# An interactive ray-tracing based simulation environment for generating integral imaging video sequences

Roger Olsson[a] and Youzhi Xu[b]

[a]Dep. of Information Technology and Media, Mid Sweden Univ., SE-851 70 Sundsvall, Sweden;
[b]Dep. of Electrical and Computer Engineering, Jönköping Univ., SE-551 11 Jönköping, Sweden

## ABSTRACT

The next evolutionary step in enhancing video communication fidelity over wired and wireless networks is taken by adding scene depth. Three-dimensional video using integral imaging (II) based capture and display subsystems has shown promising results and is now in the early prototype stage. We have created a ray-tracing based interactive simulation environment to generate II video sequences as a way to assist in the development, evaluation and quick adoption of these new emerging techniques into the whole communication chain. A generic II description model is also proposed as the base for the simulation environment. This description model facilitate optically accurate II rendering using MegaPOV, a customized version of the open-source ray tracing package POV-Ray. By using MegaPOV as a rendering engine the simulation environment fully incorporate the scene description language of POV-Ray to exactly define a virtual scene. Generation and comparability of II video sequences adhering to different II-techniques is thereby greatly assisted, compared to experimental research. The initial development of the simulation environment is focused on generating and visualizing II source material adhering to the optical properties of different II-techniques published in the literature. Both temporally static as well as dynamic systems are considered. The simulation environment's potential for easy deployment of integral imaging video sequences adhering to different II-techniques is demonstrated.

**Keywords:** integral imaging, simulation, rendering, ray-tracing

## 1. INTRODUCTION

Three-dimensional (3D) video systems have for decades been pursued as the video format of the future. Various approaches for providing a perceived depth have been invented [1]. Stereoscopic techniques that require specific user worn equipment have over the years received a lot of attention. Depending on which technique that is used factors such as color, time, space or polarization separately transfer different perspectives of a depicted scene to a viewers left and right eye, thereby allowing for binocular vision. However, none of these techniques have managed to provide enough fidelity for becoming a 3D video standard. This is mainly due to a few drawbacks all types share. The view point is established at the time of capture unless expensive eye tracking equipment is used. To perceive the scene depth the viewer must use special goggles. The splitting of the 3D presentation into display and goggles results in conflicting depth cues, causing eye strain and fatigue [1]. Autostereoscopic techniques on the other hand remove the requirement of user worn equipment for depth perception. Instead more complexity is incorporated into the the display and scene depth is perceived merely by viewing the display.

Integral imaging (II) is an autostereoscopic technique that also allows a user to change view point and thereby see different parts of the depicted scene. The II-technique stems from integral photography which was invented by G. Lippmann almost a century ago [2]. In its original form an II-camera relays incoming light through an array of lenses, which together spatially multiplex scene depth, into a capturing pixel array. Every lens provides information to a subset of pixels in the array, a so-called elementary image. Thus, an II is composed of a large number of elementary images. Each being a low resolution projection of the depicted scene. An II-display inverses this process and extracts the scene depth from the stored disparity using a lens system with similar properties as the one used by the camera. An example of a two-tier II-camera and a remotely located II-display is shown in Figure 1. For proper operation II relies on high spatial resolution of pixel arrays in both the camera and the display. Recent years of research into LCDs and CCDs has made II a promising candidate for future 3D video formats. A few II-based real-time video systems have even been demonstrated [3,4].

Further author information: (Send correspondence to Roger Olsson.)
Roger Olsson: E-mail: Roger.Olsson@miun.se, Telephone: +46 (0)60 14 86 98
Youzhi Xu: E-mail: Youzhi.Xu@ing.hj.se, Telephone: +46 (0)36 15 78 59

