

# Enhancing 3D Scenes Using a Laser Projector in Real-time

F. J. Menendez<sup>†</sup>, O. Halabi, T. Fujimoto and N. Chiba

Iwate University, Aomori University<sup>†</sup>

## Abstract

In this paper we present an innovative way to combine a laser (vector) projector with a video (raster) projector, using the former to enhance or augment the 3D images projected with the latter. As we will show, two main problems arise from this setting. One is to find a relation between the video projector space and the laser projector space, so that we can transform pixels into laser coordinates. The second problem is related to hiding virtually occluded laser segments to create a seamless fusion of the two projected parts. Also, we apply a calibration technique to compensate visual deformations caused by projector placement and/or non-planar screens.

## 1 Introduction

Although the concept of using lasers to project images has been around since the late 1960's, laser projectors have been mostly used as decorative props in entertainment venues and shows.

However, we think that laser projection technology has an important potential as a display device to be used in Computer Graphics [HMC\*11]. For this reason, in this paper we propose a method to use a laser (vector-based) projector in order to enhance or augment the three-dimensional images displayed by a video (raster-based) projector.

The motivation behind this work lies on the eye-catching nature of the laser beam itself. By combining these two technologies, we can make parts of the scene stand out more and call the attention of the viewer. We can overlay information of the 3D scene being shown, and we can even use it to simulate light sources such as light bulbs, neon signs, and even fireworks.

Thus, in this paper we will present a method to combine these two projectors: the video projector will display computer-generated 3D images, and the laser projector will enhance parts of those images. The goal is to do it in a way so we can seamlessly use the same set of 3D world coordinates for the objects we mean to display, regardless of the device they will be displayed on.

Furthermore, we will make the assumption that the projectors will not always be in a perfect position against the screen, or have a completely flat surface to display on. To address this issue, we will show a grid-based method to calibrate the projectors prior to displaying the images.

Finally, since both the laser and video projectors work in a completely different set of units (pixels for the video projector and vectors for the laser projector); we must find the necessary transformations to convert from the former units to the latter.

### 1.1. Laser projection background

Lasers projectors can be categorized as vector-based displays since they operate by drawing segments between pairs of x-y coordinates.

The simplest, most economic projectors usually employ

semiconductor-based lasers to generate the beam. This beam is then bounced off two perpendicular mirrors; one for the horizontal direction, and one for the vertical direction (x and y respectively). Each mirror is attached to its corresponding galvanometer.

By powering these galvanometers, the mirrors rotate and change the angle in which the beam reflects, ultimately repositioning the beam to a new x-y position. By repeating this process several thousand times per second, the projected laser dot is seen instead as solid lines of light on the projection surface, due to the persistence of vision effect.

Because of this mechanism, images displayed with laser projectors are infinitely focused and do not require any lenses to project. Furthermore, laser light is commonly brighter and has a wider color range than most video projectors.

Laser projectors usually connect to an outside signal in order to move the galvanometers. Projectors used in the entertainment industry generally use an audio signal. In this way, the galvanometers, laser color and intensity change along the rhythm of the music. More recently, however, projectors use a computer-generated signal. In this case, the positioning, color and intensity of the beam can be controlled to form specific shapes and animations.

### 1.2. Related study

Although there is no previous work regarding the combination of laser and video projectors, there has been significant work in regards to multi-projection and calibration for video projectors. Particularly, Raskar et al. [RWC\*98] [RWF98] [RBY\*99] [Ras00] [RBW\*03], have made significant advances in the 3D geometry-based method. In their study, a single (or stereo) camera is used to capture a predefined pattern that is projected on the screen by the video projector(s). This pattern is then detected and used to create a 3D mesh in projector space. Thus way, if the 3D configuration of the projection surface is known, the visual errors can be compensated by determining the position of the viewer. Calibration techniques developed by Johnson et al. [JGS\*07] [JF07] and Brown et al. [BMY05] are based on the method of Raskar et al.; that is, they essentially create a 3D configuration of the surface before applying the corrections. Using this 3D information, they can combine two or more projectors to create a wider display area.

