Opt Rev (2015) 22:443–447
DOI 10.1007/s10043-015-0078-5

REGULAR PAPER

Accurate depth sensing by filtering IR cost on the guide of RGB information

Xuanwu Yin[1] · Guijin Wang[1] · Chenbo Shi[1] · Qingmin Liao[1,2]

Received: 12 September 2014 / Accepted: 13 March 2015 / Published online: 21 April 2015

Abstract In this paper, we propose a depth-sensing method that combines the raw IR speckle pattern and RGB information. IR and RGB data are acquired from the same viewpoint. Matching costs between the realtime speckle pattern and the reference are computed with the IR data. Different from conventional methods, the RGB information is utilized to filter the matching costs. The depth at each pixel is then estimated by choosing the best match with the filtered costs. The proposed system can make use of the RGB information to achieve more accurate depth maps than the state-of-the-art, speckle-based depth-sensing methods.

Keywords Depth sensing · Three-dimensional image acquisition · Structured light · Multiple sensor information fusion

1 Introduction

Measuring the three-dimensional structure of a scene or an object using stereophotogrammetric methods attracted more and more attention over last decades. The 3D information can be used in scene modeling [1, 2], human-computer interaction [3], movie making [4, 5] and so on.

The most basic technique is two-view stereo vision [6–8], which searches the correspondences between two views and estimated the depth by triangulation. However it is not always easy to find these correspondences in natural scene images because of textureless regions, repeated patterns etc. To find the correspondences robustly, researchers have developed many active methods to encode the scene. Among them, random speckle pattern [9–13] has attracted more and more attentiveness by researches these years. Schaffer et al. [9] projected a sequence of speckle pattern onto the the object, and computed depth by temporal correlation on this pattern sequence. Other researchers have tried to do depth sensing with spatial correlation technique on a pair input image [10–13]. However, because of the loss of structure information in the IR image, there are always unreliable values, especially on the object boundaries.

RGB-D (RGB-Depth) data processing has attracted a lot of attention in recent years. Henry et al. [1] and Stuckler et al. [2] modeled the scene with multiple RGB-D pairs captured by ready-made RGB-D sensor (Kinect). Instead of fusing multiple depth inputs, Diebel et al. [14] tried to get a single better depth map by upsampling the depth map with a high-spatial resolution RGB image. In their method, the input is a single RGB-D pair, and the final depth map is achieved by globally optimizing a multi-layer MRF (Markov Random Field).

The focus of this work is similar with Diebel's except that the inputs of our method are a raw IR speckle pattern and an associated RGB image. To be clear, we process RGB-Raw data rather than RGB-D data. The motivation that we process RGB-Raw data is that the raw data retains the original information of the IR speckle pattern [12, 13], which makes it possible to recover more accurate depth with the assistance of RGB data.

In this work, a device that can acquire IR data and RGB data synchronously from the same viewpoint is set up. Based on this device, we propose a depth computation algorithm that makes use of both IR data and RGB data, rather than utilizing the RGB data only in post processing.

✉ Guijin Wang
wangguijin@tsinghua.edu.cn

[1] Department of Electronic Engineering, Tsinghua University, EE-Rohm Hall 6-107, Beijing 10084, China

[2] Department of Electronic Engineering, Graduate School at Shenzhen, Tsinghua University, Shenzhen 518055, China

The RGB information is applied to filter the original matching cost, rather than the calculated depth value [14]. The introduction of RGB information makes the depth values more accurate, and more coincident with the actual scene, especially on the edges, as the experiments show.

This paper will cover the following material: the structure of the data acquisition device and the overview of the depth-sensing algorithm are presented in Sect. 2; the implementation details are described in Sect. 3; the error and resolution analysis is presented in Sect. 4, along with some real scene results; a brief conclusion and discussion about future work are mentioned at last.

2 Device setup and algorithm framework

Our data acquisition system is made up of one IR pattern projector, one infrared digital camera, one visual light RGB camera, and one light splitter, Fig. 1. The Microsoft Kinect is taken as the IR projector, which projects random speckle pattern onto the scene. Aligned with the projector in Y and Z direction, the infrared camera captures the realtime speckle pattern reflected by the scene. We use a cold mirror as the light splitter. The cold mirror is put between the IR camera and the RGB camera, allows the infrared light pass through to the IR camera, and reflects the visual light to the RGB camera. As there is no depth information available when the RGB data is used, the RGB camera is guaranteed to be aligned with the IR camera physically to avoid any disparities between them.