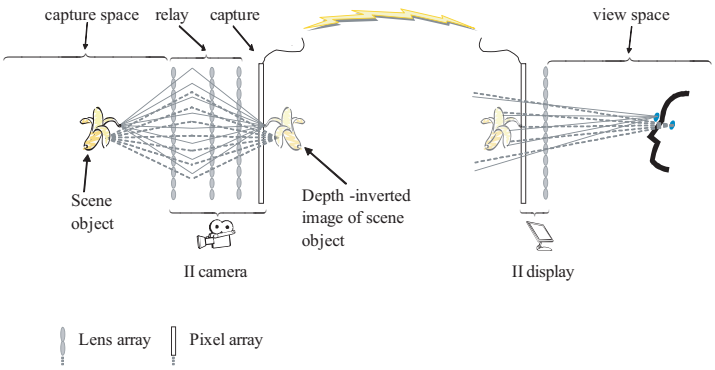


**Figure 1.** A two-tier system providing depth correct II.

Despite its good reputation II have a few unsolved problems that must be addressed in order for it to be a base for a future 3D video format. In its original form II produces pseudoscopic images, i.e. scene depth is inverted which result in convex objects looking concave and vice versa. To resolve this problem and present a depth correct image to the viewer different approaches have been taken. More complex lens systems, gradient-index (GRIN) lenses and electrical systems that rotate the elementary images 180 degrees around their centers are a few examples [1,5]. In Figure 1 a simplified version of such a lens system is shown where a two-tier lens array is used to make the II orthoscopic, or depth correct. Another problem compared to a 2D-display is that the field of view of a II-display is fairly low: in the order of tens of degrees [1]. Both non-planar displays as well as time dynamic display properties have been proposed to combat this limitation [6,7]. Effort has also been made to increase the perceived depth range and resolution, again mainly by introducing time dynamics into the camera and display properties [8,9,10].

In the work published so far result evaluation is mainly performed using the original form of II as reference. Evaluating any side-effects that two similar II-techniques might have while addressing a specific problem is more difficult. In the field of 2D image processing and compression such comparison operations are facilitated by:

1. A well defined quality metric, e.g. peak signal to noise ratio (PSNR).
2. A set of widely used reference images chosen as input signals.

Transferring this to the field of II would require the definition of at least one quality metric and a set of reference scenes. The definition of a quality metric has in part been addressed by Matthew C. Forman et al. [11] who uses the PSNR for assessing compression quality as a function of horizontal viewing angle for a II generated using a lenticular II-technique. To the authors knowledge no efforts have yet been made to define reference scenes though. Most likely because gathering and transfer exact knowledge of object size, position, color and texture as well as optics, lighting and environment properties is very which makes reproducibility of experiment almost impossible. However, by defining virtual reference scenes using computer simulation it is possible to achieve full control over the parameters mentioned. By doing that repeatability of experiments and accessibility of results are greatly enhanced. Even though work has been done to synthesize II, as well as synthesize views from II, those efforts have focused on specific II-techniques and not with the intent of generating II video sequences from a variety of II-techniques for easy deployment and evaluation [12,13].

Therefore, the aim of this project is to create a simulation environment that allows for an easy definition of arbitrarily complex reference scenes and from those synthesize integral images adhering to different II-techniques. In Section 2 we propose a basic generic II description model that provides a common base to which a given II-technique can be transformed. How this II description model, together with a description of a simple reference scene, is used to render II video sequences is presented in Section 3. Experimental results from a few different II-techniques are shown and conclusions about the simulation environment are drawn in Section 4 and Section 5 respectively.

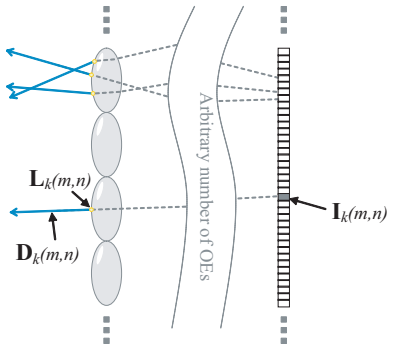


**Figure 2.** Active light rays with accompanying location points and direction vector.

## 2. A GENERIC II DESCRIPTION MODEL

The aim of our proposed generic II description model is to provide a common base to which various II-techniques can be transformed and used by the simulation tool, later described in Section 3, to render II video sequences of virtual scenes. The model construction is based on two main components:

1. An II-camera is comprised of a set of pixel arrays $I = \{I_0, I_1, \ldots, I_{K-1}\}$ with equal spatial resolution $M \times N$ and a set of optical elements $O$.
2. Geometrical optics is used to derive interaction between $I$ and $O$.

