# Three-Dimensional Integral-Imaging Display From Calibrated and Depth-Hole Filtered Kinect Information

Seokmin Hong, Adrian Dorado, Genaro Saavedra, Juan Carlos Barreiro, and Manuel Martinez-Corral

*Abstract*—We exploit the Kinect capacity of picking up a dense depth map, to display static three-dimensional (3D) images with full parallax. This is done by using the IR and RGB camera of the Kinect. From the depth map and RGB information, we are able to obtain an integral image after projecting the information through a virtual pinhole array. The integral image is displayed on our integral-imaging monitor, which provides the observer with horizontal and vertical perspectives of big 3D scenes. But, due to the Kinect depth-acquisition procedure, many depthless regions appear in the captured depth map. These holes spread to the generated integral image, reducing its quality. To solve this drawback we propose here, both, an optimized camera calibration technique, and the use of an improved hole-filtering algorithm. To verify our method, we performed an experiment where we generated and displayed the integral image of a room size 3D scene.

*Index Terms*—Bilateral filter, bilinear interpolation, camera calibration, integral imaging, kinect, median filter, 3D display.

#### I. INTRODUCTION

▼ ONVENTIONAL photography is fully adapted to record the 3D world scenes into a two-dimensional (2D) sensor. Although 2D images carry some cues from the 3D nature of scenes, they still lack important information. Fortunately, nowadays there are techniques that are able to record 3D information from 3D scenes. One interesting method is to record a depth map. A depth map can be obtained, for example, by the stereo vision technique, which takes profit from the disparity between the images captured with two cameras arranged horizontally [1], [2]. Other techniques are based on the projection of a random IR dot pattern [3], [4], or on time-of-flight technology [5]–[7]. Also interesting is to take profit from the vertical and horizontal views captured in with integral-imaging (InI) technology [8]-[12]. InI can provide 3D images in color with, quasicontinuous, horizontal and vertical parallax. For this reason it has been considered as one of the most promising technologies for next generation of 3D displays [13]–[18].

Manuscript received May 2, 2016; accepted July 21, 2016. Date of publication July 27, 2016; date of current version October 11, 2016. This work was supported in part by the Plan Nacional I+D+I under Grant DPI2015-66458-C2-1-R, Ministerio de Economía y Competitividad, Spain, by Generalitat Valenciana, Spain under Project PROMETEOII/2014/072. The work of S. Hong was supported by the University of Valencia under a predoctoral Grant UV-INV-PREDOC15-265754.

The authors are with the Department of Optics, University of Valencia, Burjassot E-46100, Spain (e-mail: seokmin.hong@uv.es; adrian.dorado@uv.es; Genaro.Saavedra@uv.es; barreiro@uv.es; manuel.martinez@uv.es).

This paper contains supplemental material available at http://ieeexplore.ieee.org.

Color versions of one or more of the figures are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JDT.2016.2594076

In a previous work we proposed the use of the depth map and a RGB image obtained with a Kinect to calculate an integral image and project it onto a 3D display system [14]. Although innovative, this research provided results that must be improved. The main problem of this previous research was the appearance of big holes in the depth map, which propagate up to the generated integral image. Another problem comes from the use the Kinect's software-development-kit (SDK). This method implements a mapping from IR camera to the RGB camera of the Kinect in order to merge the views of both cameras. But after applied, the SDK function produces some noise due to an error in decimal computation. In addition, this method can't make the mapping in the opposite direction.

In order to solve these problems, we propose some alternatives. First, we propose a new method for the camera calibration process between the two different sensors presented in the Kinect. This procedure is described in Section II. (Fig. 1(a) and (b)) Second, we propose to recover the lost depth information by using a filtering algorithm, which is explained in Section III (Fig. 1(c) and (d)). By applying these changes, we are able to obtain a better depth map, with lesser holes, due to the filtering and a better calibration process. With this improved 3D information, we generate higher quality of microimages. The microimages generation process is descripted in Section IV. Finally, in Sections V and VI, we provide experimental results and conclusions respectively (Fig. 1(e)).