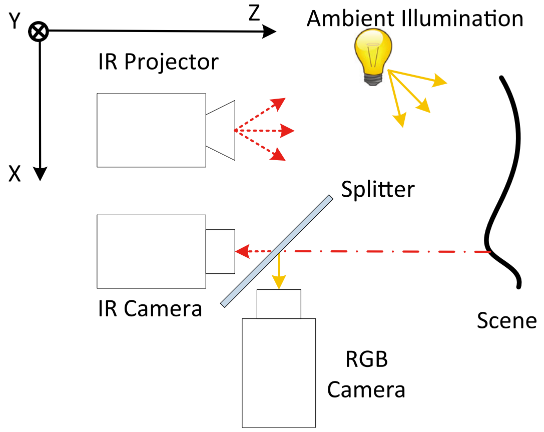


Fig. 1 Device structure

By this setup, the depth can be computed by estimating the translation of the realtime speckle pattern relative to the reference pattern. The computation consists of four steps: cost volume constructing, cost slices filtering, disparity assignment, and depth transformation, presented in Fig. 2. In the first three steps, our purpose is to assign each pixel i with coordinates (x, y) in the image I to a disparity from $\mathbf{D} = \{\mathbf{d_1}, \mathbf{d_2}, \ldots, \mathbf{d_N}\}$, with N candidate disparities in total. At last, the disparity value is transformed to depth value according to the geometry of the device.

3 Implementation

3.1 Cost volume construction

The cost volume C is a three dimensional array, in which each element at (x, y, d_n) represents the cost of choosing disparity d_n for pixel (x, y), Fig. 3. We construct the cost volume with a realtime IR image, and a fixed reference IR pattern. The reference pattern is pre-captured by putting the device's optical axis perpendicular to a flat