Both time static and time dynamic II-techniques are thereby possible, given that different subsets of the model can be active at different times. The model only cover II-techniques where the camera and display properties are principally equivalent, e.g. spatial resolution of the II and the number of lenses and their qualities. However, ways to transform an II to comply with fundamentally different properties of the display and camera have recently been presented and might be incorporated into future enhancements of the model [14].

When modeling a camera knowledge about the world to be depicted is beneficial. Adelson and Bergen [15] define a complete representation of the visible world using the plenoptic function

$$F = f\left(L_x, L_y, L_z, \theta, \phi, \lambda, t\right), \tag{1}$$

where $F$ corresponds to intensity of all light rays passing through space at location $\mathbf{L} = [L_x, L_y, L_z]^T$ with direction $(\theta, \phi)$ at time $t$ and with wavelength $\lambda$. Seeing that by sampling this function any type of camera or display could be modeled, this is the starting-point for deriving our model. Equation (1) can be simplified, without loss of generality, by transforming the light ray directions $(\theta, \phi)$ from spherical to Cartesian representation using

$$\mathbf{D} = [D_x, D_y, D_z]^T = [\sin\theta\cos\phi, \sin\theta\sin\phi, \cos\theta]^T. \tag{2}$$

Furthermore the intensity of all visible wavelengths $\lambda$ can be integrated into an approximate RGB-triplet $[R, G, B]^T$ according to an RGB Color Model [16]. These two operations together transform the plenoptic function into a more compact vector function

$$\mathbf{F} = f(\mathbf{L}, \mathbf{D}, t), \tag{3}$$

where the RGB-triplet $\mathbf{F}$ corresponds to the color of the sum of light rays passing through $\mathbf{L}$ from direction $\mathbf{D}$ at any time $t$. As stated above this continuous function must be appropriately sampled to illustrate the view from a certain camera type. In our model of a II-camera we first define a surface $S$ as the location points placed on the boundary between the camera and the capture space, i.e. on the outer edge of the optical elements $O$. Next, we define a set of active direction vectors $A$ as all direction vectors that passes through $S$, and that are aligned with light rays finally captured by a pixel array. In Figure 2 a subset of $A$ is illustrated which contain four light rays and their accompanying location points and direction vectors.

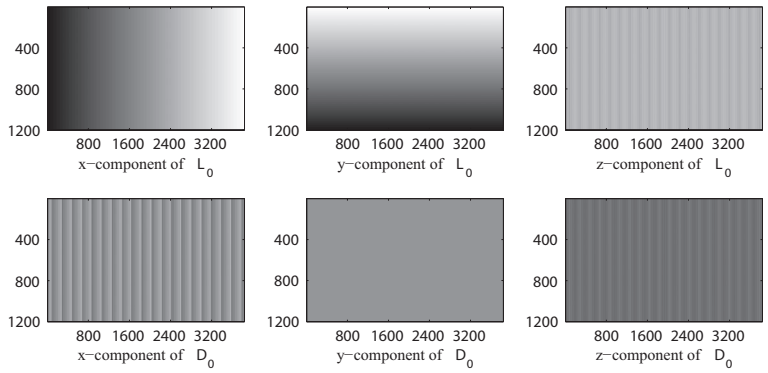


**Figure 3.** $\mathbf{L}$ and $\mathbf{D}$ for a lenticular II-technique.

As the figure illustrates, evaluating the plenoptic function at the surface of the pixel array is equivalent to evaluating the plenoptic function at the surface $S$; provided that the imposed restrictions on $\mathbf{L}$ and $\mathbf{D}$ mentioned above are applied.