## II. CALIBRATION BETWEEN THE TWO DIFFERENT TYPE OF CAMERAS OF THE KINECT

The Kinect device is well known for its capacity of capturing simultaneously, with two different types of camera, both, a color image and a dense depth map [19]. However, the field-ofview (FOV) of the two cameras are not matched properly. One solution to this drawback is the well known Camera Calibration Technique [20]–[25], which is able to correct the cameralens distortions, to figure out the focal length and to estimate the 3D location of a camera in real world coordinate system. Furthermore, this process can determine the correlation between the camera's own coordinate's unit (image's pixel coordinate) and the real world's measurement unit (millimeters, centimeters, etc.).

The calibration is a two-step process. First, in order to find a relationship between two cameras (that is, in order to obtain their intrinsic and extrinsic parameters) a special pattern must be captured. Second, the two captures are merged together using

1551-319X © 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications\_standards/publications/rights/index.html for more information.



Fig. 1. From *left* to *right*: (a) Captured raw color intensity image. (b) Processed image: coordinate conversion from color intensity camera to Infra-Red (IR) camera of the Kinect by using calibrated camera parameters with bilinear interpolation method. (c) Captured raw depth map image. (d) Image obtained using our proposed depth-hole filtering algorithm. (e) Final result: single-frame excerpt from the recorded video of the implemented integral imaging monitor.



Fig. 2. Sequential steps of the process to detect the chessboard pattern: the image is segmented into different parts. Otsu threshold is applied to each part and the results are accumulated. The procedure depends on the scene; therefore, we applied several segmentations to the captured scene, from 1 to 15 times flexibly. Upper row shows the results of original Otsu algorithm. Bottom row shows the result of adding accumulative procedure.



Fig. 3. Processed result using the proposed threshold technique. (a) Raw image from IR camera; (b) raw image from RGB camera; (c) thresholded image from (a); and (d) thresholded image from (b). Chessboard's corner points are found correctly even when the image has low illumination.

a transformation that takes into account the data previously obtained. Therefore, in order to calibrate we use a chessboard as the reference pattern for both cameras. The main reason of using a chessboard is that a regularized pattern improves the accuracy of the calibration [26]. It is worth to note that the image of the chessboard recorded with the IR camera is very dark, see for example Fig. 3(a) [27]. In order to overcome this drawback we propose to use a new algorithm that is based in the application of well-known Otsu thresholding method [28], but in iterative accumulative way. Our algorithm works as follows:

First, the image I is divided into i different parts, with a sequential ratio  $S_i = 1/i1/i$ . Then, Otsu algorithm, T, is applied to each individual part  $I_i I$ . Finally, the results are saved accumulatively into destination  $I_{dst}$ , as shown in Eqs. (1) and (2)

$$I_{\rm dst} = \sum_{i=1}^{n} \left[ \left\{ T\left( I_i \right) + I_{\rm dst} \right\} / 2 \right]$$
(1)

where

$$I_i = S_i\left(I\right) \tag{2}$$

The main feature of this new algorithm is that, independently of the complexity of the whole image, it highlights the chessboard pattern. Figs. 2 and 3 show this procedure in detail.

Once applied our algorithm, we can calculate the intrinsic and extrinsic parameters of the cameras. The intrinsic parameters are: the focal length, the aspect ratio and the central point of the view. The extrinsic parameters are the camera 3D location and orientation. We can use the values of theses parameters to fuse both cameras coordinate systems. The matrix Eqs. (3) to (5) rule the process to merge the FOV of the two cameras. In these equations K represents the intrinsic parameters; 2D point coordinate within each image is represented by p; P is each camera's



Fig. 4. From *left* to *right*: (a) Raw RGB intensity image; (b) calibrated RGB image from RGB camera to IR camera, (c) interpolated image's result and (d) and (e) raw mapping result in original and calibrated scene respectively. We can see that there are gaps (like as black stripes) in image (b). This is due to an error coming from the decimal computation procedures between each calibrated pixels. We can fix this by using a bilinear interpolation; the result of the interpolation is shown in (c). To attest this calibrated result, we make a mapping between the RGB image and depth map image. From the panel (e), we can check that the calibrated RGB image fuses to depth map image well.

coordinate 3D point. Finally R and T are rotation and translation matrices. These equations also permit to do an inverse mapping, so that it is possible to make the mapping (merging the FOV) between the two cameras in both directions

$$P_{\rm rgb} = \operatorname{inv}(K_{\rm rgb}) \times p_{\rm rgb}$$
(3)

$$P_{\rm ir} = R \times P_{\rm rgb} + T \tag{4}$$

$$p_{\rm ir} = K_{\rm ir} \times P_{\rm ir} \tag{5}$$

Unfortunately, there is still another problem related with the calibration process. The position of a pixel in an image is given in natural numbers. The final calibrated pixel coordinate (the one with the merged FOV) is represented by real numbers, which are rounded. This generates some gaps within calibrated pixels, see Fig. 4(b). As result, many pixels are misaligned in the calibrated image and therefore, into the 3D point cloud also same. To solve this problem, we applied a bilinear interpolation to the empty pixels between calibrated pixels; see Fig. 4 (c).