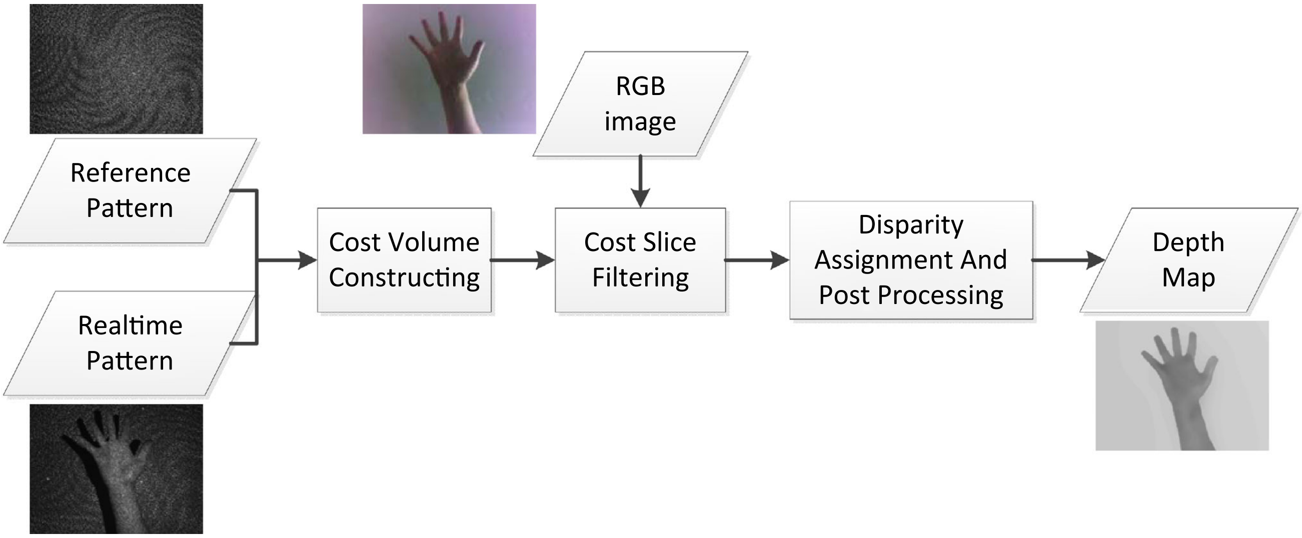


Fig. 2 Data and processing flowchart

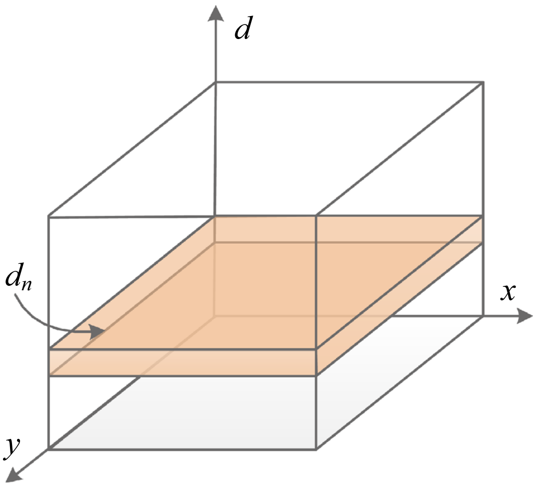


Fig. 3 Cost volume

plane, such as a wall, at a certain distance. As the realtime pattern and reference pattern are captured separately, there are significant differences in brightness and contrast between them. To handle these differences, the ZNCC (zero-mean normalized cross correlation) method is adopted to compute the costs, which is defined as

$$R_{i,d_n} = \frac{\sum_{(x,y)\in W_i} T(x,y)\cdot T'(x+d_n,y)}{\sqrt[2]{\sum_{(x,y)\in W_i} T^2(x,y)\cdot\sum_{(x,y)\in W_i} T'^2(x+d_n,y)}}, \tag{1}$$

where $T(x,y) = I(x,y) - \bar{I}(x,y)$ is the difference between the intensity value and the local averaging in the realtime pattern, $T'(x,y)$ has the similar meaning in the reference pattern, and W_i is the window around pixel i. Note that $-1 \le R_{i,d_n} \le 1$. The two points match perfectly only if $R_{i,d_n} = 1$, large values generally indicate that they closely corresponds to each other. We define the cost with ZNCC as

$$C_{i,d_n} = 1 - R_{i,d_n}. \tag{2}$$

3.2 Slice filtering

After computing the costs for all pixels at all disparities, the volume C now has N slices. The n−th 2-d slice of the volume stores the costs for all pixels at disparity d_n. We are going to filter each slice with the RGB information. The filtering output at disparity d_n of pixel i is a weighted average of the pixels in the same slice:

$$C'_{i,d_n} = \sum_j w_{i,j}(I^g)C_{j,d_n}, \tag{3}$$

C' is the filtered cost volume and i and j are pixel indexes. The filter weights $w_{i,j}$ is derived from the RGB guidance image I^g. We use the guided filter's [15] weights here, which is defined as

$$w_{i,j} = \frac{1}{|W|^2}\sum_{k:(i,j)\in W_k}\left(1+(I_i^g-\mu_k)^T(\Sigma_k+\epsilon U)^{-1}(I_j^g-\mu_k)\right), \tag{4}$$

where W_k is a square window centered at pixel k, and $|W|$ the number of pixels in it. I_i^g, I_j^g and μ_k are 3×1 color vectors, μ_k is the mean vector in W_k, and the covariance matrix Σ_k and identity matrix U are of size 3×3. ϵ is a smoothness parameter. By this filtering weight definition, if two pixels lie on different sides of the edge, the elements of Σ_k will be large. Thus the weight will be small. Otherwise if they lie on the same side, the weight will be large. Which means that pixels on the same side influence each other more than those on different sides. This property guarantees that pixels with similar colors have similar depth values, and can reduce the smoothing effects on the depth edges.

3.3 Disparity assignment and transformation

Once we obtain the filtered cost volume C', the disparity at pixel i is chosen by

$$d_i = \arg\min_{\mathrm{d}} \mathrm{C}'_{\mathrm{i,d}}. \tag{5}$$

After assigning disparity to all the pixels, we use linear interpolation [13] to obtain sub-pixel disparity. Then the sub-pixel disparity d_{sub} is converted to depth value by triangular geometry,

$$Z = \frac{s}{d_{sub} + s/Z_0}, \tag{6}$$

where s is a constant determined by the imaging system, and Z_0 is the depth of the reference plane I'.

