DIML/CVL RGB-D Dataset: 2M RGB-D Images of Natural Indoor and Outdoor Scenes

Digital Image Media Laboratory (DIML), Yonsei University, Seoul, South Korea,
Computer Vision Laboratory (CVL), Ewha Womans University, Seoul, South Korea
https://dimlrgbd.github.io/

I. Overview

This manual is intended to provide a detailed description about the DIML/CVL RGB-D dataset. This dataset is comprised of 2M color images and their corresponding depth maps from a great variety of natural indoor and outdoor scenes. The indoor dataset was constructed using the Microsoft Kinect v2 [1], while the outdoor dataset was built using the stereo cameras (ZED stereo camera [2] and built-in stereo camera) Table I summarizes the details of our dataset, including acquisition, processing, format, and toolbox. Refer to Section II and III for more details. The detailed camera specifications and RGB-D sample data are shown in Table II and III, respectively.

This dataset was taken from fall 2015 to summer 2017, in South Korea. We built it for the purpose of promoting depth-related applications based on deep learning. Our dataset differs from existing ones in the following aspects:

1) It is comprised of both indoor and outdoor RGB-D data.
2) Large-scale RGB-D dataset is provided, consisting of 1M indoor data and 1M outdoor data.
3) Unlike existing outdoor data using stereo cameras mounted on a moving vehicle, ours was taken using hand-held stereo cameras.
4) Confidence maps are provided together to quantify the accuracy of depth maps computed from stereo camera in outdoor scenes.

Remark: We are now in the process of analyzing the characteristics of the DIML/CVL RGB-D dataset. For a timely impact on research communities, we first release a subset of the dataset before completing the analysis. The comprehensive report including performance analysis will be released later with the complete 2M RGB-D data.

II. Indoor RGB-D dataset

A. Data Acquisition

1) Hardware setup: We captured RGB-D videos of different indoor scenes using the Microsoft Kinect v2 (time-of-flight sensor) [1]. It has been known that compared to the Kinect v1 (structured light method), the Kinect v2 provides sharper and more detailed depth maps with high-quality color streams, while it does not work well on slanted surfaces. We believe that our indoor dataset is complementary to the NYU dataset taken with the Kinect v1 [3] in numerous depth related applications. Fig. 1 shows the hardware setup for indoor data acquisition. An external battery was utilized to enhance the mobility of the Kinect v2 in the hand-held setup. Additionally, our RGB-D data was captured with a tripod and slider as shown in Fig. 1. The average camera velocity was slow enough (10 frames per second) to avoid motion blur in color images.

Fig. 1. Hardware setup for indoor data: (from left to right) Microsoft Kinect v2 [1], external battery, and tripod and slider.

2) Data Acquisition: The color and depth streams were originally captured with 1920×1080 and 512×424 resolutions, respectively. We set a threshold of 0.5∼7 meters due to inaccuracies in depth measurements at near and far ranges. We also provide additional 0.1M IR images aligned to color images.

Our indoor scenes include a variety of commercial buildings and residential areas including office, room, exhibition center, to name just a few, which are located in different 9 districts of 4 cities (Seoul, Daegu, Goyang, and Anyang), as shown in Fig. 2. The total numbers of categories and scenes are 18 and 283, as summarized in Table IV.

The categories include warehouse, cafe, classroom, church, computer room, meeting room, library, laboratory, bookstore, corridor, bedroom, livingroom, kitchen, bathroo, restaurant, billiard, hospital, and store. Especially, "bedroom" and "livingroom" contain a variety of scenes in indoor interior exhibition halls.

B. Data Processing

The overall pipeline for processing indoor data is illustrated in Fig. 3. To co-align color image and depth map, we first projected the raw depth map into the RGB camera coordinate. The holes in the warped depth map were then filled using color-guided depth upsampling algorithm.

1) Calibration: We calibrated and aligned the indoor data using 1) Kinect SDK or 2) 'iai_kinect v2 libary' [4]. Approximately 476,000 image pairs (taken from fall 2015 to summer 2016) out of 1M RGB-D data were processed using the MS Kinect SDK. A depth map was projected into the RGB camera coordinate using the Kinect SDK built-in functions.

The 'iai kinect v2 libary' was used to process the remaining RGB-D image pairs (taken from fall 2016 to summer 2017).

TABLE I
SUMMARY OF DIML/CVL RGB-D DATASET.