Finally, taking this into account, the calibrated RGB image can be mapped well to the IR image. Also, we are able to display in real time the calibrated images of both cameras. After all this calibration process, now it is possible to use the depth map and the RGB image to generate a 3D virtual point cloud in which we assign to each point its corresponding 3D position and RGB intensity. The Fig. 5 shows two views of a single shot of the virtual 3D point cloud corresponding to the recorded scene.

### III. DEPTH HOLE FILTERING

From the information captured with the Kinect we can compose a collection of microimages ready to be displayed on an InI monitor, as we showed in our previous paper [14]. However, that research had an important drawback. There were depthless pixels in the recorded depth map, which generated noise into the calculated microimages.

In the Kinect, the IR light source emits a known pattern and the depth information is calculated after comparison, by using triangulation method, between the known illumination pattern and the observed dots at the captured scene [29]. The problem arises when some reflective surfaces reflect IR light into another direction or when the IR light penetrates into transparent surfaces. This produces a loss in the depth information provided by the Kinect and thus, generates the holes in the depth map.



Fig. 5. 3D point cloud in the virtual space. From both panels, it is clear that each point has its own (x, y, z) position and RGB color intensity. Note that each 3D point is ordered on the basis on real world's measurement unit.

To avoid this drawback, we propose here the use of a depthhole filtering process based on Camplani and Salgado work [30]. In order to make the algorithm more efficient, we propose here some improvements on the original version of the algorithm. The key idea of Camplani and Salgado filtering process is iteration. In their proposal the depth map is captured several times. Every acquired depth-map frame is filtered in order to remove the spatial noise and purify the object boundaries. This filtered depth map is used to update both, the depth model and the filtering algorithm. Therefore, each acquired depth map increases the quality of the depth model and the applied filter. So, after any iteration more reliable depth information is obtained.

The flow chart of the hole-filtering algorithm, including our proposed improvement, is shown in Fig. 7. The real depth information D and the color intensity I are captured at every loop. A computed depth-map model  $D_{model}$  and consistent depth map  $C_{depth}$  are the core of this algorithm. The  $D_{model}$  is the result of applying the filter to the depth map and the  $C_{depth}$  is a version of the depth map that only stores the maximum depth values of all the iterations results.



Fig. 6. Comparison between original (1-(a)-(e), see also Media 1) and proposed (2-(a)-(e), see also Media 2) filtering algorithm: (a) is initial frame, (b) 5 iterations, (c) 20 iterations, (d) 80 iterations, and (e) 219 iterations. Through the panels, we can compare the process clearly. Above all things, the proposed strategy can recover the depthless pixel more efficient than the original method.



Fig. 7. Flow chart of the proposed hole filtering strategy.

After capturing the 3D scene information, the second step is collecting and classifying the depthless pixels by using the captured depth information and a computed depth-map model. If D has depthless pixels, they are replaced by the corresponding pixel of  $D_{\text{model}}$  if the pixel value is reliable (if  $C_{\text{depth}}$  is greater than threshold value  $d_{\text{thres}}$ ). And if  $D_{\text{model}}$  has depthless pixels, they are replaced by the corresponding pixel from D. Due to this change in information, D and  $D_{\text{model}}$  become D' and  $D'_{\text{model}}$ respectively. Note that on the first iteration, the value of  $D_{\text{model}}$ and  $C_{\text{depth}}$  are 0, and D' will be assigned with the value of D.

Next, the depth data is filtered using a joint (or cross) bilateral filter (JBF) [31], in order to improve the classified depth information's accuracy. JBF makes depth values reliable and is able to distinguish edges from surface's regions by checking and comparing neighbor pixels on both, the depth map and the RGB image. Note that JBF is an improved version of the similarity kernel of the bilateral filtering technique. The bilateral filtering is an edge-preserved and noise-reduced smoothing filter. To manage each pixel the filter has only two main kernel functions: the similarity kernel and the closeness kernel. These kernels are based on a Gaussian distribution and the pixel value is replaced by a weighted-average from their neighbor pixels [32].