In contrast, 2D image based methods [ZLB06] [OD06] [Sur95] do not use 3D geometry, but rather use visual cues and/or homographic transformations to perform geometric registration of the screen.

In our calibration method, presented in [MHC07] and [MHF\*09a], we also used a two dimensional, image based method to detect deformations on the surface. However, we increased the robustness of the method by introducing an iterative approximation algorithm.

Furthermore while, in a multi-projection setting, the goal is to tile the projection areas of each projector to create a large display surface, in this work we attempt to overlap the two areas (video and laser) into one in order to enhance the projected images.

## 2 Method Description

The method has two main processes. One process is the set-up step in which the system is calibrated. The second process involves using the values obtained in the set-up step in order to obtain a correctly combined projection.

The first process involves:

- Calibrating each projector independently to obtain a correct image regardless of the screen and/or projection angle.
- Obtaining a transformation (homography) matrix to convert from pixels coordinates into laser vectors.

The second process involves:

- Using the calibration data to project the images correctly
- Using the homography matrix to convert pixels into laser vectors when necessary.
- Hiding laser segments if they are virtually occluded by a 3D object displayed with the video projector.

The first process is done only once prior to displaying any images. This process does not need to be redone unless something changes in the original setting (for example, moving one of the projectors to a different position).

The second process is done once per each frame of the animation. Each step in this process is synchronous, meaning that one must finish before the next can start (for example, all 3D objects must be drawn before any laser displaying can happen).

### 2.1. Prerequisites

The only requirement is that the laser projection area fully overlaps that of the video projector. This is necessary to ensure that every pixel in the video coordinate system has a corresponding x-y coordinate in the laser coordinate system. (figure 1)

### 2.2. Calibration

Since we do not make any assumptions about the projector placement or type of surface the images will be projected on, we must first calibrate the two devices involved, prior to obtaining the coordinate-transform relation between them.

The laser projector is calibrated by using the method we developed in [MHC07]. Briefly put, the laser projector displays a grid pattern on the projection surface. This pattern is registered

with a digital camera, placed where the viewers would be (this is usually referred to as “sweet spot” calibration). Next, the computer uses the data taken by the camera to calculate an inverse version of the observed grid, that we call “calibration” grid

To perform the corrections, we run each vector through the calibration grid to obtain a set of smaller, calibrated vectors. These new vectors, when displayed on screen, compensate for the observed irregularities, forming a correct image when viewed from the “sweet spot”.

Since we need to display with the video projector on the same surface, we must ensure that any irregularities on the resulting projection are compensated as well. To do so, we simply took the same method and adapted it for the video projector. The process to obtain the grid is similar. However, to perform the corrections, we use a 2-pass rendering technique.

First, the image is rendered normally. The result, however, is not displayed but stored into a texture. This texture is then applied to a 2D grid made of quadrilateral polygons, formed by the values obtained on the calibration step. The shape of the grid deforms the texture, countering the perspective and surface deformations as observed by the camera.

Once both projectors have been calibrated, we store the calibration matrices for both projectors, so we can then search for the relation between video coordinates (pixels) and laser coordinates (vectors).

### 2.3. Find the video-to-laser relation

To find the video-to-laser relation, we use homography matrices [MHF\*09b]. A homography matrix establishes can transform point coordinates between two planes, using projective geometry. Using a camera as an intermediate plane, we can find the relationship between the projectors.

This setting is shown in figure 1. We project the four corners of both the video (gray) and laser (green) projector limits onto a flat screen. These points are then registered by a digital camera (black). Using this information (as seen below the camera), we can estimate the relation between the video and laser projectors as follows

First, we find the homography matrix between the laser projector and the camera

$$p_L \cdot H_L = p_C$$

where  $p_L$  is a point in homogeneous coordinates, in laser units,  $H_L$  is the laser-to-camera,  $3 \times 3$  homography matrix and  $p_C$  is the point  $p_L$  as seen from the camera, also in homogeneous coordinates.