	Indoor dataset	**Outdoor dataset**
Data acquisition	Microsoft Kinect v2	Stereo camera (ZED and built-in camera)
Data processing	Calibration using the Kinect SDK [1] Calibration using 'iai_kinect v2 libary' [4] Color guided depth upsampling [5]	Calibration and rectification using Caltech toolbox [11] Stereo matching [10] Confidence estimation [15]
Data format	**Color images** - Cropped images **Depth maps** - Original depth map - Projected depth map - Hole-filled depth map **Calibration parameters** - Intrinsic/extrinsic parameter for color and IR cameras - Shift parameter between IR and depth map	**Color images** - Rectified left and right images **Disparity, depth, and confidence map** - Left disparity and depth map - Left Confidence map **Calibration parameters** - Intrinsic/extrinsic parameters for stereo camera
MATLAB Toolbox	Fill depth	Confidence estimation

TABLE II
CAMERA SPECIFICATIONS USED FOR DATA ACQUISITION.

	Kinect v2 [1]	**ZED stereo [2]**	**Built-in stereo**
Color Resolution	1920 x 1080	1920 x 1080 1280 x 720	1920 x 1080 1280 x 720
Depth Resolution	512 x 424	1920 x 1080 1280 x 720	1920 x 1080 1280 x 720
Range of Depth	0.5 - 8.0 m	0.5 - 20 m	2 - 80 m
Baseline	-	12 cm	40 cm
Focal length		2.8 mm	3.5 mm
Data acquisition	Color and IR images, depth map	Left and right color images Depth and confidence maps	Left and right color images Depth and confidence maps

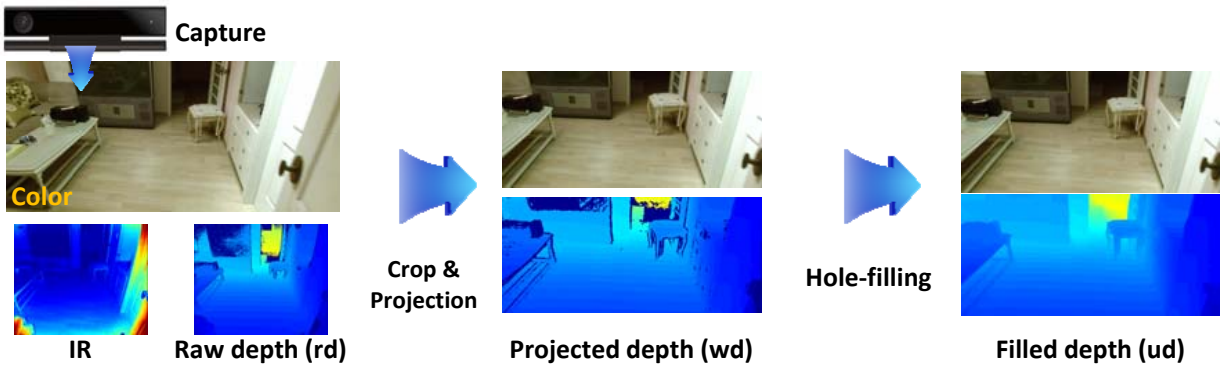


Fig. 3. The processing pipeline for our indoor dataset.

TABLE III
SAMPLE DATA FROM THE DIML/CVL RGB-D DATASET

Indoor				Outdoor			
Color	**Depth**	**Color**	**Depth**	**Left**	**Right**	**Disparity**	**Confidence**