The JBF works as follows; c(j, k) is the domain term like as bilateral filter,  $s(||D'_{model} - D''_{model}||)$  is the similarity kernel in classified  $D_{model}$  and  $s(||I^j - I^k||)$  is from the similarity kernel of color intensity. The scalar  $R^j$  is a normalization factor, and all of its calculated result is represented by  $D'_{filtered}$  [see, the Eqs. (6) and (7)]

$$D_{\text{filtered}}^{\prime j} = 1/R^{j} \iint_{k \in \Omega_{j}} D^{\prime k} c(j,k) s\left(\left\|D_{\text{model}}^{\prime j} - D_{\text{model}}^{\prime k}\right\|\right)$$
$$s\left(\|I^{j} - I^{k}\|\right) \tag{6}$$

where

$$R^{j} = \iint_{k \in \Omega_{j}} c\left(j, k\right) s\left(\left\|D_{\text{model}}^{j} - D_{\text{model}}^{k}\right\|\right) s\left(\left\|I^{j} - I^{k}\right\|\right)$$

$$\tag{7}$$

The fourth step consists on improving the previous filtered result. If  $D'_{\text{filtered}}$  still has some depthless pixels or regions, some of the missing depth information can still be recovered using all the data previously obtained.  $H(C_{\text{depth}}, \Omega_j)$  is a binary function that evaluates which pixels need to be updated with the information stored in D' and I. c(j, k) and  $s(I^j - I^k)$  are the same filtering functions as Eq. (6)

$$D_{\text{filtered}}^{j} = H\left(C_{\text{depth}}, \Omega_{j}\right) / R^{j} \iint_{k \in \Omega_{j}} D'^{k} c\left(j, k\right)$$
$$s\left(\left\|I^{j} - I^{k}\right\|\right) \tag{8}$$

where

$$R^{j} = \iint_{k \in \Omega_{j}} D^{\prime k} c(j,k) s\left(\left\|I^{j} - I^{k}\right\|\right)$$
(9)



Fig. 8. Collection of microimages generated from the 3D points cloud captured with the Kinect. These panels show about different focused planes; (a) is focused in 870 mm, (b) is 1290 mm, (c) is 1520 mm, (d) is 2385 mm, (e) is 3145 mm, respectively. Each focused plane shows an object with clear shape.

$$H\left(C_{\text{depth}},\Omega_{j}\right) = \begin{cases} 1 \text{ if count}\left[C_{\text{depth}} \left(\Omega_{j}\right) > d_{\text{thres}}\right] / \text{Area}\left(\Omega_{j}\right) > th_{\%} \\ 0 \text{ otherwise} \end{cases}$$
(10)

The fifth's step is to update the filtered depthless pixels into both,  $C_{depth}$  and  $D_{model}$ . Parameter  $\alpha$  is a constant weight factor whose value is obtained from our empirical evidence. The aim of this value is to obtain stability on the process, giving more importance to the previous results

$$D_{\text{model}}^{j} = \alpha D_{\text{filtered}}^{j} + (1 - \alpha) D_{\text{model}-\text{OLD}}^{j}$$
(11)

$$C_{\rm depth}^{j} = \begin{cases} D_{\rm model}^{j} \ if \ D_{\rm model}^{j} > C_{\rm depth}^{j} \\ \text{otherwise} \end{cases} .$$
(12)

The last step is the application of a median filter, which is our contribution to the process. It helps to expand reliable depth values into their neighbor pixels or clean up the noise in object's boundary/edge regions. The filter chooses the medium value between its neighbor pixels. For that reason, it can remove efficiently and correctly small rubbish particles and, as result,  $D_{model}$  and  $C_{depth}$  are updated and becomes a reliable filtered and computed result.

After a few repetitions using this proposed filtering process, we can get a clear hole-filled depth map. Fig. 6 shows the results of applying both, the original algorithm and the proposed one, to some specific frames. All the images in Fig. 6 correspond to the consistent depth map  $C_{depth}$ . Note that when we add the median filter, the small and big depth-hole regions were recovered more efficiently than in the original algorithm [30]. Also, instead of the traditional 256 depth scales used in the original paper we used a real depth scale of 3976 (chosen for empirical reasons). This means that we have more abundant depth information than the original one.