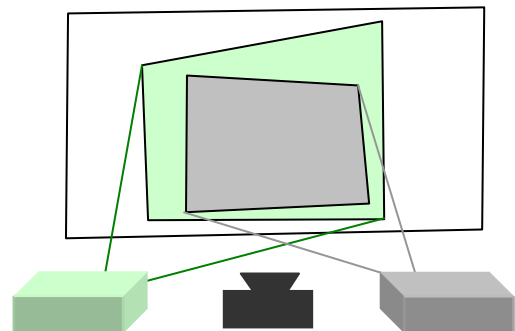


Figure 1: An example of a setting for the system. A laser projector area (green) fully overlaps a video projection area (gray). The two areas are then captured from a camera (black)

Similarly, we repeat the process for the video projector

$$p_R \cdot H_R = p_C$$

where  $p_R$  is a point projected from the video projector, in homogeneous coordinates,  $H_R$  is the video-to-camera homography matrix and  $p_C$  is the point  $p_R$  as seen from the camera

Since the points observed by the camera are in the same space, it's safe to assume that:

$$p_L \cdot H_L = p_R \cdot H_R$$

And since we want to find a way to transform pixels into laser coordinates, we do:

$$p_L = p_R \cdot H_R \cdot H_L^{-1}$$

If we call:

$$H_{RL} = H_R \cdot H_L^{-1}$$

Then:

$$p_L = p_R \cdot H_{RL}$$

where  $H_{RL}$  is the transformation matrix we need to go from the video projector plane to laser projector plane

There are different ways to compute the homography matrices based on matching pair of points in both planes [AJN05]. In our case, we simply used the homography functions from the computer vision library OpenCV [OCV], using four non-collinear points in each plane as input data. These four points are, in fact, the boundary corners of the projection area of each projector, and their corresponding matches as seen by the camera.

As we can see from the equations, the camera works only as a bridge between both coordinate spaces. Thus, it is not necessary to be in any particular position (i.e.: the camera can be set up in an arbitrary position respect of both projectors). It is a requirement, however, to be in a place where it can capture the projection extents of both projectors, with as much precision as possible.

Once the matrix has been found, the values are stored for later use and it does not need to be recalculated unless there is a change in position of either projector.

Now that we have our transformation matrix, we can go from the 2D coordinate system of the video projector to the 2D coordinate system of the laser projector.

## 2.4. Performing virtual occlusion

How do we, then, go from the 3D coordinate system of the application to the 2D coordinate system of the laser projector? Moreover, there is the question of how to properly hide sections of the laser images when occluded by virtual objects in drawn by the video projector.

To understand this problem, let us take a look at the example in figure 2. The setting on the left side of the picture is an example of a 3D scene we wish to display. The grey objects are to be displayed with the video projector, while the green star shape is

going to be shown with the laser projector.

Since the two projectors are unrelated, when we try to display the outlined star, it overlaps the rest of the scene, when in fact it should be partially occluded by the box in front of it.

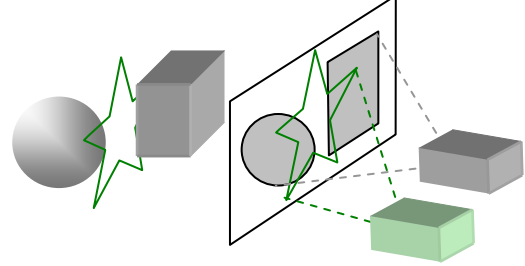


Figure 2: The laser images are projected on top of the images displayed by the video projector since no occlusion is calculated.

