2010; Revised August 12, 2010; Accepted October 5, 2010
† Graduate School of Engineering Nagoya University
(Furo-cho, Chikusa-ku, Nagoya, 464-8603, Japan)
†† Graduate School of Engineering, Nagoya Institute of Technology
(Gokiso-cho, Showa-ku, Nagoya, 466-8555, Japan)
††† Graduate School of Science and Engineering, Tokyo Institute of Technology
(2-12-1 Ookayama, Meguro-ku, Tokyo, 152-8550, Japan)

jects. E_{smooth} is the smoothing term representing the smoothness between pixel p and its neighboring pixel q. The labels corresponding to the disparity of pixel p and q are indicated by f_p and f_q, respectively. The most commonly used non real-time energy minimization algorithms are Belief Propagation[10)11)14)], and Graph Cuts[8)9)]. In our depth estimation method we use the Graph Cuts implementation of Kolmogorov[8)] because it is one of the fastest implementations while obtaining good optimization performance.

One of the main problems in stereo matching is caused by occlusion areas, containing pixels which are visible in one view only. Occlusion occurs at the boundaries of foreground objects, were background pixels are occluded by the foreground object. If more than two camera views are available, occlusions can be handled by using more input views. For example, if we consider three cameras left, center, and right, then pixels occluded in the left camera are normally visible in the right camera. In our method we use three camera views to reduce the problem of occlusions, and we perform matching between the center and left, and center and right camera.

Another problem is caused by image areas which contain little texture. In these areas, all neighboring pixels contain similar color, which causes the matching cost to be nearly constant for all disparity values. As a result, the global minimum energy in those areas does not necessarily yield the correct disparity.

Both occlusion and areas of low texture cause problems for many automatic depth estimation methods to accurately find object boundaries. Recently, segmentation-based stereo approaches (for example [11)~14)]) have gained popularity, as they can reduce the difficulties caused by textureless areas and occlusion, by segmenting the input images into regions of similar colors. Segmentation based methods assume that pixels with similar color have similar disparity, and that there are no large depth discontinuities within each segment. Although these methods can improve the depth in smooth areas, and define clear object boundaries, they tend to cause problems in textured areas. Furthermore, segmentation errors can cause wrong depth boundaries that result in very visible rendering artifacts.

The rest of this paper is organized as follows: We start by a short overview of our method in section 3. 1. Then we outline the manual input data in section 3. 2 followed by the technical details of the energy minimization in section 3. 3. We show the performance of our algorithm in the experiments in section 4, and section 5 concludes our paper.

3. Proposed Algorithm

3. 1 Overview of our method

The semi-automatic depth estimation method presented in this paper is an extension of an automatic depth estimation method based on the energy minimization framework. Although we use Graph Cuts for our energy optimization, the presented method also applies for algorithms based on e.g. Belief Propagation. Our goal is to improve the depth accuracy of the automatic depth estimation, and as a result, reduce view synthesis artifacts. Furthermore, our method includes a temporal propagation algorithm, which helps to reduce the amount of manual work, and improves the temporal consistency of the output depth video.

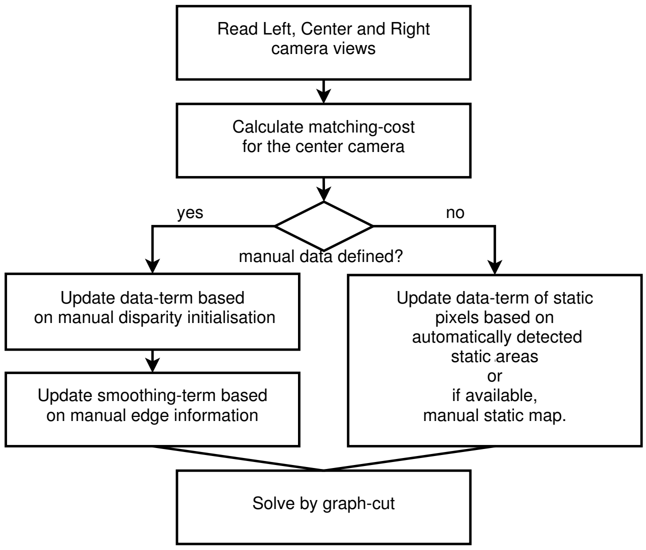


Fig. 1 Simplified flow diagram.