Thereby

$$\mathbf{C}(t) = \sum_k \mathbf{F}(\mathbf{L}_k, \mathbf{D}_k, t)_{\mathbf{L}_k \in S, \mathbf{D}_k \in A} \tag{4}$$

describes the world that can be captured by an II-camera at any given time $t$. In the capturing process Equation (4) is implicitly made spatially discrete since each pixel array contains a finite number of pixels and by considering one light ray per pixel Equation (4) is non-uniformly sampled at $M \times N \times K$ $\mathbf{L}$ and $\mathbf{D}$ pairs. However, a larger number of light rays could also be evaluated per pixel resulting in super-sampled versions of $\mathbf{L_k}$ and $\mathbf{D_k}$. Thus, a critically sampled II-camera model is fully described by $M \times N \times K$ pairs of $\mathbf{L}_k(m,n)$ location points and $\mathbf{D}_k(m,n)$ direction vectors respectively, where $(m,n)$ corresponds to a pixel on the m:th column and n:th row of pixel array $k$. For modeling II-techniques with non-planar pixel arrays the geometry of the pixel arrays must also be explicitly defined which extends the II-camera model with $M \times N \times K$ pixel positions $\mathbf{I}_k(m,n)$. A planar pixel array is presumed to be located in the xy-plane and with the camera looking down the positive z-axis, i.e. the model is described using a left-handed coordinate system. For models with $K > 1$ one pixel array is defined as being primary, and the relative position and orientation of the other $K-1$ pixel arrays is stored as metadata. When the virtual II-camera described by the model is later used for rendering it is arbitrarily placed and oriented in the virtual scene using up-, right- and look-at-vectors [17]. By arranging the location point and direction vector pairs into $2 \times K$ pixel maps, with spatial resolution $M \times N$, the x-, y- and z-components of $\mathbf{L}_k$ and $\mathbf{D}_k$ is conveniently stored the red, green and blue channels of $K$ images using a 16 bit per channel image format. For II-techniques requiring large dynamics in specifying $\mathbf{L}_k$ and $\mathbf{D}_k$ the high-dynamic range image format RGBE is preferably used where one byte per channel is used together with a one byte shared exponent, i.e. 32 bit per pixel. For high precision, as well as increased dynamics, formats such as OpenEXR should be chosen since pixels can then then be stored using floating point representation. An example of a description model for a time-static lenticular II-technique with only horizontal parallax is shown in Figure 3.

How to actually perform the calculations of $\mathbf{L_k}$ and $\mathbf{D_k}$ is outside the scope of the model. Depending on the complexity of the II-technique, and the required fidelity of the generated II sequence, the calculations can range in complexity from piecewise linear transformations to full ray-tracing solutions.

## 3. SIMULATION TOOL

To make use of the model a simulation environment composed of two parts was constructed. An interactive tool with a GUI coupled to a rendering engine based on the open source ray-tracing package MegaPOV. The interactive tool provides access to:

- A generic scene description language for defining scenes of arbitrary choice, complexity and precision.

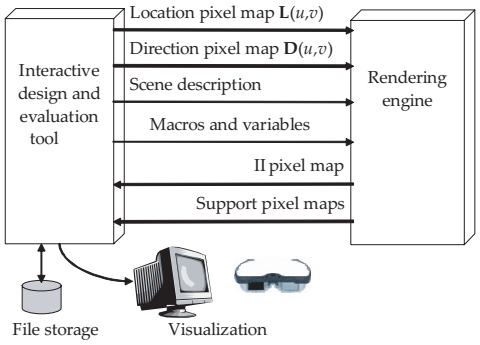


**Figure 4.** Block diagram of simulation tool.

- A common interface to generate and store II sequences as well as edit II description model and scene description data.

By using MegaPOV optically accurate rendering is achieved as well as access to the POV-Ray scene description language (SDL) for defining virtual scenes of arbitrary choice and precision. An example of a simple virtual scene is given in Appendix A, and further discussed in Section 4. More complex scenes designed especially to excite specific II properties could be imagined. To make use of the II description model described in Section 2 a set of extension macros was developed in the SDL to provide functionality such as metadata handling, oversampling in time dynamic systems, relative position and orientation of pixel arrays for $K > 1$ systems et cetera. In Figure 4 the information flow of the simulation environment illustrated where the model is constructed and transfered from the simulation tool to the rendering engine, together with a virtual scene description and metadata. After rendering II pixel maps are received back for visualization and storage. Supplementary information from the rendering process can also be received. A perspective projection overview of the virtual scene overview is one example. Another is ground truth scene depth provided on a pixel per pixel basis, useful for example when evaluating II-based depth-extraction algorithms or for post-processing of synthesized II.