To solve this problem, we propose the following algorithm:

- Prepare the scene by performing all the necessary transformations (rotations, translations, perspective, etc)
- Render all raster objects and store all of the 3D matrices
- Retrieve the z-buffer from the scene.
- For every 3D segment  $s(xyz)$  we wish to display with the laser projector
  - Obtain the 2D coordinates of the projected segment  $s'(xy)$  and calculate its place in the z-buffer  $s'(z)$ .
  - Subdivide the  $s'$  segment into smaller sections  $s'(xy)_i$ ,  $0 \leq i < \text{number\_of\_sections}$
  - For each section, compare the stored z-buffer value with  $s'(z)_i$ . If  $s'(z)_i$  is bigger (further away) than the stored value, skip the section  $s'(xy)_i$ . Otherwise, transform  $s'(xy)_i$  to laser units using  $H_{RL}$  and display.

Before we begin to draw any laser lines, it is necessary to render all the primitives that are to be displayed with the video projector. We need to do this so we can have a valid z-buffer to work on later.

The first step towards correctly displaying the laser lines is to store the matrices used by the application to transform 3D coordinates into 2D coordinates (pixels) for the video projector. We will use these matrices later to transform the 3D segments we wish to display with the laser projector as well. Once we have stored these matrices we cannot perform any new rotations or translations until all laser segments have been processed and drawn.

The next step takes a pair of coordinates that define a single solid line to be displayed with the laser projector. These coordinates are expressed in 3D space. In this way, we can position the laser lines in the relative to the other objects in the 3D world. We use the matrices previously stored to project them into 2D. These new coordinates are expressed in pixels and are in the same space than the objects drawn with the video projector.

Next, we address the problem of occlusion by subdividing the segment into smaller parts. For each subsection, we obtain the un-projected z value for the end point of the segment, by performing the inverse calculation using the stored projection matrices. This z value is compared to the value stored in the z-buffer used when the raster image was created. Those segments that fail to pass the z-buffer test are discarded, creating the illusion of being "behind" the raster objects.

The segments that pass the z-buffer test are then transformed into laser coordinates, using the  $H_{RL}$  conversion matrix we calculated

on the first part of this work. These final coordinates are sent to the laser projector to be displayed.

It is important to observe that the occlusion test is only done for visible lines. For invisible movements of the laser mirrors used for positioning (also known as blanking), we just perform the projection to 2D, followed by the  $H_{RL}$  calculation, without any sub-segmentation taking place.

### 3 Results

For our experiment, we used an LCD-based video projector, at 1024x768 pixels of resolution and an ILDA-compatible laser projector. Figure 3 shows the setting of both projectors and figure 4 shows their display areas. As explained in section 2, the laser area fully overlaps that of the video projector.

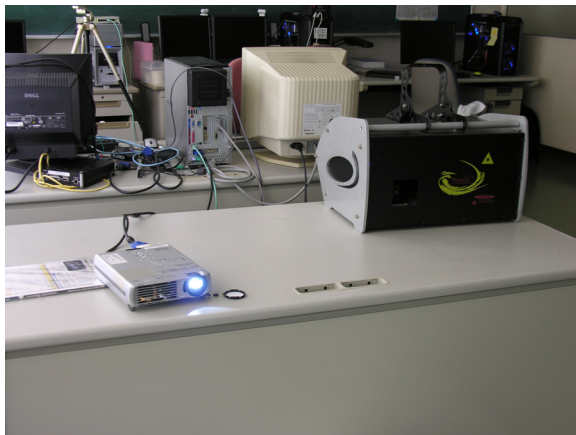


Figure 3: Projector setting. On the left, the video projector, and on the right, the laser projector

Next, we calibrated both projectors (figure 5) and obtained the homography matrix by establishing the relation between the corners of the display area of both projectors. The obtained calibration grids and homography matrix were then used to run two models: a motel sign and an airport runway.

