SANDIA REPORT

SAND2015-7936
Unlimited Release
Printed September, 2015

3D Imaging with Structured Illumination for Advanced Security Applications

Gabriel C. Birch, Amber L. Dagel, Brian Kast, Collin S. Smith

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online ordering: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2015-7936
Unlimited Release
Printed September, 2015

3D Imaging with Structured Illumination for Advanced Security Applications

Gabriel C. Birch
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-MS1004
gcbirch@sandia.gov

Amber L. Dagel
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-MS1082
aldagel@sandia.gov

Brian Kast
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-MS1007
kast@sandia.gov

Collin S. Smith
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-MS0780
colsmit@sandia.gov

Abstract

Three-dimensional (3D) information in a physical security system is a highly useful discriminator. The two-dimensional data from an imaging systems fails to provide target distance and three-dimensional motion vector, which can be used to reduce nuisance alarm rates and increase system effectiveness. However, 3D imaging devices designed primarily for use in physical security systems are uncommon. This report discusses an architecture favorable to physical security systems; an inexpensive snapshot 3D imaging system utilizing a simple illumination system. The method of acquiring 3D data, tests to understand illumination design, and software modifications possible to maximize information gathering capability are discussed.

Contents

Appendix

Figures

Tables

1 Summary

The purpose of this project was to explore the potential for active 3D imaging systems that could enable real-time 3D imaging outdoors while maintaining eye safe levels of illumination. The uses of such a system are far reaching, but these qualities are particularly needed in the domain of physical security.

Several sub-topics have been investigated within this project:

- Method of collecting 3D information using structured illumination
- Fabrication and testing of 3D imaging system
- Illumination design enabling outdoor operation
- Software development to enable rapid 3D imaging
- Exploration and optimization of projected pattern

A robust method of single image 3D imaging was evaluated called pseudo random binary array (PRBA) 3D imaging. This method projects a unique binary matrix onto the scene of interest. Sub-windows within the pattern are non-repeating, and thus enable identification. This method was chosen because a single image of the pattern projected onto the scene contains all information necessary for 3D reconstruction. Thus, PRBA based systems can operate in a real-time 3D imaging configuration. A custom diffractive optical element was created to enable projection of the PRBA and tested in a laboratory and outdoor setting.

Designing an active illumination system capable of operating outdoors without being lost in the natural solar irradiance and maintaining eye safe levels of illumination is challenging. The operation within several near infrared absorption windows is proposed and transmission is analyzed. By filtering all but a narrow spectral window, and transmitting only hundreds of meters, illumination systems designed to operate in spectral absorption regions enable outdoor active 3D imaging systems that are eye safe.

Software is proposed that utilizes the unique nature of the PRBA to build confidence based on the number of known pattern elements extracted from the scene. Unique testing was developed to generate simulated imagery in software and evaluate complex edge cases.

While PRBAs offer many attractive benefits, they may not be the final solution. An exploration of the fundamental mathematics of projecting patterns onto arbitrary complex scenes is included. A general framework for rectifying any pattern incident upon a complex geometry is discussed, and examples of such a system are presented.

2 Introduction to structured illumination using pseudo random binary arrays

Structured illumination for 3D image reconstruction is similar to the use of stereo vision except one camera is replaced with a projector that projects a known pattern onto the scene. This approach has the benefit of overcoming some of the difficulties with the texture and reflectivity properties of the scene. Structured illumination takes advantage of known properties of the illumination to encode the scene (e.g., encode the pattern position) and as a result, simplifies the correspondence problem. As compared to other 3D imaging structured illumination approaches, no scanning or moving parts are required. Because of this, 3D scene reconstruction can be accomplished more quickly and with simpler hardware.

2.1 Advantages of pseudo random binary arrays

The use of structured illumination for the reconstruction of 3D scenes requires the acquisition of fewer images to enable 3D reconstruction. One of the primary advantages of PRBAs is that every sub-window of a given minimum unit cell for that array is unique. Practically, this means once the observed scene with the superimposed PRBA is indexed, image processing techniques can be used to recover the 3D position, orientation, and scaling of objects in the scene. From the illuminated pixels that can be located in the scene, a sub-section of the PRBA can be reconstructed and evaluated until a complete sub-window is found. Since each sub-window is unique, this enables full identification of the window absolute coordinates within the PRBA. This property enables knowledge of the absolute position in real space [1].

2.2 Construction of a pseudo random binary array

One approach to constructing a pseudo random binary array (PRBA) is to start with a De Bruijn (pronounced “de BRINE”) sequence. A binary De Bruijn sequence is a string of bits in which every string of a given length, n, appears exactly once. For example, given a length, $n = 3$, the length of the string is $2^n = 8$ and every string of length 3 (e.g. $000; 100; 010; 001; 110; 011; 101; 111$) occurs exactly once in this sequence. Note that there are multiple possible strings that meet this condition.

De Bruijn sequences can be constructed using several different methods, including a Hamiltonian path through an n-dimension De Bruijn graph, by concatenating, in lexiographic order, all Lyndon words whose length divides n, or by using shift registers [2,3].

In this project, a shift register was used to construct the De Bruijn sequence. The initial shift register is generated using a seeded random number generator. In our case we used the Matlab `rng('shuffle')` command in combination with `rand(·)`. The first seeds the random number generator. The seed can therefore be controlled to reproduce the same set of random numbers or a different set each time. Using `'shuffle'`, seeds the random number generator with the clock time so `rand(·)` produces a different sequence of numbers each time. However, when a shift register is used to produce a pseudo random binary sequence (PRBS), the same sequence will always result from a given m-bit shift register, so the PRBS is not truly random. The important characteristic is that the PRBS has an autocorrelation equal to 1 so that every sub-window is unique.

A PRBS of maximum length $n = 2^m - 1$ can be generated using an m-bit shift register. For example for a shift register length m = 3, the initial shift register may be 000 or 101 or any combination of binary m-bit sequences. The shift register is generated in Matlab as `shiftReg = floor(rand(k1,k2)+0.5)`. `rand(k1,k2)` produces a $k_1 \times k_2$ vector of uniformly distributed random numbers between 0 and 1. For a 1D array, $k_1 = 1$. The length of the shift register, is k_2 (= m).

To construct a 1D pseudo random binary sequence from the shift register, at each step, the bits of the shift register are shifted one place to the right. The right-most bit that drops off is placed at the end of the PRBS that is being generated. The open spot on the left-hand end of the shift register is filled by the sum mod 2, an XOR gate, of the two right-most elements of the shift register prior to the shift.

For example, if a shift register contains elements: $a_{i+3}, a_{i+2}, a_{i+1}$, and a_i at time i, then at a subsequent time $i+1$, the elements of the shift register are shifted right and the left hand element becomes $a_{i+4} = (a_{i+1} + a_i) \mod 2$. The remaining elements are a_{i+3}, a_{i+2}, and a_{i+1}.

An example construction of a PRBS from a 3-bit shift register, maximum length n = 7, follows: Let the initial 3-bit shift register be $1, 0, 0$

step	shift register	PRBS
0	1,0,0	
1	0+0,1,0 = 1,1,0	0
2	1+0,1,1 = 0,1,1	0,0
3	1+1,0,1 = 1,0,1	0,0,1
4	0+1,1,0 = 0,1,0	0,0,1,1
5	1+0,0,1 = 0,0,1	0,0,1,1,0
6	0+1,0,0 = 0,0,0	0,0,1,1,0,1
7	0+0,0,0 = 1,0,0	0,0,1,1,0,1,0

Notice that at the 7th step, the shift register has returned to $0 + 0, 0, 0 = 1, 0, 0$, the initial shift register. The sequence is said to have period equal to 7, the maximum period.

The generation of the PRBA is done by folding the PRBS and is based on the approach by MacWilliams and Sloane [4]. A $1 \times k_2$ PRBS is used to fill an $n_1 \times n_2$ array with a flat autocorrelation function. We have the following definitions:

$$\begin{aligned} n &= 2^{k_1 k_2} - 1 \\ n_1 &= 2^{k_1 - 1} \\ n_2 &= \frac{n}{n_1} \end{aligned} \tag{1}$$

To fold the PRBS for the construction of the PRBA, the elements of the PRBS are written down the main diagonal of the $n_1 \times n_2$ matrix (top left to bottom right) and then the PRBA continues to be filled from the opposite side whenever an edge is reached.