4 Experimental evaluation

4.1 Error analysis

The resolution of depth estimation depends on the random error of the disparity estimates. According to (6), the variance of Z can be calculated as

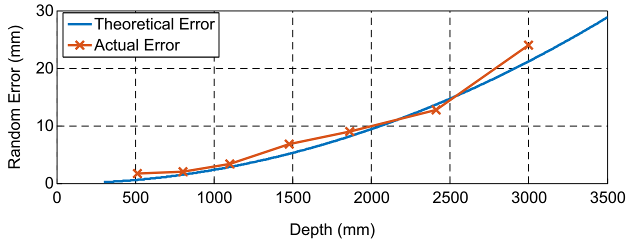
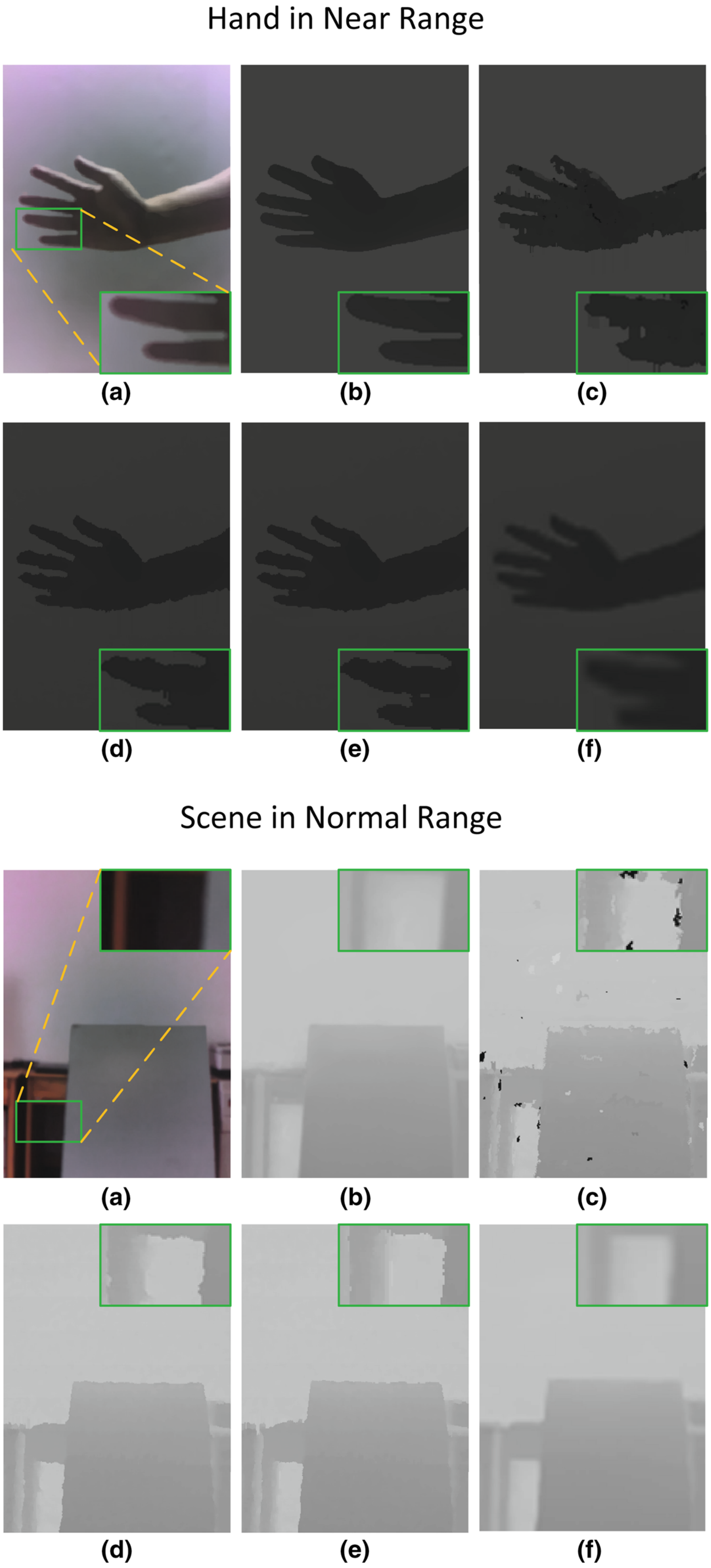


Fig. 4 Depth estimation error

Fig. 5 Depth results compared with state-of-the-art methods, insets are the local details. **a** RGB Scene. **b** Proposed. **c** Naive NCC. **d** Kinect. **e** IGMP. **f** MRF upsampling



$$\sigma_Z^2 = (\frac{\partial Z}{\partial d})^2 \sigma_d^2. \tag{7}$$

where σ_d^2 is the variance of the estimated disparity d_{sub}. Plug (6) into (7), finally we can get

$$\sigma_Z = \frac{Z^2}{s} \sigma_d, \tag{8}$$

which means that the random error of depth is proportional to the random error of disparity, and the square of the depth.