To finish this section we summarize the parameters used in the algorithm in Table I.

## IV. MICROIMAGES GENERATION

In order to generate the microimages, we follow a process equivalent to the one reported in our previous paper [14], but



Fig. 9. The overview of our experimental system. We moved the recording device vertically and horizontally to record different perspectives of the integrated image.

TABLE I Algorithm Parameters

$\delta$ for closeness filter c	4.5
$\delta$ for similarity filter s for the depth (0 – 3975 scales)	$9 \times 9$
$\delta$ for similarity filter s for the color (0 – 255 scales)	$9 \times 9$
$d_{ m thres}$	5
$th_{\%}$	0.65
α	0.04
Median filter size	$7 \times 7$

adapted to a new display device. Specifically, in our experiment the InI monitor is composed by a Samsung SM-T700 (359 pixels/inch), and a micro-lenslet-array (MLA) consisting of 113 × 113 lenslets of focal length  $f_L = 3.3$  mm and pitch p = 1.0 mm (Model 630 from Fresnel Technology). The generated microimages are then composed by  $15 \times 15$  px, the gap between the microlenses and the display is fixed to g = 49.5 px, and the full size of the integral image is  $1695 \times 1695$  px.

The VPA used to capture the synthetic microimages is placed into the point cloud's coordinate system. Note that the position of the VPA will determine the reference plane. Then we project, using projection mapping, each point of the 3D cloud through each pinhole of the VPA to obtain the microimages, as in [33]. We resize the image to  $1597 \times 1197$  px to take into account the resolution of the display system (14.13 px/mm).



Fig. 10. Comparison between the original method (*Top-Left* image, and also Media 3) and the proposed method (*Top-Right* image, and also Media 4). We have highlighted some specific parts of the images. It is clear that the result obtained with our proposed method shows more abundant 3D information.



Fig. 11. Different views of the InI monitor in the vertical and horizontal direction, showing that the InI monitor has full parallax. The distance between different images is 29 mm. The total viewing zone is of  $58 \times 58$  mm. A video of the views is shown in Media 4.

In particular we show, in Fig. 8, the microimages calculated from a VPA situated at different positions. The reference plane position determines which parts of the 3D image are in front or behind the screen.

## V. DISPLAYED 3D IMAGE

Finally, the generated microimages are displayed onto our InI monitor. The MLA was properly aligned in front of the display system. The InI monitor displays and integrates the microimages

towards the observer's eyes. Thus, a binocular observer can see some parts of the displayed scene in front of the monitor and some other behind. However this full-parallax effect cannot be directly observed in a manuscript or even in a video. In order to demonstrate here this effect we proceeded as follows. First we replaced the observer by a monocular digital camera. The Fig. 9 shows our experimental system's overview. Then we obtained a collection of pictures after displacing horizontally and vertically the camera along a region of  $58 \times 58$  mm. With these pictures we composed a video in which the InI monitor was observed from different horizontal and vertical perspectives. The Figs. 10 and 11 show the experimental results with more clarity. As you can see in Fig. 10, the hole-filtered depth map generates better images, recovering some of the lost depth information in the original one. In Fig. 10, we have highlighted with color rectangles the areas where the differences are clearly shown. Finally, Fig. 11 shows the different perspectives, vertical and horizontal, of the InI display system.

#### VI. CONCLUSION

In this paper, we have reported how to generate improved microimages using manipulated 3D information, obtained with a Kinect device. For that, we use the camera calibration technique with bilinear interpolation method. Also, we have proposed an efficient hole-filtering algorithm to fill the depth holes, which appear in the depth map captured by the Kinect. Therefore, this well-refined depth information reduces the noise in the recorded 3D information. In order to project our synthesized 3D information onto an InI display system, we generate microimages by using projection mapping through a VPA. To demonstrate the utility of our proposal, we projected the microimages onto an InI monitor, providing different, depth-hole free and continuous horizontal and vertical perspectives to the observer.