For PRBS $= a_0, a_1, a_2, a_3, ...a_p$,

PRBA =

$$\begin{vmatrix} b_{i,j} & b_{i,j+1} & \dots & b_{i,n_2} \\ b_{i+1,j} & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ b_{n_1,j} & \dots & \dots & b_{n_1,n_2} \end{vmatrix}$$

where the elements $b_{i,j}$ are arranged from the PRBS elements a_p by:

$$\begin{aligned} i &= p \mod n_1 \\ j &= p \mod n_2 \end{aligned} \tag{2}$$

If the length of the PRBS is $n = 15$, using the equations 1, we find

$n = 15 = 2^4 - 1$, which implies, $k_1 * k_2 = 4$, $k_1 = k_2 = 2$ (the 1 x 4 solution is not considered because we want an array)

$$\begin{aligned} n_1 &= 2^k_1 - 1 = 3 \\ n_2 &= \frac{n}{n_1} = \frac{15}{3} = 5 \end{aligned} \tag{3}$$

k_1 and k_2 define the subwindow size in PRBA, while $k_1 \times k_2$. $n_1 \times n_2$ defines the full PRBA array size. n defines the length of the PRBS to fill the array. n_1 and n_2 must be relatively prime, meaning the only common divisor between the two numbers is 1.

Every possible $k_1 \times k_2$ subwindow array appears exactly once in the full PRBA.

Examples of array sizes and subwindows:

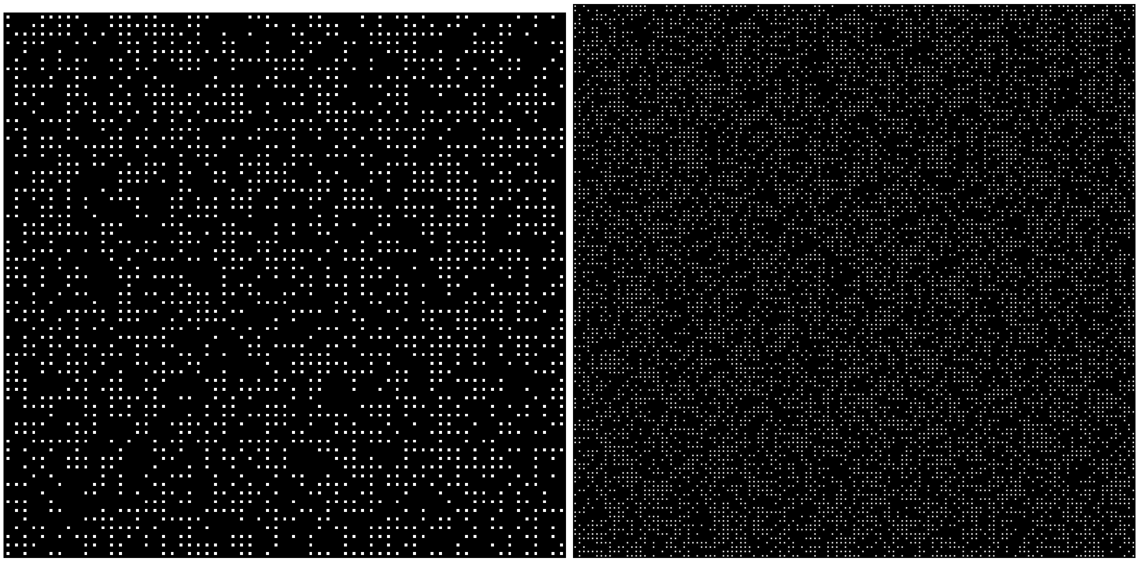
subwindow	PRBA size
6 ×2	63 ×65
7 ×2	127 ×129
9 ×2	511 ×513

2.3 PRBA fabrication

Binary PRBA bit patterns were provided to HOLOEYE Photonics AG, who completed the design, layout and fabrication of the diffractive optical elements (DOEs).

For ease of fabrication, the DOEs were designed to produce symmetric arrays; e.g. for each spot at [x,y] there is also a spot at [-x,-y]. No effort was made to suppress the undiffracted zeroth order diffraction spot in this demonstration. As a result it is the brightest spot in the scene. Additional DOE design would enable more efficient use of the illumination to only the locations of interest.

Two different array sizes were designed, one an array of 65 x 63 spots and the other 129 x 127 spots, the two arrays are shown in Figure 1 They were nominally designed to project a 1 m x 1 m array at 2 m away. This means the two arrays functionally provide different resolutions (spot densities) on target. The spot spacing in the scene directly impacts the resolution that can be achieved in the reconstructed scene.



(a) 65 x 63 spot PRBA.

(b) 129 x 127 spot PRBA.

Figure 1. PRBAs constructed in this project.

In the 65 x 63 spot array for a 1 m x 1 m field of view, the spot spacing is 15.625 mm. In the more dense pattern, 129 x 127 spots in the same 1 m x 1 m field of view, the spot spacing is 7.813 mm. The spacing in the x and y directions is identical, although this is a parameter that can be modified, which may be of interest in future implementations.

As these PRBAs were implemented as diffractive elements, they are fabricated by e-beam writing and reactive ion etching in 1.5 mm thick fused silica. They are optimized for one specific wavelength, in this case, for a helium-neon (HeNe) laser at 632.8 nm. The active area of the DOE is 6.4 mm, the diced size of the DOE is 8 mm and is mounted in a standard stainless steel holder with a 12.7 mm diameter. The anticipated diffraction efficiency of the elements is $70 - 75\%$. There is a 4% loss from Fresnel reflection off the flat surface, which could be mitigated by an anti-reflection coating in a future implementation.

It is interesting to observe that the chromatic aberration that would result by using a broad band source with the same DOEs could be used in the future to obtain higher resolution with the same physical optical element (same array spacing).

A binary PRBA matrix has neighboring elements in contact; in a fabricated array, the spots a DOE produces has spaces between projected “on” and “off” spots. If no beam shaping is implemented, the projected spots are round, not square. These characteristics make detection of the “on” verses the “off” spots more challenging since an “off” spot appears the same as a gap between spots.

2.4 PRBA experimental demonstration

To demonstrate the implementation of the PRBA approach requires first calibrating the camera so that intrinsic and extrinsic matrices of the camera-projector pair are known. The PRBA can then be projected onto the scene of interest. This is done by illuminating the small DOE with a HeNe laser. The DOE element is approximately 1 m from the scene. The camera is approximately 0.5 m from the DOE. The projected pattern is shown in Figure 2