To improve the automatic depth estimation algorithm, additional manually created data is input for one or multiple frames. The three main purposes of this manual input data are: (a) to provide disparity values for areas where automatic depth estimation fails to find an accurate value (e.g. due to little texture, noise, and reflections etc.), (b) to provide object segmentation information, (c) to provide information on static areas. A simplified flow-diagram of our proposed method is depicted in **Fig.1**. Our method can be divided into two main steps, which we denote as *Init Stage*(the left flow in **Fig.1**), and *Temporal Stage*(the right flow in **Fig.1**). In both cases we read three camera views to obtain a depth map for the center view. The matching cost is

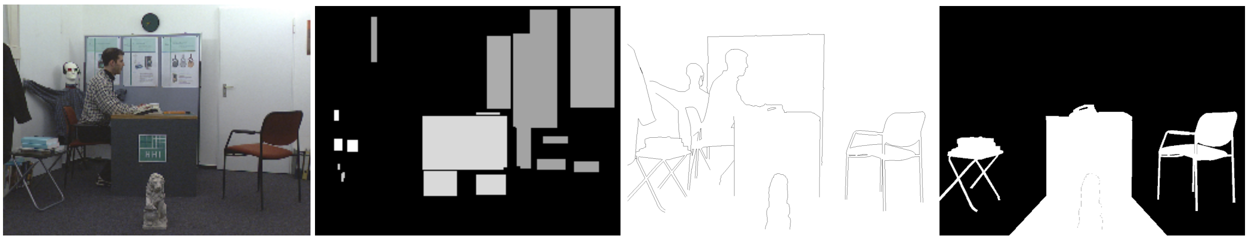


Fig. 2 Left to right: camera center view, manual disparity map, manual edge map, and manual static map.

obtained by matching between the center, and the left- and right-view. In the *Init Stage*, manual disparity initialization and edge information is provided, which is used to update the energy function. The disparity data helps to make the energy data-term more distinctive so the global minimum energy converges to the correct disparity. The edge information is used in the energy smoothing term to cut disparity smoothing at disparity edges. If a scene contains static objects, the depth of these objects remain static over time (assuming the camera is not moving). Therefore, we want to propagate accurate depth obtained in the *Init Stage* into following frames. This results in two benefits, namely: it reduces the amount of manual input data, and it improves the temporal consistency as the depth in static areas is kept constant over time. In the *Temporal Stage*, static areas between the current and previous frame are detected automatically or defined via the manual input data. The energy data-term is updated to propagate depth of static areas from the previous frame to the current frame. Whether pixels moved from one frame to the next is automatically detected based on intensity difference. This automatic motion detection is often difficult if the intensity changes because of e.g. shadow, reflection, or noise. Therefore, our temporal consistency algorithm allows pixels to be manually assigned as static through the manual input data. Finally, a per-pixel disparity map is obtained by solving the energy function using Graph Cuts.

3. 2 Manual input data

As mentioned in the previous section, in our method manually created data is input which provides disparity initialization, object segmentation information, and motion information. This manual input data is input as bitmap images that can be created in any standard image editing software which supports multiple layers. We define three types of manual input data, which can be supplied for any arbitrary frame (please refer to **Fig.2** for an example snapshot):

• *Manual disparity map*: This is a grayscale image containing disparity initialization, for example for areas with low texture, noise, or reflections. A pixel with intensity 10 in the manual disparity map, means that the disparity value for that pixel is initialized in the Graph Cuts energy function to disparity value 10. The Graph Cuts smoothing will propagate the disparity initialization spatially to surrounding low textured areas. This reduces the amount of initialization required in the manual disparity map. For areas where no initialization is required, the intensity in the manual disparity map is set to zero(black), as in the example in **Fig.2**. The disparity value used for initialization is obtained manually from shift in matching points. We created a simple utility which overlays the input images, and allows the user to shift the images manually. This enables the user to easily obtain disparity for object edges and other feature points. For low texture areas such as the white background behind the clock in **Fig.2**, it is very difficult, but unnecessary, to obtain the "ground truth" disparity value. In this case it is sufficient to set the disparity of the background such that it is behind the (farthest) object (e.g. the clock in **Fig.2**).