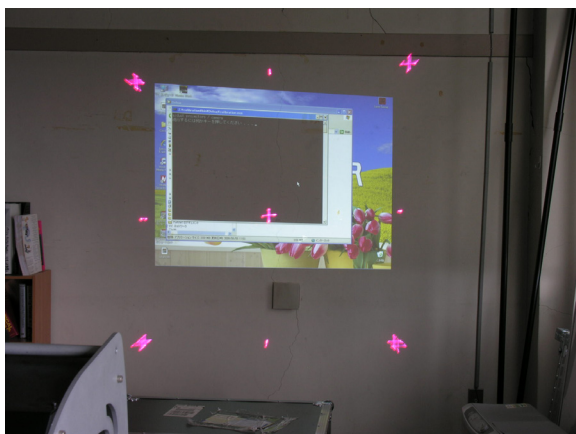


Figure 4: The laser projection area (magenta) is fully overlapping the raster area. Neither projector is calibrated at this point.

The motel sign example shows the occlusion solving capabilities of the algorithm. As the fence bars move in front of the sign, the neon sign projected with the laser gets properly hidden behind (figure 6).

Figure 7 shows a close up of the neon sign, with and without lasers. We used the same aperture and exposure to capture the difference in brightness of the two modes. The laser-enhanced example produces a much brighter neon sign and a more interesting visual effect.

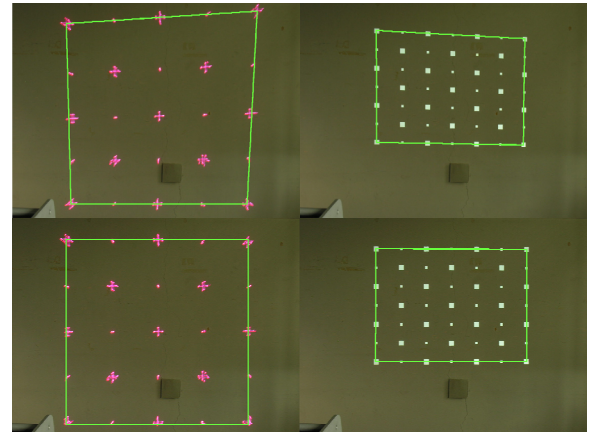


Figure 5: The two projection areas before (up) and after (down) calibration. Laser is on the left side and the video projector on the right side.

The second animation shows a simulated landing on an airport runway (figure 8). In this animation, the lights on the runway are displayed with the laser projector to show the potential it has to enhance projected video images.

The final animation (figure 9) shows an example of a moving character displayed with the laser projector, instead of moving the camera.

In all three cases, the models were designed using a regular 3D modeler (Blender). Both the 3D geometry and the laser sections are created in the same modeler and exported as separated objects. The program is later told which object should be used with which projector. This allows the modeler to work seamlessly in the same 3D space.

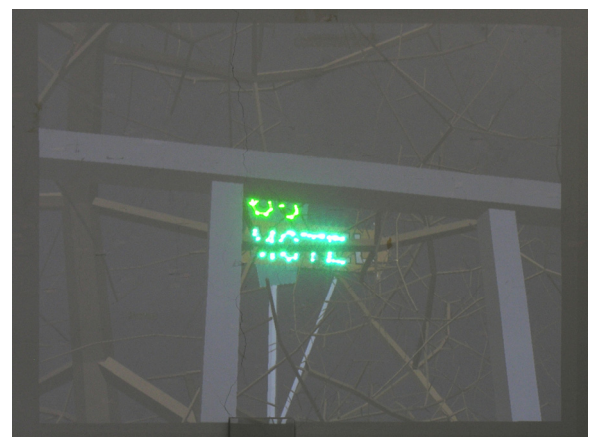


Figure 6: The neon sign (laser projected) gets properly occluded by the fence bar and tree branches (video projected)