## 4. RESULTS

A simple virtual scene was defined to evaluate the proposed model's ability to synthesize II video sequences adhering to different II-techniques. The description of the scene can be studied in detail in Appendix A, where the size, position, rotation and texture of three basic primitives are defined: a red cube, a green sphere and a blue cone. As time progress they move, with different velocities, further into the scene and away from the camera. Two II description models where used to generate the II video sequence and the result is presented in Figure 5. The first II-technique provides horizontal parallax only and is designed to use an IBM T221 LCD for display. The second adhere to the properties of an II-based TV-system presented by Okano et al. [1]. The parameters of the II-techniques are presented in Table 1 where the negative focal length of the second technique is a consequence of the GRIN-lens properties and indicate that it produces orthoscopic II. To emphasize the depth conveying properties of II orthographic projection was preferred which keeps the object size

**Table 1.** II-system parameters.

| Parameter | Lenticular II | Hexagonal GRIN-based II |
|---|---|---|
| Pixel array size [pixels] | 3840 x 2400 | 1920 x 1035 |
| Lens array size [lenses] | 1195 x 1 | 54 x 63 |
| Lens focal length [mm] | 2.2 | -2.65 |
| Lens index of refraction | 1.49 | variable |
| Elementary image size [pixels] | 3 x 2400 | 20 x 20 |