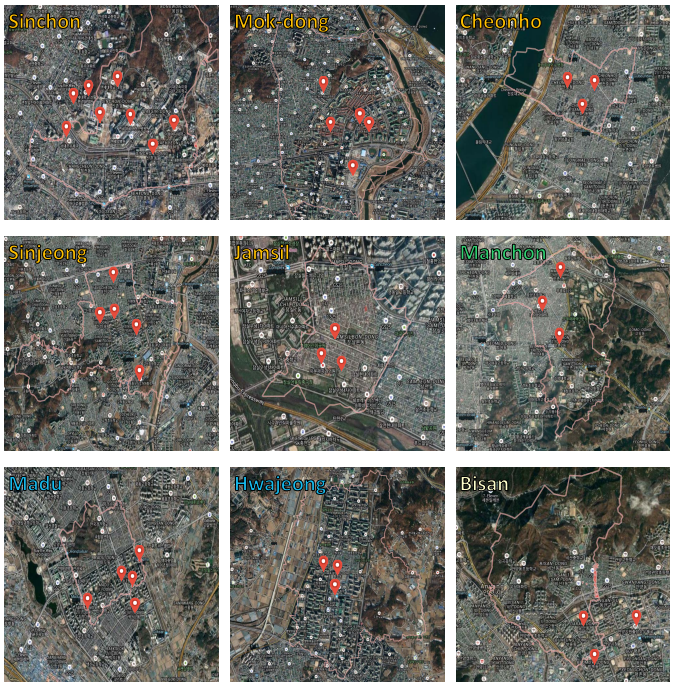


Fig. 2. Shooting locations of indoor data. Our indoor scenes were captured at various places, e.g., offices, rooms, dormitory, exhibition center, street, and so on. These are located in different 9 districts of 4 cities (South Korea).

We acquired several pieces of color, IR, and raw depth images containing a checkerboard pattern of 5×7 as shown in Fig. 4. Here, the distance between corners was set to 0.03m. Note that for an accurate calibration, the pattern images should be taken in a way of covering most of the image at various ranges. Using the calibration toolbox provided in the 'iai kinect v2 libary' [4], we estimated the shift parameter between IR and depth images and the projection matrix between IR and color images. The warping process consists of two parts. The depth map was first warped into IR image coordinate using a shift parameter. The depth map on IR camera coordinate was finally projected into the RGB image coordinate.

Note that warped depth maps has two different sizes depending on calibration toolbox used. The MS Kinects SDK [1] warped the depth map into the RGB image coordinate of 1/4 spatial resolution (480×270), while the 'iai kinect v2 libary' [4] performed the warping into the cropped RGB image coordinate. After projecting depth values to the RGB camera coordinate, we discarded the region exceeding the field of view of IR camera. Thus, the cropped color image is of 1408×792.

2) Depth upsampling: We applied the color-guided depth upsampling algorithm [5] to fill the holes in depth maps.

C. Data Format

We stored depth maps in mm using a 16bit portable network graphics (png) format. The indoor dataset includes the following files. Here, "**in**" denotes the indoor data.

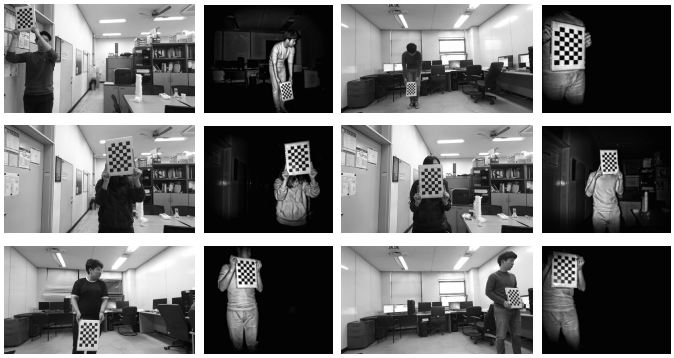


Fig. 4. Calibration board images for indoor data. We obtained several images by tilting the calibration pattern vertically and horizontally. At least 150 images were taken so that calibration pattern covers in all areas of the image.

- Color image
  "**in_k_14-digits_c.png**"
  24bit cropped RGB image (1408×792)

- Depth map
  "**in_k_14-digits_rd.png**"
  16bit original depth map in mm (512×424)
  "**in_k_14-digits_wd.png**"
  16bit projected depth map in mm (480×270),(1408×792)

  "**in_k_14-digits_ud.png**"
  16bit filled depth map in mm (1408×792)

- Calibration parameters
  "**calib_color_2-digits.mat**"
  intrinsic parameter for color camera
  "**calib_ir_2-digits.mat**"
  intrinsic parameter for IR camera
  "**calib_depth_2-digits.mat**"
  shift parameter between IR and depth
  "**calib_pose_2-digits.mat**"
  extrinsic parameter between color and IR cameras

A total of four Kinect v2 cameras were used, and we assigned an unique ID (14 digits) to each RGB-D image pair. The ID consists of 2 digits for camera number (01 $\sim$ 04), 6 digits for date (year, month, day), and 6 digits for order of acquisition , respectively (for example, 'in_k_01_160220_000001').

D. Indoor data toolbox

- "**fill_depth.m**" - Fills in the holes in depth map.

Refer to [1] and [4] about projecting the depth map and cropping the color region exceeding the field of view of IR camera.