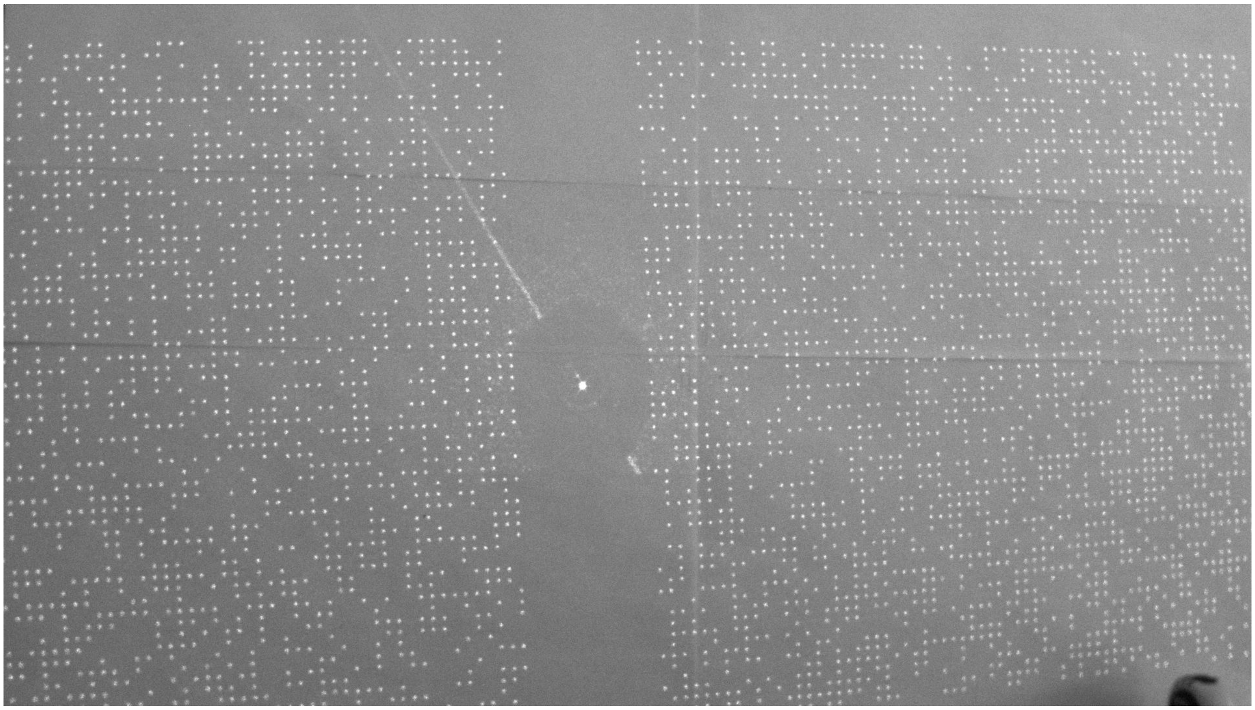
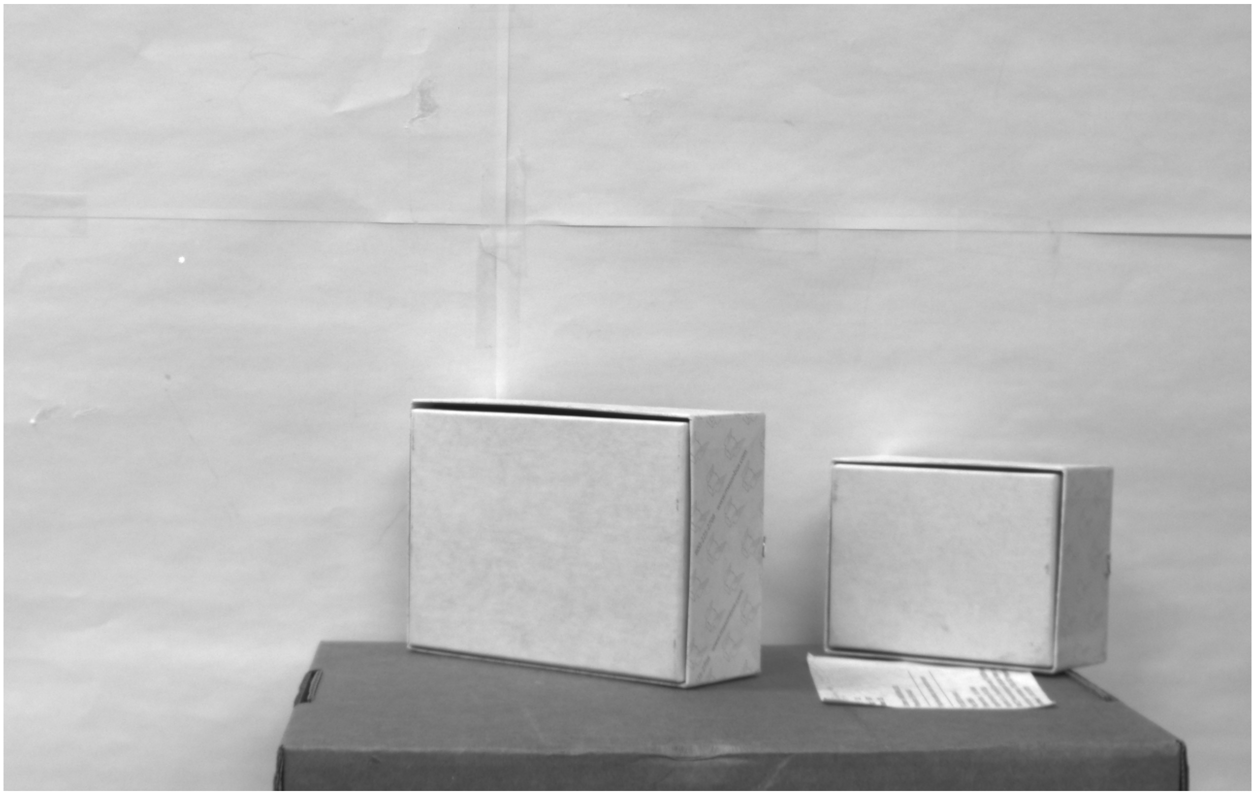
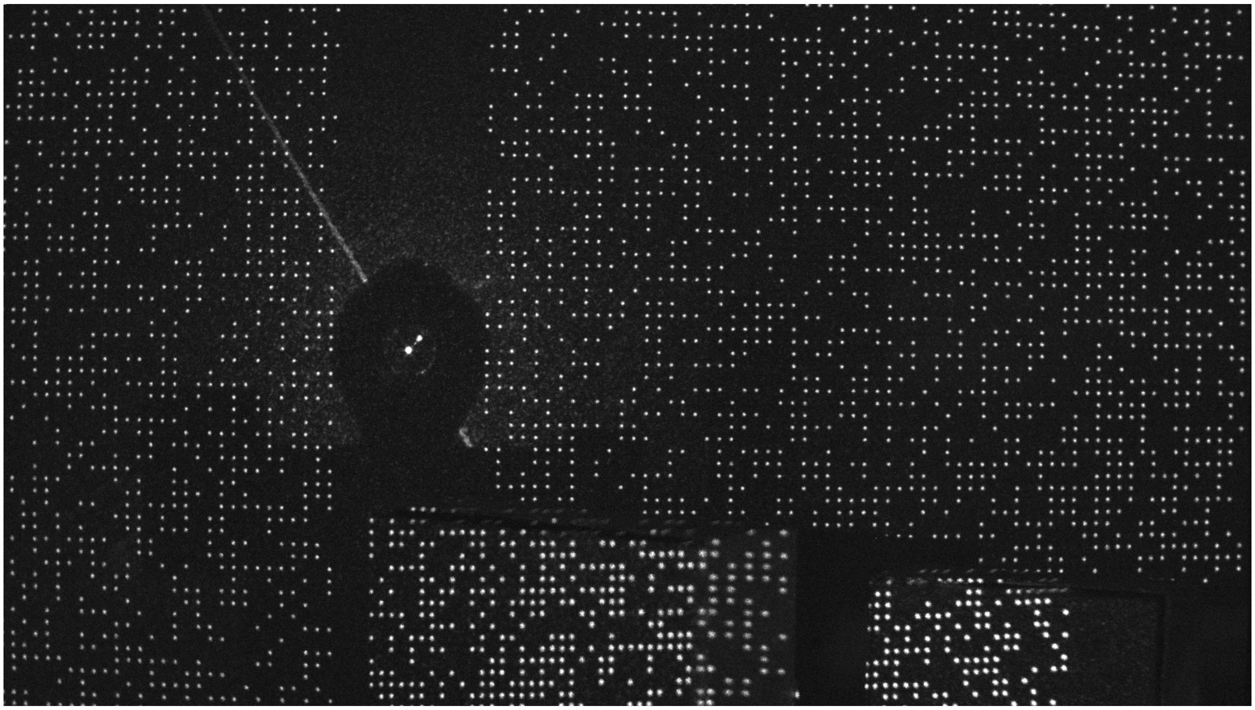


Figure 2. A portion of the pseudo random array of spots projected by the DOE. The brightest spot at the center is the 0th diffractive order, the left and right arrays are symmetric. For every spot at [x,y] there is a spot at [-x,-y], so every sub-window is unique to within the left or right array. This is not by design, but for ease of fabrication.