• *Manual edge map*: This binary bitmap is manually drawn and defines object edges that have a disparity jump. The edge information is used in the Graph Cuts optimization to cut disparity smoothing. If the manual edge map indicates an edge, the disparity in the depth map is expected to jump. If this edge-map indicates no edge, then the disparity is expected to change smoothly. In all of our test sequences it worked best to overlay the input color image and edge map in different layers in the image editor, and manually trace the edge in the layer of the edge map. It may be helpful to first histogram equalize the color image to enhance object edges. In our experience, using an edge detector output such as Canny, seemed not useful because deleting false object edges and correcting missed edges took more time than manual tracing an edge. Currently, our method supports only edge maps at integer pixel accuracy, which seemed sufficient in all our experiments.

• *Manual static map*: If the automatic detection of static pixels is inaccurate e.g. due to shadow, reflection, or noise, this map can overrule the automatic detection mechanism. Any non-zero pixels in this map indicate static pixels and corresponding depth values are fixed static temporally until another static map is supplied. Note that this map only indicates areas where the automatic detection of static pixels is inaccurate, and is not required at all in some sequences. When drawing the manual static map it is easiest to use the manual edge map as starting point and "flood fill" object areas (see **Fig.2**).

Recall that the goal of our method is to improve the depth such that view synthesis artifacts are reduced, not necessarily to obtain "ground truth" depth. Therefore, we can reduce the amount of manual work by providing only manual data in the manual disparity and static maps for areas where this is necessary. For example for the manual static map in **Fig.2**, the automatic detection of static areas is not accurate enough around the edge of the desk and the thin chair-legs, for all other static areas the automatic detection was accurate enough.

In our method, the described manual data can be input for any arbitrary frame. It depends on the sequence how many frames, and which frames require manual input data. For the test sequences used in section 4, "Book arrival" and "Doorflowers" consist of 100 frames, and "Newspaper" and "Champagne tower" of 200 frames. For "Book arrival", "Doorflowers", and "Newspaper", we supplied manual data for frame 0 only, for "Champagne tower" we supplied manual data for three frames. As an example, the amount of time required to make the manual disparity map, edge map and static map of **Fig.2**, is about 10, 45, and 10 minutes respectively. Of course with the use of specially designed software the amount of manual work could be greatly reduced.

3.3 Depth estimation by energy optimization

In this section we will describe how the manual input data is used in the Graph Cuts optimization. As mentioned before, depth estimation can be considered a labeling, which can be formulated in terms of an energy function, given by:

$$E(f) = \sum_{p \in P} D_p(f_p) + \sum_{p,q \in N} V_{p,q}(f_p, f_q) \qquad (2)$$

where D_p is the data term and $V_{p,q}$ is the smoothing term. P is the set of all pixels in the image, and N is the set of direct neighboring pixels p and q. We use three input views in our depth estimation algorithm to handle occlusion. As matching cost for the data term we use pixel matching, or 3-by-3 block matching based on absolute intensity differences. We obtain the matching cost $M_{x,y,d}$ by calculating the cost between center and left, and center and right camera, and select the smallest cost:

$$\begin{aligned} M_{x,y,d} &= \min(Lcost, Rcost), \\ Lcost &= \|L(x+d, y) - C(x,y)\|, \\ Rcost &= \|R(x-d, y) - C(x,y)\| \end{aligned} \qquad (3)$$

where d indicates disparity, $C(x,y)$ is the intensity of pixel p at (x,y) in the center camera, and similarly R, L, are the intensity in the right and left camera at $(x-d,y)$ and $(x+d,y)$, respectively. In the *Init Stage*(the left flow in **Fig.1**), the disparity values of the manual disparity map are used to update the data term of (3). If the intensity of the manual disparity map $Dm(x,y)$ at coordinates (x,y) is other than $''0''$ it represents a disparity initialization value, which is used to obtain the data term as:

$$D_p(f_p) = \begin{cases} M_{x,y,d} & \text{if} Dm(x,y) = 0, \\ 0 & \text{if} Dm(x,y) = d, \\ 2M_{x,y,d} & \text{else} \end{cases} \qquad (4)$$