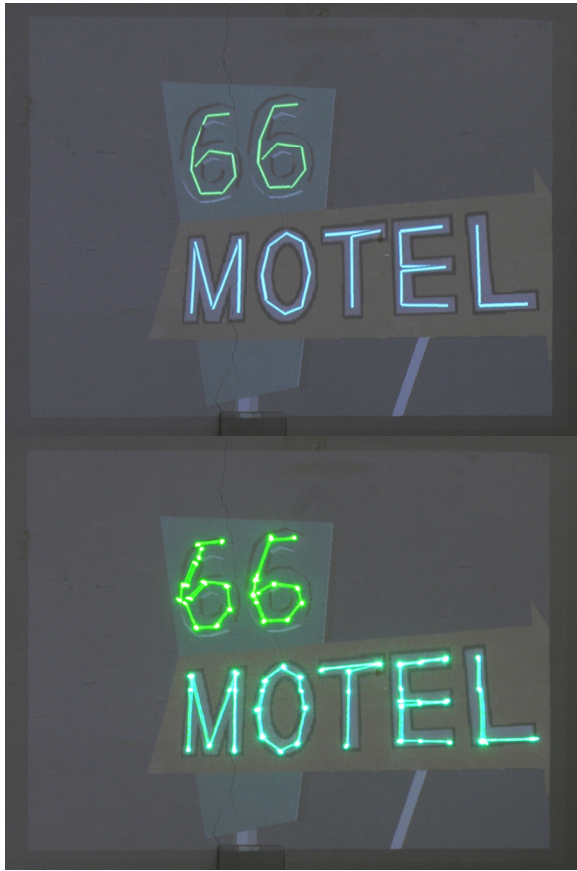


Figure 7: Close-up of the same neon sign, displayed using only a video projector (above) and the combined method (below).



Figure 8: Imitation of a flight simulator scene. The landing runway is lit up using the laser projector.

### 3.1. Performance

The method is fast enough to achieve real-time interactivity. However, due to hardware limitations in the laser projector, as the number of laser vectors increases, the performance drops.

This can be seen in table 1, where we show the results for the motel sign example, the fish example, and two settings of the airport runway.

Both the motel sign and fish example scenes contain a large amount of triangles to be rendered with the video projector. In both cases, the performance with the laser turned off is more than satisfactory for real-time, achieving over 35 frames-per-second in average.

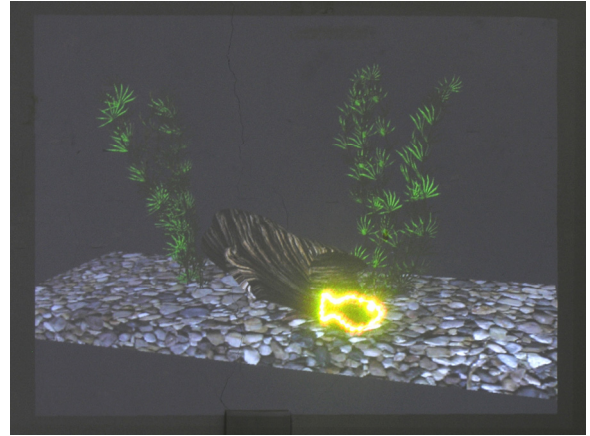


Figure 9: A moving character is drawn with the laser projector, while the rest of the scene is displayed with the video projector..

Table 1: Performance of the method in different examples

	Motel Sign	Airport Runway 1	Airport Runway 2	Fish
Number of triangles	20795	14	14	25128
Number of source vectors	82	194	502	76
Number of projected vectors	77.85	222.79	575.90	82.16
Average FPS (laser on)	21.76	10.97	4.57	17.98
Average FPS (laser off)	40.24	190.63	190.63	35.23

However, as we add the laser elements, we notice a drop in speed. This drop is not directly related to the original amount of vectors, but the number of vectors resulting after applying the method; namely, the vectors that are finally displayed with the laser projector.