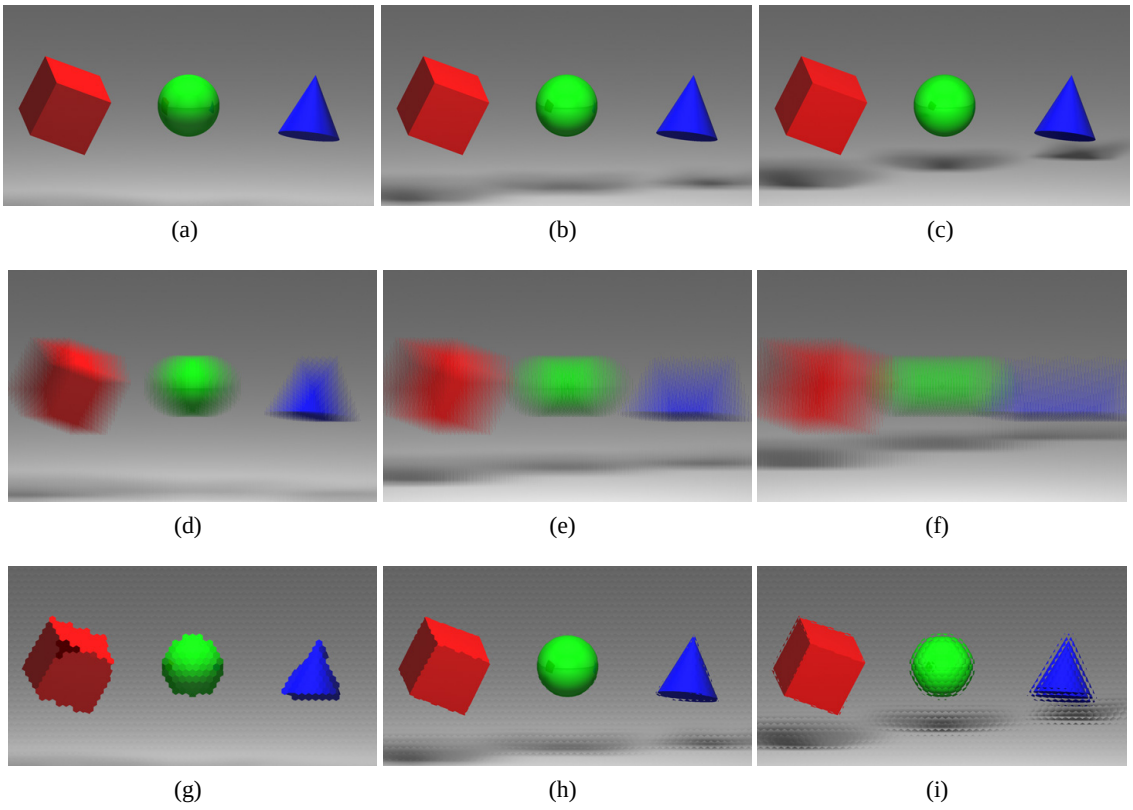


**Figure 5.** Generated II video sequences from (a)-(c) a 2D orthographic projection for reference, (d)-(f) a lenticular II downscaled to 50% and (g)-(i) a hexagonal GRIN-based II.

independent of object-camera distance. Objects spreading into more elementary images is then related to the II properties only which is verified in Figure 5. A large object-camera distance leads to an object being projected into a large set of elementary images. In Figure 5 (d)-(f) the lenticular II's ability to only provide horizontal parallax can also be observed as the lack of vertical spreading as the objects move further away from the II-camera. A more complex scene is synthesized in Figure 6 where a dolphin is depicted. The scene is composed of a low density mesh with 7765 vertices for the dolphin geometry and 6 accompanying texture maps for skin, eyes, teeth et cetera.

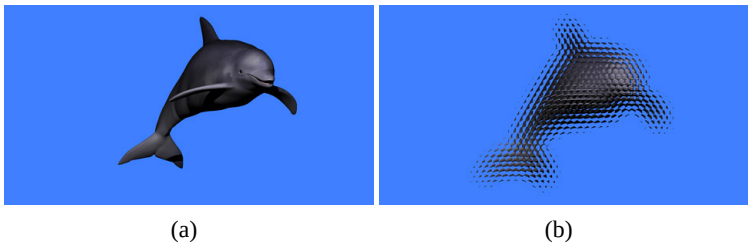


**Figure 6.** Generated II from (a) a 2D orthographic projection for reference and (b) a hexagonal GRIN-based II.

## 5. CONCLUSION

We have shown that the presented simulation environment, together with our proposed generic II description model, can be used to easily synthesize II video sequences adhering to various II-techniques. Due to the use of an open scene description

language different II-properties can be independently stressed and arbitrarily complex scenes can be depicted. By separating scene and camera definitions integral images from different II-techniques can easily be obtained by transforming the II-camera into the description model, i.e. as a set of location and direction pixel maps. Thus, compared to experimental research the simulation tool allows for generation of integral images adhering to different II-techniques in a feasible and cost effective way. In our future work our conceptually simple II description model will be extended to also handle effects such as optically introduced point spreading.

## APPENDIX A. VIRTUAL SCENE DEFINITION IN POV-RAY SDL

```
#version unofficial MegaPov 1.10;
// Include shape definitions and II support macros and -variables
#include "shapes.inc"
#include "shapes2.inc"
#include "iicamera.inc"
// Global definitions
background{ <0, 0, 0> }
global_settings { ambient_light <0.5, 0.5, 0.5> }
// II-camera placement
IICamera(<0, 0, 0>, <1.5, 0, 0>, <0, 1.5, 0>, <0, 0, 1.5>, clock)
// Texture definitions for objects and room walls
#macro ObjTex(Red, Green, Blue)
        pigment { color <Red, Green, Blue> }
        finish { ambient 0.5 diffuse 1 reflection 0.125 }
#end
#macro RoomTex()
        pigment { color <1, 1, 1> }
        finish { ambient 0.5 diffuse 1 }
#end
// Box, sphere and cone objects
box {   <-0.125, -0.125, -0.125>, <0.125, 0.125, 0.125>
        ObjTex(1, 0, 0)  rotate <-20, -20, -20> translate <-0.5, 0, 1*clock+1> }
sphere { 0, 0.125  ObjTex(0, 1, 0)  translate <0, 0, 1.5*clock+1> }
cone {  <0, -0.125, 0>, 0.125, <0, 0.125, 0>, 0
        ObjTex(0, 0, 1) rotate <10, 40, 0>  translate <0.5, 0, 2*clock+1> }
// Box room
box { <-10, -0.6, 10>, <10, 10, -10>  RoomTex()  rotate <-7, 0, 0> }
// Ceiling light
light_source {
        <0, 1, 0>
        color <0.75, 0.75, 0.75>
        area_light <1.5, 0, 0>, <0, 0, 1.5>, 10, 10
        adaptive 15
        jitter
}
```