For temporal consistency we want to propagate depth for static pixels. The automatic motion map is obtained by the Mean Absolute Difference (MAD) of the current frame and previous frame of the center camera view. It is used to update the matching cost, similar as in[15], by a weighted difference of the current disparity and previous disparity value. For pixels that are detected as non-static, the matching cost remains unchanged, otherwise the data term is updated based on the manual static map and the automatically obtained motion map. Therefore, in *Temporal Stage* (the right flow in **Fig.1**), the data term becomes as follows:

$$D_p(f_p) = \begin{cases} 0 & a, \\ 2M_{x,y,d} & b, \\ M_{x,y,d} + |d - D_{prev}(x,y)| & c, \\ M_{x,y,d} & else \end{cases} \qquad (5)$$

a : if $MS(x,y) = \text{static}$ & $d = Dinit(x,y)$
b : if $MS(x,y) = \text{static}$ & $d \neq Dinit(x,y)$
c : if $motion\_map(x,y) = \text{static}$
where $Dinit$ is the disparity map obtained in the

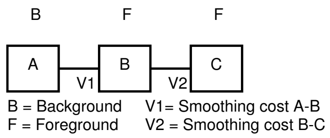


Fig. 3 Smoothing between 3 adjacent pixels.

most recent *Init Stage*, MS the manual static map, $motion\_map$ is the automatic motion map, and $Dprev$ is the disparity map of the previous frame.

The smoothing term is updated to cut the smoothing at disparity edges as indicated by the manual edge map, and is defined as follows:

$$V_{p,q} = \beta\lambda|f_p - f_q| \tag{6}$$

where $\lambda|f_p - f_q|$ is a commonly used smoothing term, which we scale by scaling factor β. λ is an empirically chosen smoothing factor, which ranges between 1.0 and 4.0 for our test sequences. If the manual edge map is defined and indicates an edge, then $\beta = 0.1$, else it is 1.0. Note that β cannot be set to 0, which we will explain using **Fig.3**. We consider three neighboring pixels A,B, and C, with horizontal smoothing term only, as indicated by $V1$ and $V2$ in **Fig.3**. Here we assume pixel A is on the background, and pixel B and C belong to a foreground object. Furthermore, pixel B is indicated by the manual edge map as on a edge. By setting β very low, e.g. $\beta = 0.1$, we greatly reduce the smoothing but the smoothing cost between pixel A and B is still larger than between pixel B and C (because the disparity jump between A and B is larger), so $V1 > V2$. This keeps pixel B and C weakly connected. If $\beta = 0$, pixel B will be completely isolated from all its neighbors which is undesirable.

Finally, after obtaining the updated energy function, the per-pixel disparity map is obtained by Graph Cuts optimization.

4. Experimental results

To evaluate the performance of our algorithm, we carry out depth estimation and view synthesis experiments using four MPEG test sequences. For the view-synthesis we used the MPEG View Synthesis Reference Software (VSRS version 3.5)[16]. For each sequence we obtain depth videos using automatic depth estimation and the proposed semi-automatic method and analyze the depth maps and view synthesis results. We use four test videos, namely Bookarrival, Doorflowers, Newspaper, and Champagne-tower. **Fig.4** shows the depth estimation results and the used manual input data for one frame of all four sequences. Note that the intensity values in the manual disparity map have been scaled for better visibility. From **Fig.4** it can be clearly seen that the semi-automatic depth estimation results are much improved over the automatic results. The disparity initialization from the manual disparity map propagates to surrounding areas due to the smoothing term in the energy definition. The amount of initialization required, depends on how distinct the data term is. In **Fig.4**, the top three rows show results of the *Init Stage*. Note that the last row (Doorflowers) shows frame 52, which is a result of the *Temporal Stage*. In this Doorflowers experiment, only frame 0 was initialized using the manual disparity and edge maps as shown in **Fig.4**, and the manual static map as shown in **Fig.2**. The view-synthesis results for this case is shown in **Fig.5**. The view-synthesis artifacts around the chair legs and door are greatly reduced by the semi-automatic depth estimation.

To show the temporal consistency of our depth maps, we take 4 consecutive frames of the Bookarrival sequence as shown in **Fig.6**. It can be seen that the depth propagated from the *Init Stage* results in much better temporal consistency for the static areas like the background. When compressing both depth maps using H.264, the rate-distortion plot (see **Fig.6b**) clearly shows that the semi-automatic depth is much easier to compress because it is more temporally stable.