Various scenes were created to probe the dependencies on material and the depth resolution that could be achieved. The various scenes were constructed from cardboard boxes of various dimensions and the spacing between boxes was varied. A photo of a "constructed" scene is shown in figure 3 (a), with the same scene illuminated by the PRBA in figure 3 (b). Notice that even without reconstruction algorithms, changes in the spot pattern as compared to a flat scene allude to structure in the scene.



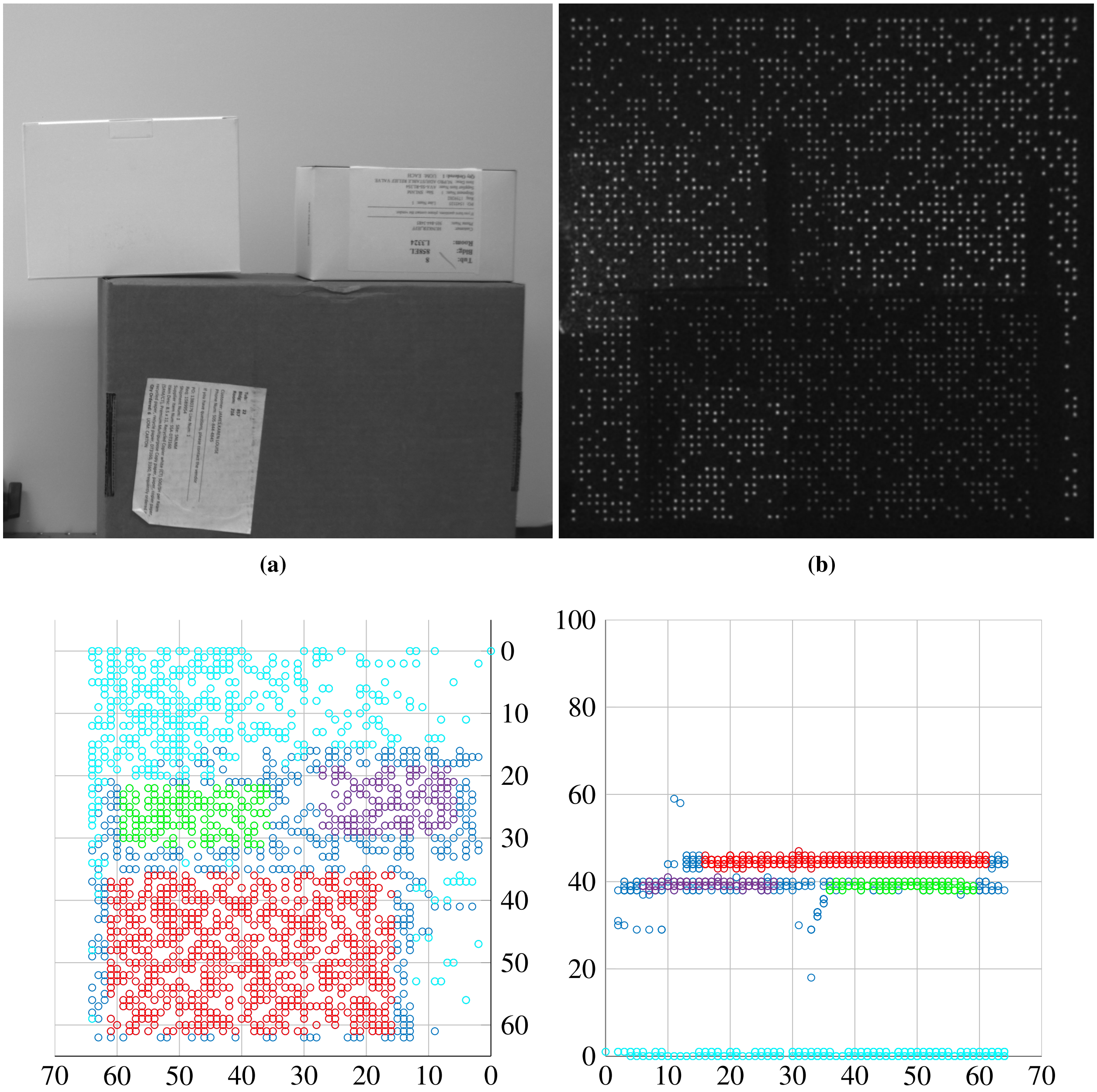
(a) Scene with two boxes at 45 degrees to a flat background.



(b) Scene illuminated with PRBA.

Figure 3. Experimental demonstration of PRBA incident on a complex scene.

Figure 4 shows the reconstructed 3D depth of sub-elements within the projected PRBA.



(a) (b)

(c) (d)

Figure 4. (a) visible image of a scene with several objects of differing depths and albedos, (b) image taken of PRBA pattern incident on targets in (a), (c) X-Y plane of extracted targets, with red circles being closest to the camera, and teal circles being farthest, and (d), X-Z plane of extract targets.

Table 1 shows statistics for the depth values measured from targets in figure 4.

Target color	Depth value mean	Depth value standard deviation
Red (closest box to camera)	44.7656	0.6694
Green (mid-range box 1)	38.9685	0.6291
Purple (mid-range box 2)	39.3077	0.5940
Cyan (wall)	0.2461	0.4313

Table 1. Statistics of measured depth values for targets shown in figure 4.

2.5 Future direction: Open challenges in the use of PRBAs

There are a few challenges associated with the use of PRBAs for 3D imaging, some are intrinsic to the field of structured illumination, some to PRBAs themselves. To take advantage of the uniqueness of subwindows, the pattern profile needs to be preserved, which places some demands on the material and lighting properties of the scene.

The challenge of global or ambient illumination is generic to the use of structured illumination. The operator is interested in the illumination he projects onto the scene, but this may be confounded by strong ambient illumination, which may further complicate the problem by causing shadowing or reflections. The challenges may be overcome by employing any number of techniques, which will depend on the optimization parameters of the particular application. A few of these techniques include single or multiple-thresholding, setting an albedo threshold (pixel-by-pixel thresholding), or dual pattern thresholding (acquire bright, all pixels "on," and dark, all pixels "off," images). This will be difficult or will have to be repeated in a situation where the global illumination is changing dynamically.

Other challenges that face structured illumination generally include occlusion, shadowing, non-lambertion BRDFs (transparent object, reflections, and objects in the scene that exhibit sub-surface scattering, such as glass or bottles of water).

Ignoring the multiple acquisitions that may be employed to accommodate ambient illumination, PRBAs enable reconstruction of a 3D image from a single snapshot of the scene. This means temporal characteristics of the scene can be captured through sequential acquisitions.

The PRBA method is dependent upon the identification of the unique sub-windows within the array. If a projected spot cannot be identified, the sub-window cannot easily be identified. Some of the challenges of the sub-window can be overcome by enhancing the spot-identification algorithm, with the obvious trade-off of increased complexity. A few of the scenarios that result in lost spot identification include too many "off" pixels in a row, sharp changes in depth, and particularly bright ambient illumination.

3 Outdoor illumination design

Two competing constraints are present when designing an active illumination system that must function outdoors:

1. Maintaining enough signal to overcome the natural illumination present in the environment of interest
2. Keeping the illumination eye safe

In regards to environments, we are particularly interested in both the controlled scenario presented by indoor 3D imaging, and the uncontrolled, high intensity illumination present in outdoor illumination during daylight hours. Since the outdoor problem is most challenging, we have focused our analysis on this domain, with the expectation that a successful outdoor concept would also solve the indoor problem.

The problem of actively illuminating a scene with more irradiance than the sun is a challenging task, especially if wide-band illumination is to be used. To reduce the complexity of this problem, narrow spectral band laser systems are frequently used instead. This, however, has its own set of technical problems, particularly with regards to eye safety. Figure 5 shows the ANSI Laser Safety Standard ANSI Z136.1 maximal allowed continuous wave powers for laser classes as a function of wavelength.

Figure 5. ANSI Laser Safety Standard ANSI Z136.1 maximal allowed continuous wave powers for laser classes as a function of wavelength.