#### REFERENCES

- Bumblebee2, (2006, Aug. 23). [Online]. Available: http://www.ptgrey. com/stereo-vision-cameras-systems
- [2] ZED, (2015, Feb. 18). [Online]. Available: https://www.stereolabs.com/
- [3] Microsoft Kinect v1.0, (2010, Nov. 4). [Online]. Available: http://www. xbox.com/en-US/xbox-360/accessories/kinect
- [4] Primesense Calmine 1.08 & 1.09, (2013). [Online]. Available: https://en. wikipedia.org/wiki/PrimeSense
- [5] Canesta sensor, (2002). [Online]. Available: https://en.wikipedia.org/wiki/ Canesta
- [6] Microsoft Kinect v2.0, (2013, Nov. 22). [Online]. Available: http://www. xbox.com/en-US/xbox-one/accessories/kinect-for-xbox-one
- [7] Intel Realsense, (2014, Jan. 6). [Online]. Available: https://software. intel.com/en-us/realsense/
- [8] X. Xiao, B. Javidi, M. Martínez-Corral, and A. Stern, "Advances in threedimensional integral imaging: Sensing, display, and applications," *Appl. Opt.*, vol. 52, pp. 546–560, 2013.
- [9] Lytro camera, (2011, Oct. 19). [Online]. Available: https://www.lytro.com/
  [10] PiCam: Pelican Imaging Camera, (Nov. 2013) [Online]. Available: http://www.pelicanimaging.com/
- [11] Raytrix camera, (2010). [Online]. Available: http://www.raytrix.de/
- [12] B. Javidi, J. Sola-Pikabea, and M. Martínez-Corral, "Breakthroughs in photonics 2014: Recent advances in 3D integral imaging sensing and display," *IEEE Photon. J.*, vol. 7, 2015, Art. no. 0700907.
- [13] S. Hong, D. Shin, J. Lee, and B. Lee, "Viewing angle-improved 3D integral imaging display with eye tracking sensor," J. Inf. Commun. Convergence Eng., vol. 12, pp. 208–214, 2014.

- [14] S. Hong *et al.*, "Towards 3D television through fusion of kinect and integral-imaging concepts," *J. Display Technol.*, vol. 11 no. 11, pp. 894–899, Nov. 2015.
- [15] A. Stern and B. Javidi, "Three-dimensional image sensing, visualization, and processing using integral imaging," *Proc. IEEE*, vol. 94, no. 3, pp. 591–607, Mar. 2006.
- [16] F. Okano, H. Hoshino, J. Arai, and I. Yayuma, "Real time pickup method for a three-dimensional image based on integral photography," *Appl. Opt.*, vol. 36, pp. 1598–1603, 1997.
- [17] J. S. Jang and B. Javidi, "Improved viewing resolution of threedimensional integral imaging by use of nonstationary micro-optics," *Opt. Lett.*, vol. 27, no. 5, pp. 324–326, 2002.
- [18] Y. Kim, K. Hong, and B. Lee, "Recent researches based on integral imaging display method," 3D Res., vol. 1, no. 1, pp. 17–27, 2010.
- [19] Specification of Kinect v1.0, (2013, Sep. 17). [Online]. Available: http://msdn.microsoft.com/en-us/library/jj131033.aspx
- [20] Camera calibration in OpenCV, (2000, Jun.). [Online]. Available: http://docs.opencv.org/doc/tutorials/calib3d/ camera\_calibration/camera\_calibration.html
- [21] Camera calibration toolbox from Caltech, (2004, Jul. 20). [Online]. Available: http://www.vision.caltech.edu/bouguetj/calib\_doc/
- [22] Robust multi-camera calibration, (2005, Jun. 7). [Online]. Available: http://graphics.stanford.edu/~vaibhav/projects/calib-cs205/cs205.html
- [23] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," in *Proc. IEEE 7th Int. Conf. Comput. Vis.*, 1999, pp. 666–673.
- [24] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, Nov. 2000.
- [25] P. Sturm and S. Maybank, "On plane-based camera calibration: A general algorithm, singularities, applications," presented at the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, Fort Collins, CO, USA, 1999, pp. 432–437.
- [26] B. Gary and A. Kaehler, *Learning OpenCV: Computer Vision With the OpenCV Library*. North Sebastopol, CA, USA: O' Reilly Media Press, 2008.
- [27] Infrared stream in Kinect v1.0, (2013, Sep.). [Online]. Available: http://msdn.microsoft.com/en-us/library/jj663793.aspx
- [28] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Sys., Man., Cyber.*, vol. 9, no. 1, pp. 62–66, Jan. 1979. doi:10.1109/TSMC1979.4310076
- [29] D. Alex Butler et al., "Shake'n'sense: Reducing interference for overlapping structured light depth cameras," in Proc. SIGCHI Conf. Human Factors Comput. Syst., 2012, pp. 1933–1936.
- [30] M. Camplani and L. Salgado, "Efficient spatio-temporal hole filling strategy for Kinect depth maps," in *Proc. SPIE Int. Conf. 3-D Image Process. Appl.*, 2012, pp. 1–10.
- [31] G. Petschnigg et al., "Digital photography with flash and no-flash image pairs," in Proc. Annu. Conf. Comput. Graph. Interactive Techn., 2004, pp. 664–672.
- [32] C. Tomasi, and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. 6th Int. Conf. Comput. Vis.*, 1998, pp. 839–846.
- [33] M. Martínez-Corral, B. Javidi, R. Martínez-Cuenca, and G. Saavedra, "Formation of real, orthoscopic integral images by smart pixel mapping," *Opt. Express*, vol. 13, pp. 9175–9180, 2005.