5. Conclusion

In this paper, we have proposed a semi-automatic depth estimation algorithm based on an energy minimization framework. Our approach is an extension of an automatic depth estimation algorithm, whereby additional manually created data is input for one or multiple frames. The manual data provides disparity initialization, information on object edges, and information of static areas. In our experiments, we have shown that our proposed method can generate depth with clear object boundaries that is much improved over automatically generated depth, and consequently, view-synthesis artifacts were reduced. One limitation of our method is that it cannot cope well with textureless slanted surfaces, unless enough disparity initialization is supplied. In our approach, we rely on the Graph Cuts smoothing to spatially propagate the disparity initialization values. Segmentation based depth estimation methods often use a plane-fitting algorithm (for example[14]), which could benefit our method too. Another limitation is that our method is best suited for sequences with limited motion. If the scene contains a lot of motion, the temporal algorithm cannot propagate depth into following frames much and manual data may be required for an increasing number of frames. A solution would be to include motion or optical flow into the algorithm. To accurately obtain motion especially around object boundaries is a challenging topic.

Finally, this research is partially supported by Strategic Information and Communications R&D Promotion Programme (SCOPE) 093106002 of the Ministry of Internal Affairs and Communications. We would like to thank Heinrich-Hertz-Institut (HHI), for providing the "Book Arrival", and "Doorflowers" sequences, and Gwangju Institute of Science and Technology (GIST) for providing the "Newspaper" sequence.

〔References〕

1) C. Fehn, "Depth-image-based rendering (DIBR), compression and transmission for a new approach on 3D-TV", Proc. SPIE Stereoscopic Displays and Virtual Reality Systems, XI, pp.93-104 (Jan.2004)
2) A. Kubota, A. Smolic, M. Magnor, M. Tanimoto, T. Chen, and C. Zhang, "multiview imaging and 3DTV", Signal Processing Magazine, 24(6):10-21 (2007)
3) T. Fujii, "A Basic Study on Integrated 3-D Visual Communication", Ph.D dissertation in engineering. The University of Tokyo (1994)
4) T. Fujii, M. Tanimoto, "Free-viewpoint Television based on the Ray-Space representation" Proc. SPIE ITCom 2002,175-189 (2002)
5) M. Tanimoto, "FTV (Free viewpoint TV) and Creation of Ray-Based Image Engineering", ECTI Transaction on Electrical Engineering, Electronics and Communications, **6**,1, pp.3-14 (2008)
6) J.Konrad, M.Halle, "3D displays and signal processing: an answer to 3-D ills?", IEEE Signal Processing Magazine, **24**,6 (2007)
7) D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms", IJCV, 47(1-3):7-42 (2002)
8) V. Kolmogorov, R. Zabih, "What Energy Functions Can Be Minimized via Graph Cuts?", IEEE transactions on pattern analysis and machine intelligence, **26**,2 (Feb.2004)
9) V. Kolmogorov and R. Zabih, "Multi-camera scene reconstruction via Graph Cuts", in European Conference on Computer Vision (ECCV), **3**, pp. 82-96 (2002)
10) P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient belief propagation for early vision", IEEE CVPR, **1**, pp. 261-268 (2004)
11) J.Sun,Y.Li, S.B.Kang, H.Shum, "Symmetric Stereo Matching for Occlusion Handling", Proc. IEEE Conf. Computer Vision and Pattern Recognition (2005)
12) P.Belhumeur, D.Mumford, "A Bayesian treatment of the stereo correspondence problem using half-occluded regions", Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR 1992)
13) H. Tao, H. S. Sawhney, and R. Kumar, "A global matching framework for stereo computation", ICCV, **I**:532-539 (2001)
14) A. Klaus, M. Sormann and K. Karner, "Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure", (ICPR 2006)
15) Sang-Beom Lee, Yo-Sung Ho, "Multi-view Depth Map Estimation Enhancing Temporal Consistency", 23rd International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC2008)
16) ISO/IEC JTC1/SC29/WG11, "View Synthesis Algorithm in View Synthesis Reference Software 2.0 (VSRS2.0)", Doc. M16090, Lausanne, Switzerland (February 2009)

Meindert Wildeboer received the MSc degree in Electronics from the University of Hertfordshire, UK, in 1996. He worked in R&D of several companies before joining the Graduate School of Engineering, Nagoya University, where he is currently working as a researcher. His current research interests include 3D image processing and computer vision.