In the motel example, the turning on-and-off of the sign and the amount of occlusion produced by the trees and fence, produce an overall lower amount of projected than in the fish scene. Since the drop in average FPS is related to this number, we can only assume it caused by speed limitations in the hardware.

This proves particularly true when we examine the runway examples. In this case, the geometry is fairly simple (only 14 triangles). We created two settings, one with 194 source vectors for the laser projector and another with 502 vectors. The second setting suffers a drastic drop in average FPS, while the first one barely manages to display a continuous animation.

In other words, the limitation in speed is directly related to the projection speed of the laser projector. This is reasonable,

considering the mechanical nature of its operation (as we discussed in 1.1).

## 4 Conclusions

In this work, we have presented a simple yet efficient way to augment a video projector using lasers. The results show that laser enhanced images are more eye-catching, since the brightness of the laser beam stands out from the rest of the image.

This gives laser projectors a great potential to enhance virtual light sources in the image, such as neon signs, light bulbs, highlights, etc. Laser overlays can also be used to call the attention of the viewer to specific sections of the scene.

We have also managed to make the calculations fast and simple, allowing for real time transformation of the coordinates between the projectors and opening the possibility for interactive applications as well; granted, being limited only by the speed of the hardware.

## References

- [AJN05] AGARWAL A., JAWAHAR C. V., and NARAYANAN P. J., 2005. A survey of planar homography estimation techniques. *Technical report, IIT-Hyderabad*.
- [BMV05] BROWN M., MAJUMDER A. and YANG R. 2005. Camera-Based Calibration Techniques for Seamless Multi-Projector Displays. *IEEE Transactions on Visualization and Computer Graphics*, March/April 2005 (Vol. 11, No. 2) pp. 193–206
- [JF07] JOHNSON T., and FUCHS H. 2007. Real-Time Projector Tracking on Complex Geometry Using Ordinary Imagery, *IEEE Conference on Computer Vision and Pattern Recognition*. p.1-8, June 2007
- [JGS\*07] JOHNSON T., GYARFAS F., SKARBEZ R., TOWLES H., and FUCHS H. 2007. A Personal Surround Environment: Projective Display with Correction for Display Surface Geometry and Extreme Lens Distortion. *IEEE Virtual Reality 2007*, March 2007
- [MHC07] MENENDEZ F., HALABI O. and CHIBA N. 2007. Camera-Based Calibration for Laser Projectors, *Tohoku-Section Joint Convention of Institutes of Electrical and Information Engineers 2007* p.3
- [MHF\*09a] MENENDEZ F., HALABI O., FUJIMOTO T., and CHIBA N. 2009. A Correction Algorithm for Distorted Laser Images. *The Journal of the Society for Art and Science* Vol 8-1 p.1-14
- [MHF\*09b] MENENDEZ F., HALABI O., FUJIMOTO T., and CHIBA N. 2009. A Hybrid Laser-Raster Projection System for Planar Screens, *Tohoku-Section Joint Convention of Institutes of Electrical and Information Engineers 2009*.
- [HMC\*11] HALABI O., MOTOMURA K., CHIBA N. 2011. Spiroraser: an interactive visual art system using hybrid raster laser projection, *International Journal of Arts and Technology*, Volume 4, Number 4, p.373-391
- [OCV] The Open Source Computer Vision Library. <http://sourceforge.net/projects/opencvlibrary/>.
- [OD06] OKATANI T., and DEGUCHI K. 2006. Autocalibration of an Adhoc Construction of Multiprojector Display Systems. *Proceedings of the International Workshop of Projector Camera Systems*.
- [Ras00] RASKAR R. 2000. Immersive Planar Display using Roughly Aligned Projectors. *Proceedings of IEEE Virtual Reality*, p. 109-116,
- [RBW\*03] RASKAR R., BAAR J., BEARDSLEY P., WILLWACHER T., SRINIVAS R., and FORLINES C. 2003 iLamps: Geometrically Aware and Self-Configuring Projectors. *Proceedings of ACM SIGGRAPH*, 2003.
- [RBY\*99] RASKAR R. BROWN, M., YANG R., CHEN W., WELCH G., TOWLES H., SEALES B., and FUCHS H. 1999. Multi-Projector Displays Using Camera-Based Registration. *Visualization Conference, IEEE*, p. 26, 1999.
- [RWC\*98] RASKAR R., WELCH G., CUTTS M., LAKE A., and FUCHS H. 1998. The Office of the Future: A Unified Approach to Image-Based Modeling and Spatially Immersive Displays. *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, p.179-188, July. 1998
- [RWF98] RASKAR R., WELCH G., and FUCHS H. 1998. Seamless Projection Overlaps Using Image Warping and Intensity Blending. *Fourth International Conference on Virtual Systems and Multimedia*, (Gifu, Japan).
- [Sur95] SURATI R. 1995. Scalable Self-Calibrating Display Technology for Seamless Large-Scale Displays. *Ph.D. Thesis*.
- [ZLB06] ZOLLMAN S., LANGLITZ T., and BIMBER O. 2006. Passive-Active Geometric Calibration for View-Dependent Projections onto Arbitrary Surfaces. *Workshop on Virtual and Augmented Reality of the GI-Fachgruppe AR/VR*, pp. 181-191.