We test the error by computing the depth of planes at different distances. The result is presented in Fig. 4. In our proposed system, σ_d is assumed to be 0.2 pixel. The actual error coincides with the theoretical one, and the depth resolution is about 10mm at 2m.

4.2 Scene test

We have tested the system on some different scenes, Fig. 5. The first is one hand about 57 cm away. The second is an indoor scene about 2.5 m away. To show the effectiveness of introducing RGB data, We compare our method with some state-of-the-art methods, including naive NCC method, Kinect, IGMP (Iterative Grid Message Passing) method proposed in [13] and MRF Upsampling proposed in [14].

The difficulty of computing depth from raw speckle pattern only lies in the sparsity and deformation of the pattern. Thus naive NCC matching produces many errors in the depth map, especially on the boundaries. Kinect and IGMP optimize the depth map based on the initial matching and perform better than NCC. However, the sparsity of speckle pattern ensures that the depth edges do not coincide with the objects' boundaries.

The MRF Upsampling method processes the depth map directly by utilizing RGB information. Although the depth edges seem to coincide with the boundaries better, the depth map is actually blurred, as presented in Fig. 5. Besides, this method needs to globally optimize a multi-layer MRF, thus is restricted by the limited storage space and computing speed.

Different from MRF Upsampling, the proposed method filters the cost volume instead of the depth map, thus can produce sharp depth edges that coincide with the boundaries. This makes it possible to segment objects more accurately, and brings large benefits to many depth based applications, such as object labeling, scene understanding and human-computer interaction. And this filtering approach avoids optimization, thus is more efficient than MRF Upsampling.

5 Conclusion and future work

We built a hybrid, depth-sensing system with an IR camera and an RGB camera, which can take both structured IR pattern and natural RGB information into account. Rather accurate depth maps can be obtained by the proposed device, especially on the boundaries. In our device, the distinction of one pixel is guaranteed by the randomness of the pattern. However in the aspect of information theory, random pattern is not the best. In the future, we will work on designing better pattern to make each pixel distinguish from others, in the sense of encoding. And actually, it is possible to extract much more information than RGB value from the color image, we will investigate to provide better inputs for the guided filter, instead of the currently used RGB value.

Acknowledgments Thanks for the comments of Dr. Jinghao Xue, University College London. This work is partially supported by NSFC 61327902, NSFC 61271390 and National High-tech R&D Program 2015AA016304.

References

1. Henry, P., Krainin, M., Herbst, E., Ren, X., Fox, D.: Int. J. Rob. Res. **31**, 647 (2012)
2. Stuckler, J., Behnke, S.: Proc. of the IEEE Int. Conf. on Multisensor Fusion and Information Integration, p. 162 (2012)
3. Liu, B., Wang, G., Chen, X., He, B.: Proc. SPIE 9045, 2013 Int. Conf. Opt. Inst. and Tech., p. 90450L (2013)
4. He, B., Wang, G., Shi, C., Yin, X., Liu, B., Lin, X.: IEICE T. Inf. Syst. **96**, 2096 (2013)
5. He, B., Wang, G., Zhang, C.: J. Vis. Commu. Image R. **25**, 1031 (2014)
6. Scharstein, D., Szeliski, R.: Int. J. Comput. Vision **47**, 7 (2002)
7. Shi, C., Wang, G., Pei, X., He, B., Lin, X.: IEICE T. Inf. Syst. **95**, 699 (2012)
8. Shi, C., Wang, G., Yin, X., Pei, X., He, B., Lin, X.: To be published in IEEE T. Image Process
9. Schaffer, M., Grosse, M., Harendt, B., Kowarschik, R.: Opt. Lett. **36**, 3097 (2011)
10. Wiegmann, A., Wagner, H., Kowarschik, R.: Opt. Express **14**, 7692 (2006)
11. Freedman, B., Shpunt, A., Machline, M., Arieli, Y.: U.S. Patent 8150142 (2012)
12. Wang, G., Yin, X., Pei, X., Shi, C.: Appl. Opt. **52**, 516 (2013)
13. Yin, X., Wang, G., Shi, C., Liao, Q.: Opt. Eng. **53**, 013105 (2014)
14. Diebel, J., Thrun, S.: Adv. Neural Inform. Process. Syst. **18**, 291 (2005)
15. He, K., Sun, J., Tang, X.: IEEE T. Patt. Anal. **35**, 1397 (2013)