The problem of low SNR could be solved by using a class 4 laser system with high power output, but this solution creates challenges in policy and safety areas. Instead, we propose to limit the system to the maximum 500 mW value permitted of a class 3B diffusing CW laser working over the 400 nm to 1700 nm spectral band.

In order to determine possible solutions, we present two analyses; examination of spectral irradiance values to determine spectral bands in which to operate an outdoor illumination system, and practical test results attained from testing a system outdoors.

3.1 Spectral irradiance analysis

In order to determine potential regions of interest within the solar spectrum, we first must examine spectral irradiance incident on the surface of the Earth. Figure 6 shows the greatest solar irradiance measured incident on the surface of the Earth at Albuquerque during August 15, 2013, to August 19, 2013, plotted in red. The ASTM SMART reference spectra for top of atmosphere solar irradiance is shown in green.

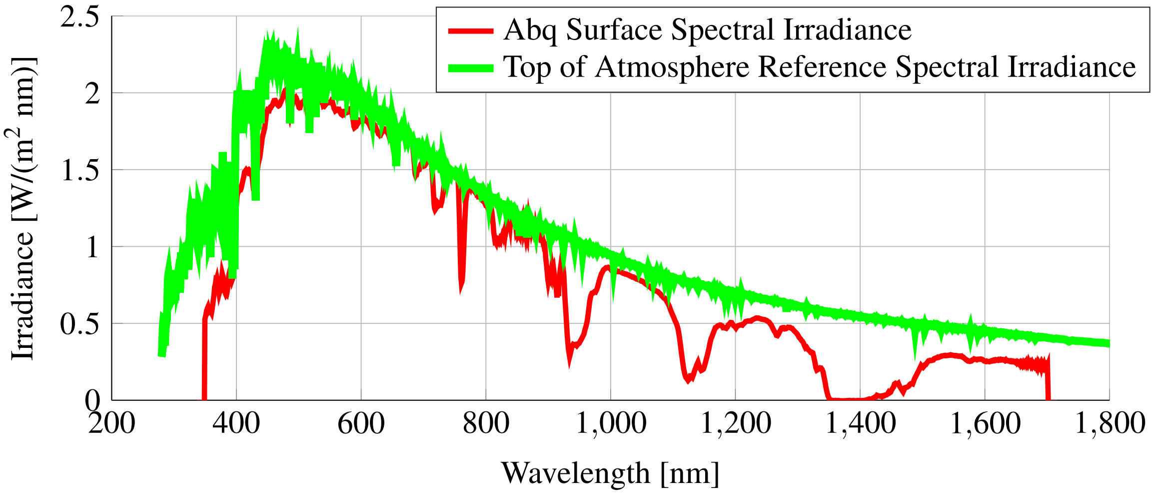


Figure 6. The red line shows the maximum incident solar spectral irradiance in Albuquerque, New Mexico, during a collection period of August 15, 2013, to August 19, 2013, while the green line shows the top of atmosphere reference spectral irradiance values from the ASTM SMART reference. Dips in the surface measured spectral irradiance occur due to absorbing constituents within the atmosphere, generally diatomic oxygen and water in the spectral region shown here.

Given the constrained maximum power of the active illumination system (i.e., limiting our conceptual design to class 3B laser illumination), operation in a region of minimized natural solar irradiance is necessary. Any operation over a broad bandwidth can be immediately removed from consideration. Operation in the 700 nm and longer domain reduces natural illumination, but particular interest is drawn towards the absorption dips at 760 nm due to O_2, and the 920 nm - 990 nm, 1105 nm - 1180 nm, and 1355 nm - 1530 nm regions due to H_2O absorption.

With the knowledge of the ground incident irradiance, and knowledge from an upper atmosphere reference spectral irradiance like the ASTM SMART reference spectra, it is possible to find the optical depth of the atmosphere,

$$\tau_\lambda = \ln\left(\frac{\text{Surface incident spectral irradiance}}{\text{Top of atmosphere spectral irradiance}}\right) \tag{4}$$

Assuming linear attenuation, we can find the attenuation coefficient using,

$$\alpha_\lambda = \frac{\tau_\lambda}{\text{Thickness of atmosphere}} \tag{5}$$

Given this estimated spectral attenuation coefficient, we can calculate transmission percent using the Beer-Lambert law. Since the proposed system is designed to operate much like typical physical security imaging systems, we specify an atmospheric path distance of 300 meters (150 meters double pass system). [1]

$$T\% = 100\% * e^{\alpha_\lambda * 300\text{ meters}} \tag{6}$$

Figure 7 shows the calculated transmission percent over the 400 nm to 1700 nm spectral band, given a 300 meter path length for the 3D imaging device.

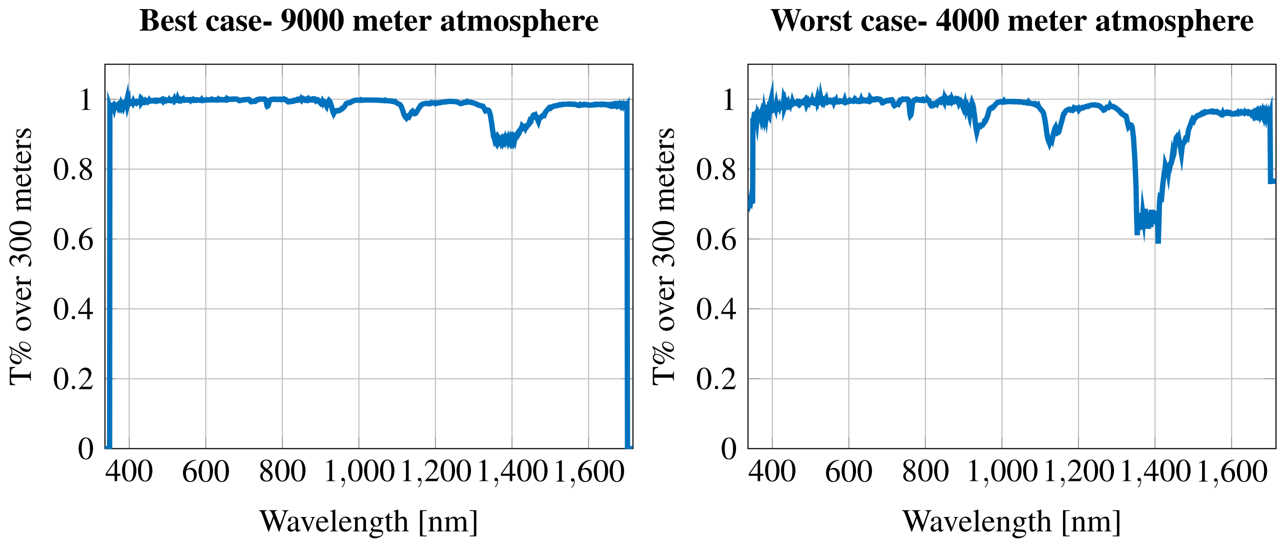


Figure 7. Transmission percent as a function of wavelength, using attenuation coefficients measured from Albuquerque, New Mexico, and the ASTM SMART top of atmosphere reference spectral irradiance. This analysis assumed a transmission distance for the theoretical system of 300 meters, and was performed for a best case thick atmosphere of 9000 meters, and a worst case thin atmosphere of 4000 meters.

The lowest transmission values are shown for the four absorption regions in table 2. Regarding the 1355 nm - 1530 nm absorption estimate, concern should be taken since this transmission value requires knowledge of the noise floor of the Albuquerque irradiance measurement, and may be substantially worse than this estimate.

---

[1] Note that the ASTM SMART top of atmosphere spectral irradiance was uniformly lower than the measured Albuquerque surface spectral irradiance, especially in the visible spectral domain. For the attenuation coefficient calculations, an offset value of 0.2 W/m^2nm was added to the ASTM SMART data so as to align the two data sets.

Wavelength Band	Absorber	T% at (300m)- best case	T% at (300m)- worst case
760nm	O_2	98.0%	95.3%
920nm-990nm	H_2O	95.6%	90.6%
1105nm-1180nm	H_2O	94.6%	87.4%
1355nm-1530nm	H_2O	87.7%	65.8%

Table 2. Transmission percent within the four primary absorption dips of the 400 nm to 1700 nm spectral band, given a 300 meter path length for the active illumination device.