## Authors' Biographies



FRANCISCO J. MENENDEZ is currently an Assistant Professor in the Faculty of Software Information Technology at Aomori University. His research interests include computer graphics, computer simulation and augmented reality. He received the B.E in computer Science from Mendoza University, Argentina and an ME in Computer Science from Iwate University in 2002 and 2006, respectively. He finished his PhD in Computer Science in 2009 at Iwate University. He is a member of IEEE



OSAMA HALABI is currently an Assistant Professor in the Department of Computer Science and Engineering at Qatar University. His research interests include human computer interaction, haptic interface, virtual reality, laser graphics, and computer graphics. He received a B.Sc., in electronic engineering from Damascus University, M.Sc. in computer science from Shanghai University, and Ph.D. in Information Systems from Japan Advanced Institute of

Science and Technology (JAIST) in 1992, 1998, and 2001, respectively. He was an assistant professor at Japan Advanced Institute of Science and Technology as a Fujitsu Endowed Chair from 2001 to 2003, a researcher in Virtual Systems Laboratory at Gifu University from 2003 to 2006, and an assistant professor in the Department of Computer and Information Sciences at Iwate University from 2006 to 2010. He is a member of VRSJ Japan, and ACM Computer Graphics.



TADAHIRO FUJIMOTO is currently an associate professor in the Department of Computer and Information Sciences at Iwate University. His research interests include computer graphics, geometric model, fractal theory, laser graphics and mathematics for shape description in general. He received a BE in electrical engineering, and an ME and Ph.D. in

computer science from Keio University in 1990, 1992, and 2000, respectively. He worked at Mitsubishi Research Institute from 1992 to 1995. He was a research associate in the Department of Computer and Information Sciences at Iwate University from 1999 to 2002, and a lecturer from 2002 to 2005. He is a member of SAS Japan, IEICE Japan, IPS of Japan, IEEE and ACM.



NORISHIGE CHIBA is currently a professor in the Department of Computer and Information Sciences at Iwate University. His research interests include computer graphics, algorithm theory and science on form. He received a BE in electrical engineering from Iwate University and an ME and DE in information engineering from Tohoku University in 1975, 1981 and 1984, respectively. He worked at Nippon Business Consultant Co., Ltd. from

1975 to 1978. He was a research associate in the Department of Communication Engineering at Tohoku University from 1984 to 1986, an associate professor of computer science at Sendai National College of Technology from 1986 to 1987 and an associate professor of the Department of Computer and Information Sciences at Iwate University from 1987 to 1991. He is a member of SAS Japan, IEICE Japan, IPS of Japan, IEEE and ACM.