III. Outdoor RGB-D Dataset

Unlike indoor scenes, obtaining an accurate and dense depth map of outdoor scenes poses additional challenges due to strong illumination and longer depth range, making it infeasible to use the time-of-flight sensor (Kinect v2). LiDAR can be an alternative for depth acquisition as in the KITTI dataset [6], but it produces a sparse depth map, e.g., for less than 6% of pixels in 1224×386 resolution. Aside from this, depth values are provided only at the bottom part of an color image due to a limited vertical field of view of LiDAR (typically 40°). To increase the density of the captured depth maps, we may accumulate multiple sparse depth maps (from LiDAR) using the iterative closest point (ICP) algorithm [8]. However, significant manual effort is required to remove outliers and artifacts in the depth map, due to the different centers of projection of the LiDAR and color camera or reflecting/transparent surfaces. Thus, this approach is not suitable for acquiring large scale RGB-D data in outdoors.

Alternatively, dense ground truth depth maps would be generated through synthetic rendering as in [9], but they incur the domain adaptation issue inherently when directly used for natural outdoor scenes.

TABLE IV
Indoor dataset categorization. We uploaded the data acquired using SDK and we will post our own calibration data later.

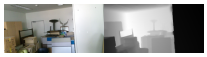
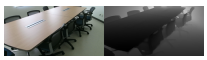
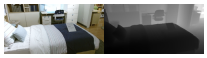
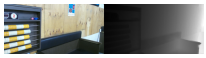
	Category	# of folders	# of files
	Warehouse	26	33923
	Cafe	22	33515
	Classroom	65	95867
	Church	10	18161
	Computer Room	7	12583
	Meeting Room	23	42241
	Library	15	22001
	Laboratory	22	29748
	Bookstore	5	9539
	Corridor	32	26236
	Bedroom	33	55215
	Livingroom	28	49502
	Kitchen	6	9274
	Bathroom	1	1140
	Restaurant	22	33034
	BilliardHall	1	3523
	Hospital	0	0
	Store	5	5606

Fig. 5. Hardware setup for outdoor data: (from left to right) ZED stereo camera [2], mvBlueFox3 sensor [7], and slider.

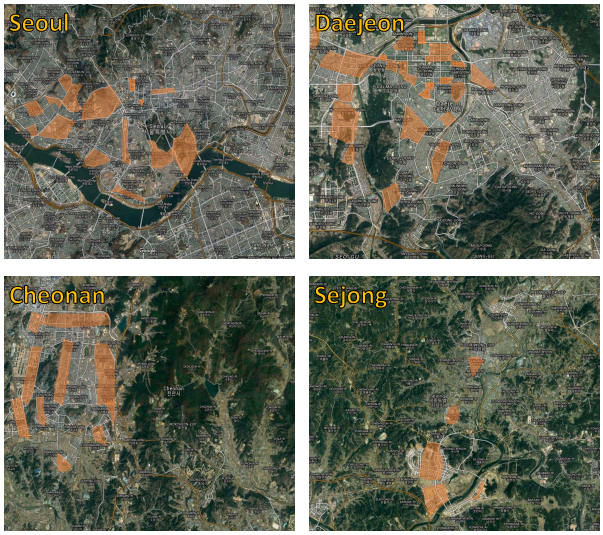


Fig. 6. Shooting locations of outdoor data. Our dataset acquired various outdoor scenes using the hand-held device, including park, building, brook, road, apartment, and so on. These are located in different 4 cities in South Korea.

Considering these factors, we adopted the stereo based approach for generating large scale RGB-D data in outdoor scenes. The depth map is computed using stereo matching algorithms [10]. To minimize the side effects of incorrectly estimated depth values, we additionally provide a confidence map [14] and [15] associated with the estimated depth map.

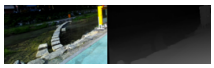
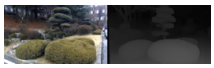
A. Data Acquisition

1) Hardware setup: We used the ZED camera and built-in stereo camera for acquiring stereo images (Fig. 5). Although the commercial ZED camera provide high quality stereo images, the sensing range is rather limited (up to 20m according to the specification sheet) due to a relatively small baseline (12cm). We designed the built-in stereo system with mvBlueFox3 sensors [7] and set the baseline to 40cm, increasing the sensing range up to 80m. The mvBlueFox3 SDK [7] was used for synchronization. The devices used for outdoor data acquisition are shown in Fig. 5.