3.2 Theoretical system

Solar illumination can be as high as 2 $\mathrm{W/m^2nm}$ in the visible spectrum, but absorption areas reduce the solar illumination background significantly. Table 3 shows the estimated solar radiation present in these absorption bands.

Wavelength Band	Absorber	Maximum Day Irradiance (MDI) per m^2	MDI per mm^2
760nm	O_2	0.7624 $\mathrm{W/m^2nm}$	0.7624 $\mathrm{\mu W/mm^2nm}$
920nm-990nm	H_2O	0.284 $\mathrm{W/m^2nm}$	0.284 $\mathrm{\mu W/mm^2nm}$
1105nm-1180nm	H_2O	0.129 $\mathrm{W/m^2nm}$	0.129 $\mathrm{\mu W/mm^2nm}$
1350 nm	H_2O	0.0122 $\mathrm{W/m^2nm}$	0.0122 $\mathrm{\mu W/mm^2nm}$

Table 3. Maximum day time irradiance at several absorption bands in the visible to near infrared spectral regions.

A pseudo random binary array diffractive optical element (DOE) can be constructed with varying numbers of pattern elements. Given a PRBA, a class 3B 500 mW laser, worst case transmission values as calculated in table 2, and the background illumination from table 3, it is possible to calculate the number of pattern elements needed to match the background illumination[2]. These values are shown in table 4.

- 500 mW laser
- 300 meter transmission in atmosphere using worst case assumptions
- 10 mm^2 laser spot size

---

[2] Explicitly, this calculation is sqrt((T% $*$ 500mW)/(Laser spot size in mm^2 $*$ Spectral window in nm $*$ Background solar irradiance))

- 1 nm spectral window

Wavelength Band	Number of PRBA elements
760nm	250 x 250
920nm-990nm	399 x 399
1105nm-1180nm	582 x 582
1350 nm	519 x 519

Table 4. Pattern elements for each absorbing band to equal background solar irradiance, given a 500 mW laser, estimated atmospheric absorption of this band over 300 meters, and a 10mm^2 laser spot size.

These results indicate the 1105 nm - 1180 nm spectral region is the best performing spectral band. However, care in choosing an imaging device which will be used to measure this near infrared region is necessary.

Silicon semiconductor has peak spectral sensitivity in the 700 nm to 900 nm spectral region, and sensitivity quickly decreases from 900 nm to 1100 nm. Therefore, it would be unlikely to utilize a silicon imaging system to image active illumination in the 1105 nm - 1180 nm spectral region. A switch to a NIR sensitive semiconductor, such as InGaAs, which is sensitive from approximately 900 nm to 1700 nm, would be required. InGaAs detectors remain expensive, on the order of tens of thousands of dollars per megapixel, and would likely be a significant cost to the overall system. For this reason the 920 nm - 990 nm spectral region should be considered. This band permits the use of inexpensive silicon semiconductors, and natural solar irradiance is reduced.

3.3 Test results

Because of availability of equipment and safety reasons, we have utilized laser systems at class 3R and below. A first proof-of-operation test was performed in an indoor lab. The following components were utilized:

- Point Grey Flea3 FL3-GE-28S4M, all exposure settings to auto
- 0.9 mW HeNe laser, 632.8 nm
- Thor Labs FLH633-5, 633 nm spectral filter with 5 nanometer full width half max bandwidth
- Roon lights on

The results of this test are included in figure 8, and show that a COTS silicon camera is sensitive enough to image the individual pattern elements using a low power narrow band source.

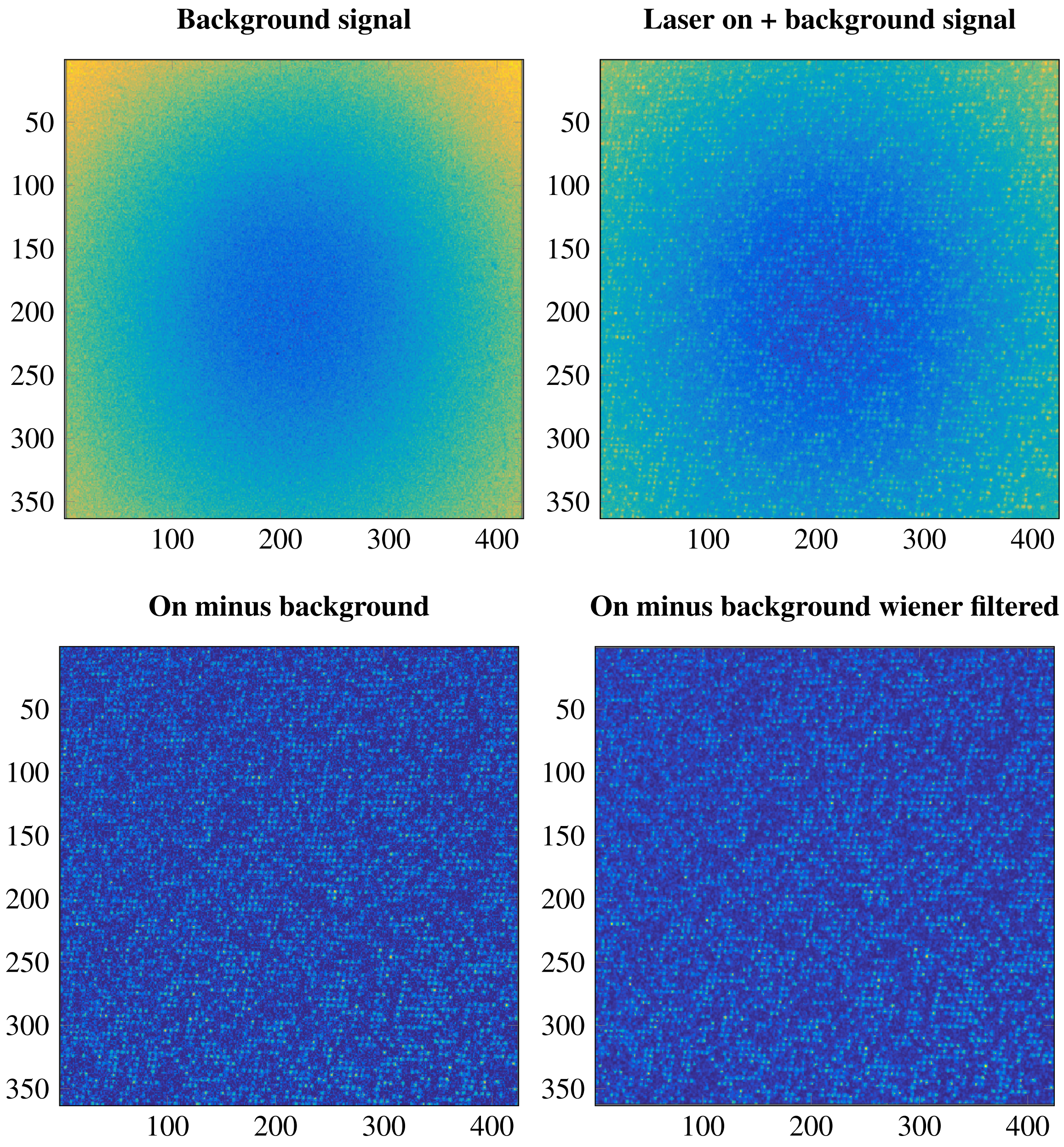


Figure 8. (a) Raw image with room lights turned on, (b) raw image with room lights turned on and laser turned on, (c) background subtracted raw image, and (d), image (c) with simple wiener filter de-noising algorithm applied.

Outdoor testing was performed to further verify illumination feasibility. The goal of this experiment is to test system performance at each spectral absorption band, but due to the costly nature of developing experimental DOEs for different wavelengths, we have chosen to use only the 632.8 nm wavelength (HeNe laser) and test the imaging system when the solar irradiance matches that of the bands of interest. Figure 9 shows solar irradiance at the HeNe wavelength over four consecutive days. Note that monsoon rains have caused most afternoon data to be unusable, but 8/16/2013 data remains relatively clear.