## ACKNOWLEDGMENTS

The authors would like to thank Wlodzimierz ABX Skiba of the MegaPov-Team and Brad Paul at Kodak for helpful discussions. The work in this paper was supported in part by the Swedish Graduate School of Telecommunications (GST).


## References


1. E. B. Javidi and F. Okano, *Three-Dimensional Television, Video, and Display Technologies*, Springer, 2002.

2. G. Lippmann, "Epreuves reversibles," *Comptes rendus hebdomadaires des Séances de l´Acadéemie des Sciences* **146**, pp. 446–451, 1908.

**3**. J. Arai, F. Okano, H. Hoshino, and I. Yuyama, "Gradient-index lens-array method based on real-time integral photography for three-dimensional images," *Applied Optics* **37**, April 1998.

**4**. H. Liao, D. Tamura, M. Iwahara, N. Hata, and T. Dohi, "High quality autostereoscopic surgical display using anti-aliased integral videography imaging," in *Proc. MICCAI 2004, LNCS*, C. Barillot and D. Haynor, eds., pp. 462–469, Springer-Verlag Berlin Heidelberg, 2004.

**5**. M. McCormick and N. Davies, "Full natural colour 3d optical models by integral imaging," *Fourth International Conference on Holographic Systems, Components and Applications* , pp. 237–242, September 1993.

**6**. S. Jung, J.-H. Park, H. Choi, , and B. Lee, "Viewing-angle-enhanced integral three-dimensional imaging along all directions without mechanical movement," *Optics Express* **11**, June 2003.

**7**. Y. Kim, J.-H. Park, H. Choi, S. Jung, S.-W. Min, and B. Lee, "Viewing-angle-enhanced integral imaging system using a curved lens array," *Optics Express* **12**, February 2004.

**8**. R. Martínez-Cuenca, G. Saavedra, M. Martínez-Corral, and B. Javidi, "Enhanced depth of field integral imaging with sensor resolution constraints," *Optics Express* **12**, October 2004.

**9**. J.-H. Park, H.-R. Kim, Y. Kim, J. Kim, J. Hong, S.-D. Lee, and B. Lee, "Depth-enhanced three-dimensional-two-dimensional convertible display based on modified integral imaging," *Optics Letters* **29**, December 2004.

**10**. H. Choi, Y. Kim, J.-H. Park, J. Kim, S.-W. Cho, and B. Lee, "Layered-panel integral imaging without the translucent problem," *Optics Express* **13**, July 2005.

**11**. M. C. Forman, N. Davies, and M. McCormick, "Objective quality measurement of integral 3d images," in *Proceedings of SPIE Vol. 4660 Stereoscopic Displays and Virtual Reality Systems IX*, 2002.

**12**. T. Naemura, T. Yoshida, and H. Harashima, "3-d computer graphics based on integral photography," *Optics Express* **8**, February 2001.

**13**. G. Milnthorpe, M. McCormick, and N. Davies, "Computer modeling of lens arrays for integral image rendering," in *Proceedings of the 20th Eurographics UK Conference*, IEEE Computer Society, 2002.

**14**. J.-H. Park, H. Choi, Y. Kim, J. Kim, and B. Lee, "Scaling of three-dimensional integral imaging," *Japanese Journal of Applied Physics* **44**(1A), pp. 216–224, 2005.

**15**. E. H. Adelson and J. R. Bergen, "The plenoptic function and the elements of early vision," in *Computational Models of Visual Processing*, M. Landy and J. A. Movshon, eds., pp. 3–20, MIT Press, (Cambridge, MA), 1991.

**16**. R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Prentice Hall, 2002.

**17**. E. Lengyel, *Mathematics for 3D Game Programming & Computer Graphics*, Charles River Media, 2002.