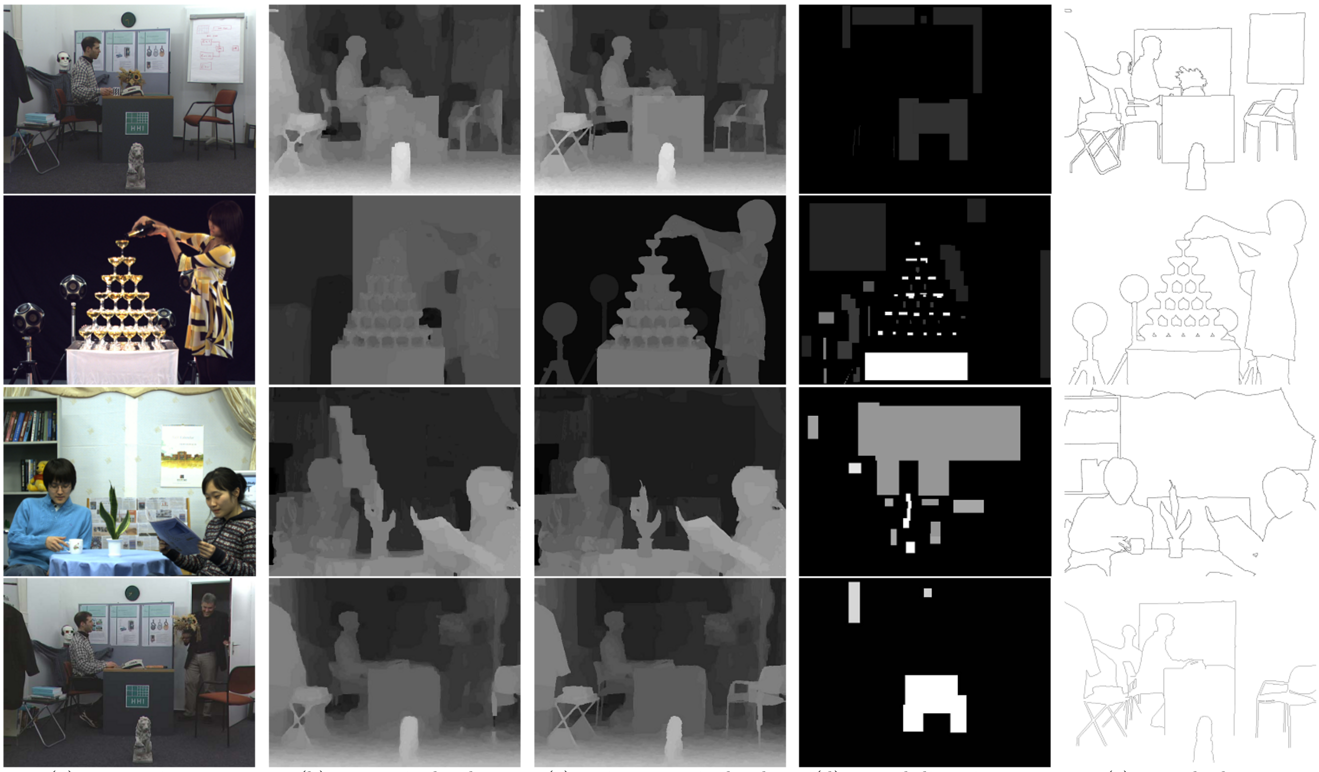
Norishige Fukushima received a B.E., M.E., and Ph.D. degree from Nagoya University, Japan, in 2004, 2006, and 2009. He is currently an assistant professor at Graduate School of Engineering, Nagoya Institute of Technology. His research interests are multi view image capturing,processing, and calibration.

Tomohiro Yendo received the B.E., M.E., and Ph.D. degrees from Tokyo Institute of Technology, Tokyo, Japan, in 1996, 1998, and 2001, respectively. He is now an assistant professor at the Graduate School of Engineering, Nagoya University. His current research interests include 3-D image display, capturing and processing.

Mehrdad Panahpour Tehrani received Dr.E. degree in Information Electronics from Nagoya University, Nagoya, Japan in 2004. Currently, he is working as an Associate Professor at the Graduate School of Engineering, Nagoya University, Japan. His research interests are 3D image processing and communication.

Toshiaki Fujii received the Dr.E. degree in Electrical Engineering from the University of Tokyo in 1995. He is currently an Associate Professor in the Graduate School of Science and Engineering, Tokyo Institute of Technology. His current research interests include multi-dimensional signal processing and their applications for ITS.