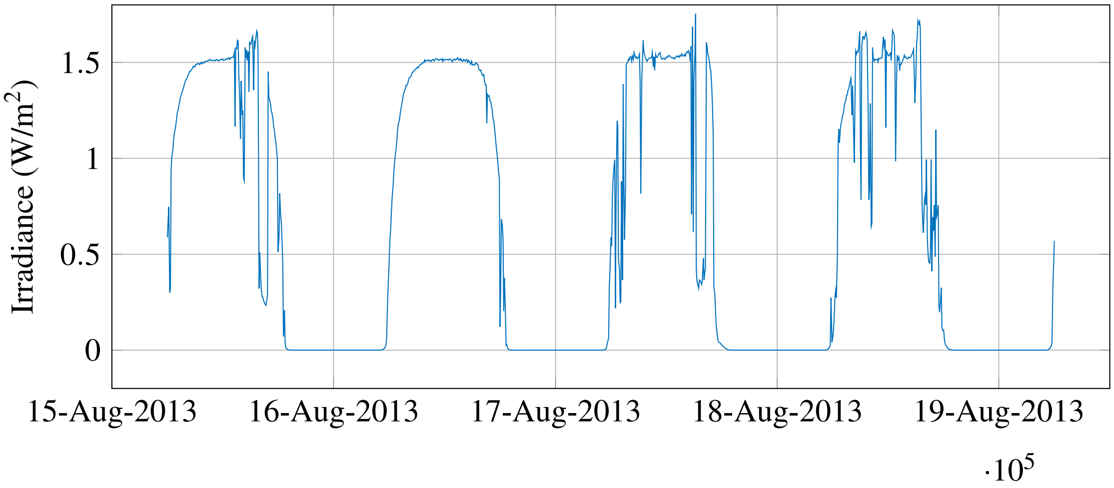


Figure 9. Irradiance at HeNe wavelength (632.8 nm) as a function of time in Albuquerque, NM.

Irradiance begins to drop at 3:30 PM and reaches the noise floor at approximately 7:00 PM. Table 5 shows the spectral region, maximum day irradiance measured in this band during the day, and the time of day at which the helium neon 632.8 nm spectral band contains the same irradiance values.

Wavelength	Maximum Day Irradiance (MDI)	HeNe Time
760.7 nm	0.762 $\mathrm{W/m^2nm}$	6:04:00 PM
934.2 nm	0.284 $\mathrm{W/m^2nm}$	6:32:30 PM
1124 nm	0.129 $\mathrm{W/m^2nm}$	6:37:00 PM
1350 nm	0.0122 $\mathrm{W/m^2nm}$	6:49:00 PM

Table 5. Maximum day time irradiance at several absorption bands in the visible to near infrared spectral regions and the corresponding time at which the Helium Neon laser 632.8 nanometer spectral band has the same irradiance, denoted as ‘HeNe Time’.

A test was performed on September 2, 2015, using the following equipment:

- Point Grey Flea3 FL3-GE-28S4M, all exposure settings to auto
- 2.3 mW HeNe laser, 632.8 nm
- Thor Labs FLH633-5, 633 nm spectral filter with 5 nanometer full width half max bandwidth
- Test target on wall of building (facing west)

Data was taken every five minutes, starting at 5:30 PM and ending at 7:00PM. Two major conclusions can be drawn from this outdoor test:

1. 2.3mW source enables pattern extraction when background subtracting
2. Several images must be averaged to reduce sensor noise

Figure 10 shows the difference between an average background and average illuminated target scene at approximately 7PM.

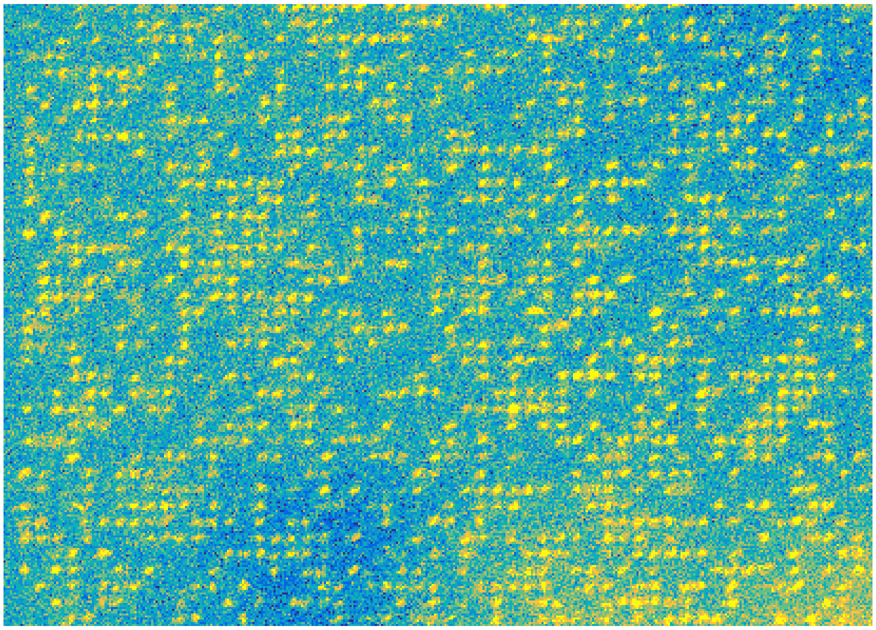


Figure 10. PRBA pattern extracted from outdoor testing using a 2.3 mW laser.

Note that this image was also taken in a challenging, 'worst case' configuration with the test target approximately normal to the sun angle. Given the results of this test using only a 2.3mW source (which is approximately 215 times less power than an upper limit class 3B laser), we believe outdoor active illumination is possible in the absorption bands listed within table 3 using a class 3B laser during peak daylight irradiance.

4 Software development

Creating software to analyze unique pattern elements is a trade-off between code complexity and handling edge-cases. A number of assumptions regarding the implementation of the system are coded into the software. Several of these are used to assist in parsing the image data. Others could be made to be configurable or adaptable but were not for the sake of efficiency. The term "dot" is used to indicate a feature in the illumination pattern. This is contrasted with the term "pixel," which is a point in the data image and is at a finer resolution.

Parameters such as target area size, location, filter sizes, dot size, and pattern dimensions are hard coded in the software. A preprocessing step that corrects for geometric distortions is also used to compensate for experiment to experiment variations in these parameters.