2) Data acquisition: The stereo image was captured with 1920×1080 or 1280×720 resolutions. 0.92M and 0.08M stereo images were obtained using the ZED and the built-in stereo camera, respectively. The average camera velocity was 10 frames per second to avoid motion blur in stereo images.

The stereo images of the KITTI dataset [6] were obtained through stereo camera mounted on a moving vehicle, and thus the outdoor data is mainly comprised of road and traffic scenes. In contrast, our dataset acquired various outdoor scenes using the hand-held device, including park, building, brook, road, apartment, and so on, which are located in different 4 cities (Seoul, Daejeon, Cheonan, and Sejong), as shown in Fig. 6. The total numbers of category and scene are 9 and 132, as summarized in Table V. All the scenes were taken steadily with a tripod and slider while walking. We believe that our outdoor dataset is complementary to the existing outdoor datasets thanks to the diverse scene coverage.

TABLE V
OUTDOOR DATASET CATEGORIZATION. WE UPLOADED THE DATA ACQUIRED WITH [14] USING ZED.

	Category	# of folders	# of scenes	# of files
	brook	2	34	23,361
	building	31	226	298,570
	construction	2	51	11,314
	driveway	127	297	302,373
	field	4	30	18,996
	overpass	1	37	13,971
	park	10	91	116,930
	street	31	234	380,118
	trail	9	53	18,762

B. Data Processing

This task involves rectifying the stereo image and estimating a disparity map and its associated confidence map. Figure 7 illustrates the overall pipeline for processing outdoor data.

1) Calibration and Rectification: We performed our own calibration using the Caltech calibration toolbox [11]. The 7×8×0.26 and 7×10×0.50 checkerboard patterns were used for the ZED and built-in stereo cameras (Fig. 8), respectively. After correcting lens distortion, we rectified stereo images onto a common image plane such that the corresponding points exist on the same horizontal coordinate.

2) Stereo Matching: Given the rectified stereo image pairs, we generate the disparity map of the left image using the MC-CNN [10]. Specifically, the cost volume filtering of [12] and semi-global matching method [13] were applied to the raw matching cost volume computed from [10]. We then recovered a depth map using calibration parameters (baseline and forcal length). A maximum disparity range was set to 228.

3) Confidence Estimation: We used two methods [14], [15] to obtain the confidence map. For approximately 400,000 pieces (taken from fall 2015 to summer 2016), the confidence maps were obtained by [14], while the remaining image pairs

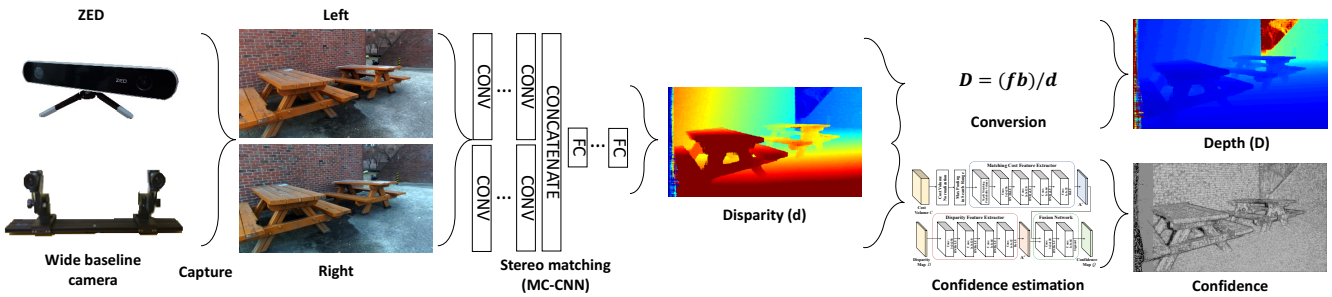


Fig. 7. The processing pipeline for our outdoor dataset. We captured stereo images, and then generated depth maps using the MC-CNN stereo matching [10], as well as per-pixel confidence maps.