Seokmin Hong received the B.Eng. and M.Sc. degrees in digital and visual contents from Dongseo University, Busan, Korea, in 2012 and 2014, respectively.

From 2012 to 2014 he was working with the Institute of Ambient Intelligence Laboratory, Dongseo University, Korea. Since 2015 he has been working with the 3D Imaging and Display Laboratory, Optics Department, University of Valencia, Spain. His research interests include image processing, computer vision, and applied computer science.

Mr. Hong received the B.Eng. Extraordinary Award from Dongseo University in 2012.



Adrian Dorado was born in Spain in 1988. He received the B.Sc. and M.Sc. degrees in physics from the University of Valencia, Spain, in 2011 and 2012, respectively. Since 2010, he has been with the 3D Imaging and Display Laboratory, Optics Department, University of Valencia, Spain. His research interests include 3D imaging acquisition and display.

**Genaro Saavedra** received the B.Sc. and Ph.D. degrees in physics from the Universitat de València, Spain, in 1990 and 1996, respectively.

He is currently a Full Professor of optics at Universitat de València, Spain, where he is Coleader of the "3D Imaging and Display Laboratory." His current research interests include optical diffraction, integral imaging, 3D high-resolution optical microscopy, and phase-space representation of scalar optical fields. He has published on these topics about 70 technical articles in major journals and three chapters in scientific

books. He has published over 50 conference proceedings, including 10 invited presentations.

Dr. Saavedra received the Ph.D. Extraordinary Award for his Ph.D. work.

**Juan Carlos Barreiro** received the B.Sc. (honored with an Extraordinary Award) and Ph.D. degrees in physics from the Universitat de València, Valencia, Spain, in 1985 and 1992, respectively.

He is currently an Associate Professor at the Department of Optics, Universitat de València. In addition, he has carried out research at CIEMAT, Madrid, Spain, and as a Postdoctoral Research Fellow at INAOE, Puebla, México and at The Institute of Optics, University of Rochester, Rochester, NY, USA. His current research interests include optical diffraction, high-resolution optical microscopy, and integral imaging. Besides that, he is actively involved in promoting Physics and Optics Education through outreach activities.



Manuel Martinez-Corral was born in Spain in 1962. He received the M.Sc. and Ph.D. degrees in physics from the University of Valencia, Valencia, Spain, in 1988 and 1993, respectively.

He is currently a Full Professor of optics at the University of Valencia, where he is Coleader of the "3D Imaging and Display Laboratory." His research interests include resolution procedures in 3D scanning microscopy, and 3D imaging and display technologies. He has published over hundred technical articles in major journals, and pronounced more than

40 invited and five keynote presentations in international meetings. He is a Topical Editor of the IEEE/OSA JOURNAL OF DISPLAY TECHNOLOGY.

Dr. Martinez-Corral has supervised 12 Ph.D. theses, three of them received the Ph.D. Extraordinary Award. In 1993 the University of Valencia honored him with the Ph.D. Extraordinary Award. He has been a Member of the Scientific Committee for more than 20 international meetings. He is Cochair of the Three-Dimensional Imaging, Visualization, and Display Conference within the SPIE meeting in Defense, Security, and Sensing (Baltimore). In 2010 he was named Fellow of the SPIE.