Masayuki Tanimoto received the B.E., M.E., and Dr.E. degrees in electronic engineering from the University of Tokyo, Tokyo, Japan, in 1970, 1972, and 1976, respectively. Since 1991, he has been a Professor at Graduate School of Engineering, Nagoya University. His current research interests include FTV and ITS. Dr. Tanimoto is a Fellow of ITE.



(a)camera view (b)automatic depth (c)semi-automatic depth (d)manual disparity map (e)manual edge map

Fig. 4 Left to right: camera view, automatic depth result, semi-automatic depth result, manual disparity map, manual edge map. Top to bottom:Bookarrival, Champagne-tower, Newspaper, and Doorflowers.

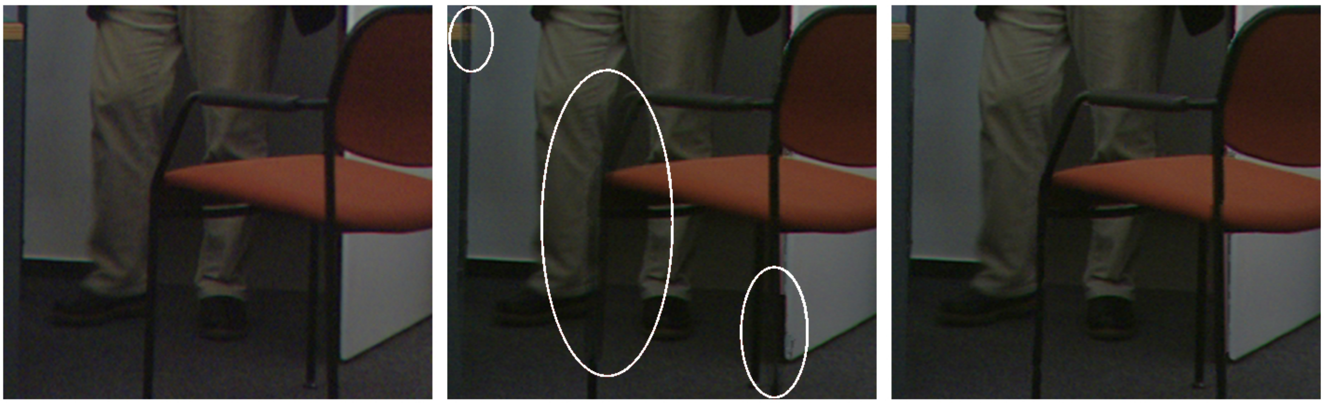
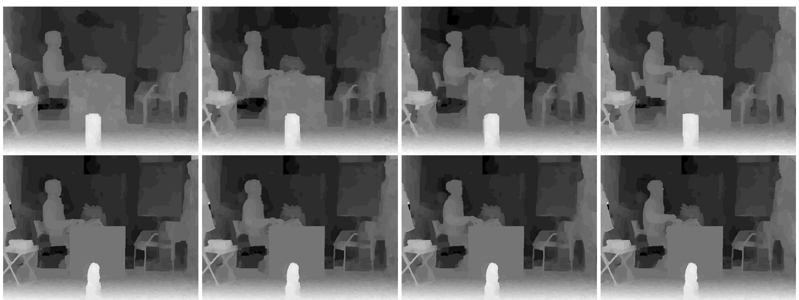
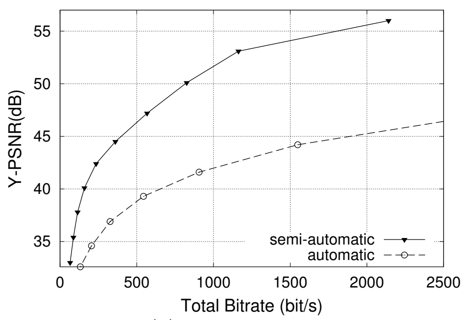


Fig. 5 View-synthesis results of Doorflowers of **Fig.4** bottom row. From left to right: ground truth, results using automatic depth, and results using semi-automatic depth.



(a)Temporal consistency of four frames of Bookarrival



(b)Rate-Distortion plot

Fig. 6 (a)Temporal consistency of four consecutive frames for the automatic depth (top row) and the semi-automatic depth (bottom row). (b)Compression experiment: H.264 RD-plot of all frames of the semi-automatic and automatic depth maps.