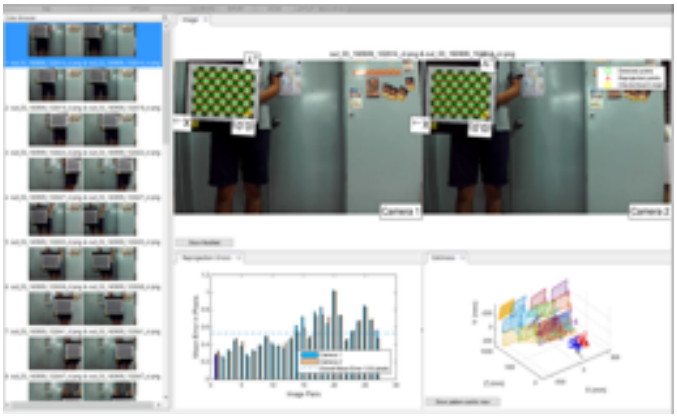


Fig. 8. Stereo camera calibration. We performed camera distortion correction and stereo image rectification.

(taken from fall 2016 to summer 2017) were processed by [15]. In [15], we devised a new deep architecture that estimates a stereo confidence. The confidence map is normalized in [0, 255], and the higher is the better. We found that with the threshold of 0.9×255, the majority of inaccurate depth values were removed. Refer to [14] and [15] for details.

C. Data Format

We stored outdoor depth maps in mm using a 16bit png format. The outdoor dataset includes the following files. Here, "**out**" denotes the outdoor data. "**zed**", and "**wb**" denote ZED stereo camera, and built-in stereo camera, respectively.

- Color image (1920×1080 or 1280×720)
  "**out_zed(wb)_14-digits_left.png**"
  24bit rectified left RGB image
  "**out_zed(wb)_14-digits_right.png**"
  24bit rectified right RGB image

- Disparity, depth, and confidence map (1920×1080 or 1280×720)
  "**out_zed(wb)_14-digits_disp.png**"
  8bit left disparity map
  "**out_zed(wb)_14-digits_depth.png**"
  16bit left depth map
  "**out_zed(wb)_14-digits_conf.png**"
  8bit confidence map

- Calibration parameters
  "**calib_zed(wb)_2-digits_6-digits.mat**"
  intrinsic and extrinsic camera parameters

Similar to the indoor data, each RGB-D image pair is identified by 14 digits. A total of four ZED cameras and single built-in camera were used. A file name includes an additional abbreviation, indicating the ZED stereo or built-in stereo cameras (for example, out_zed_03_160524_000001). The camera parameters vary according to the date (6-digits).

D. Outdoor data toolbox

- "**compute_conf.m**" - Computes the confidence map [15].

We obtain the confidence map of 400,000 stereo pairs using [14][1] and the rest using [15]. Note that [15] uses as inputs the cost volume and its associated disparity map, which were computed using [10] [2].

IV. Concluding Remarks

As future work, we will demonstrate that our dataset can indeed be used to successfully train large DNN for a wide range of computer vision and image processing applications. We anticipate that the scale and quality of our dataset could help bridge the gap between indoor and outdoor simulations, especially for depth-related tasks, e.g., monocular depth estimation, RGB-D feature learning, outdoor scene understanding, and dehazing.

References

[1] https://developer.microsoft.com/en-us/windows/kinect.
[2] https://www.stereolabs.com.
[3] https://cs.nyu.edu/∼silberman/datasets/nyu_depth_v2.html.
[4] https://github.com/code-iai/iai_kinect2.
[5] Y. Kim, B. Ham, C. Oh, and K. Sohn, "Structure selective depth super-resolution for RGB-D cameras," in *IEEE Trans. on Image Process.*, vol. 25, no. 11, 2016.
[6] http://www.cvlibs.net/datasets/kitti.
[7] https://www.matrix-vision.com/USB3-vision-camera-mvbluefox3.html.
[8] A. Segal, D. Haehnel, and S. Thrun, "Generalized-icp," in *Robotics: science and systems*, vol. 2, no. 4, 2009.
[9] N. Mayer, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016.

[1] http://cvl.gist.ac.kr/project.
[2] https://github.com/jzbontar/mc-cnn.

[10] J. Zbontar and Y. LeCun, "Computing the stereo matching cost with a convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015.
[11] http://www.vision.caltech.edu/bouguetj/calib_doc.
[12] K. Zhang, J. Lu, and G. Lafruit, "Cross-based local stereo matching using orthogonal integral images," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 19, no. 7, 2009.
[13] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, 2008.
[14] M. Park, and K. Yoon, "Leveraging stereo matching with learning-based confidence measures," *CVPR*, 2015.
[15] S. Kim, D. Min, B. Ham, S. Kim, and K. Sohn, "Deep stereo confidence prediction for depth estimation," *ICIP*, 2017.