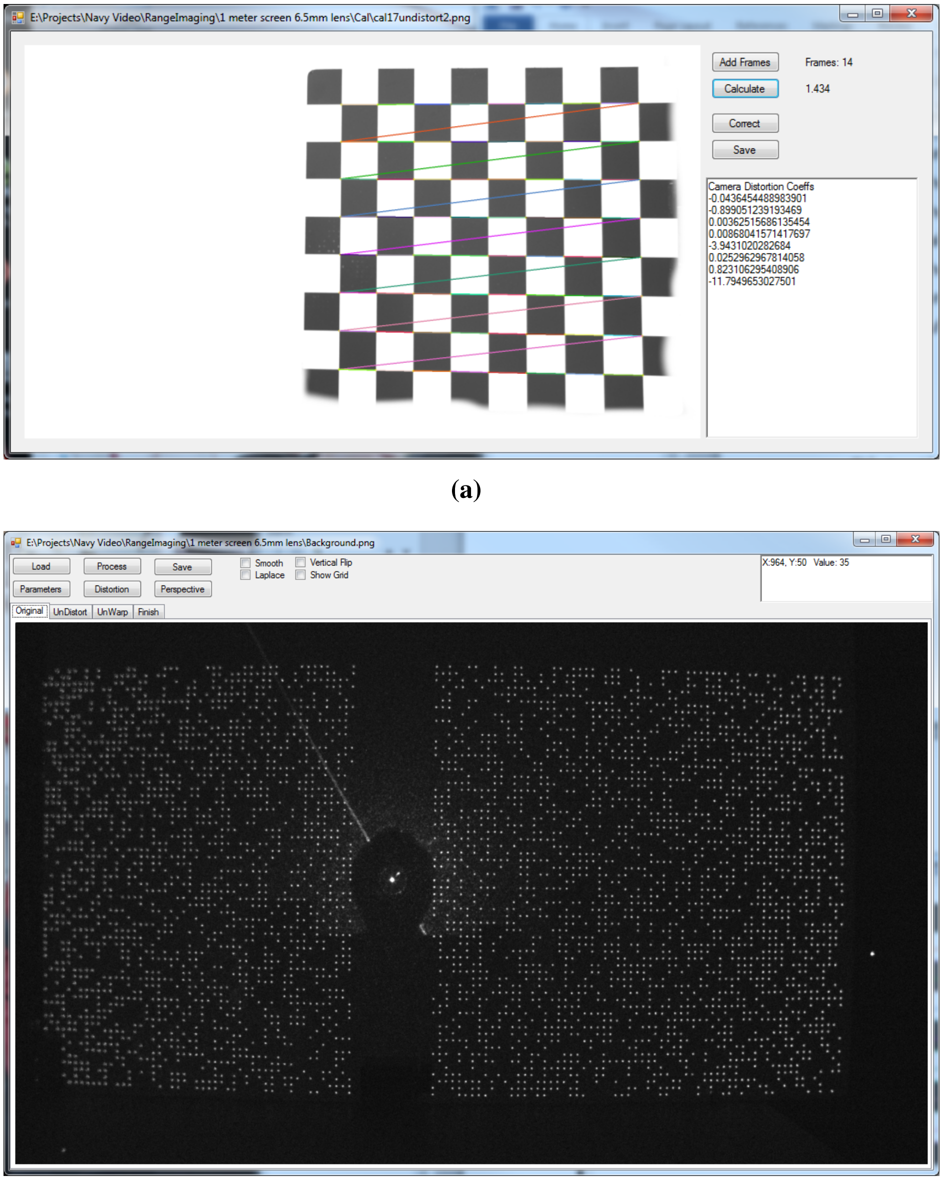
4.1 Physical configuration

Constraints for processing raw images of PRBA include:

- The illumination source is shifted horizontally to the side of the camera and centered vertically.
- The illumination source and camera are at approximately equal distances to the target.
- The illumination is a dot pattern with rows aligned to horizontal lines in the image.
- The illumination pattern rows, the line between the light source and the camera, and the camera sensor rows are all parallel and appropriately centered.
- The illumination is fixed relative to the camera. Pattern scale and offsets do not change.
- The image is sufficiently over-sampled. The example algorithm was developed with dots having a half-width of 2 to 3 pixels and a pattern period of approximately 15 pixels in the data image.
- The reference image has dots in a plane at the maximum distance: shifts due to variations in range are only in one direction.
- Sufficient contrast exists, background subtraction is not used.

4.2 Distortion

The first step in locating dots in the image is to determine the relationship between the data image and the illumination pattern. This includes correcting the geometric distortions. This system uses two steps to find the relationship. First, the program "Camera Calibration" takes a set of images of checkerboard patterns which is submitted to OpenCV's camera calibration routine.[3] A dozen images with the checkerboard filling most of the frame without cropping will provide a good start. However, the resulting transform is usually not sufficient. Additional refinement is necessary.



Frames: 14

1.434

```
Camera Distortion Coeffs
-0.0436454488983901
-0.899051239193469
0.00362515686135454
0.00868041571417697
-3.9431020282684
0.0252962967814058
0.823106295408906
-11.7949653027501
```

(a)

☐ Smooth ☐ Vertical Flip
☐ Laplace ☐ Show Grid

X:964, Y:50 Value: 35

(b)

Figure 11. (a) Checkerboard used for initial camera calibration, and (b), raw reference image.

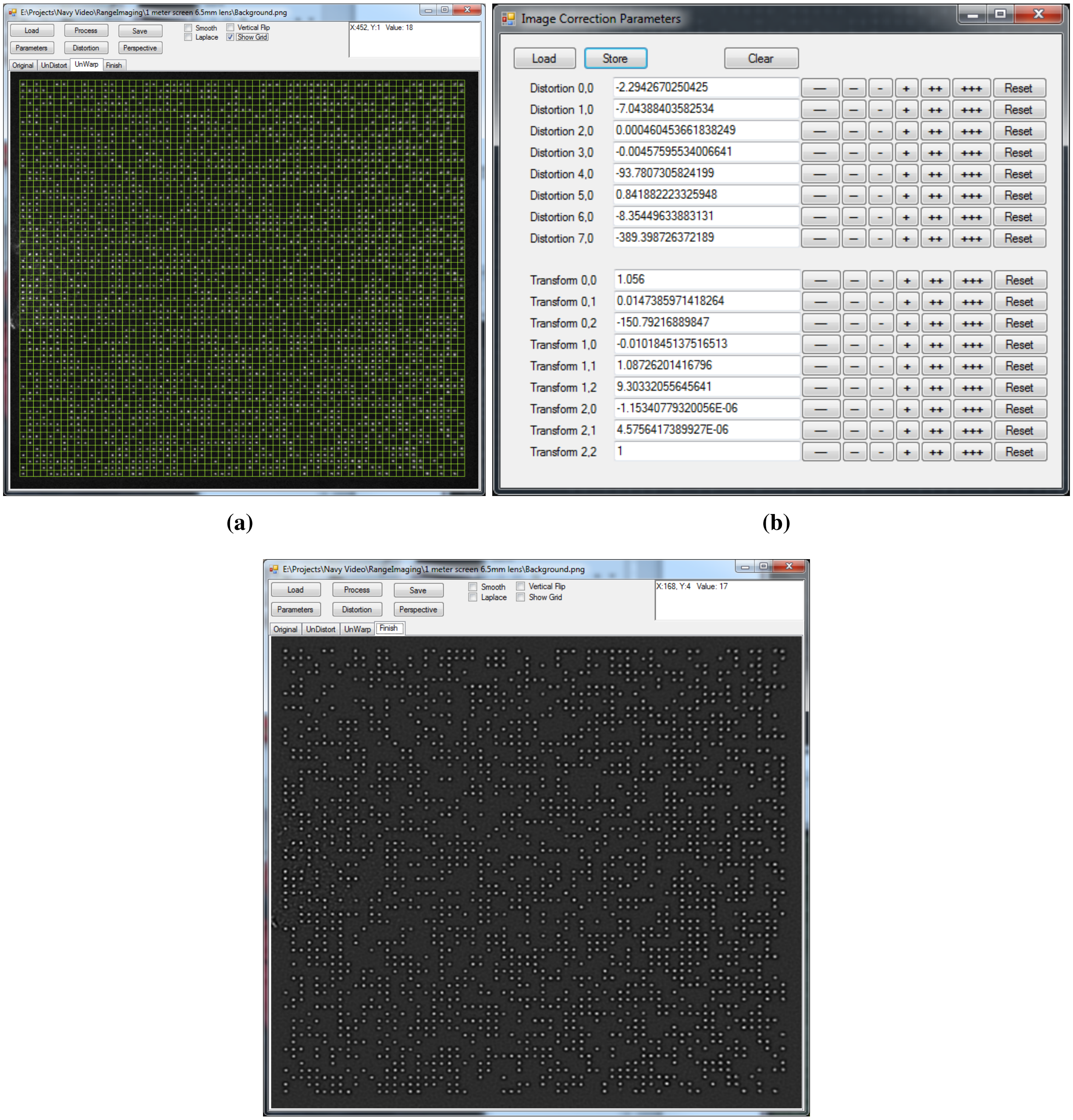
The next software program "PreProcessDoe" supports interactive adjustment of the coeffi-

---

[3] Note that OpenCV uses the internal corners of the checkerboard images. The perimeter may need to be erased to help locate the corners.

cients. An image of the reference pattern is displayed with a grid overlay and a dialog for editing distortion and transform coefficients. The values are initialized with the results of the OpenCV calculations and can be manually adjusted to improve the result. The final corrected image should have all dots roughly centered in the overlaid grid boxes. This and other initial setup operations are simpler if the reference template pattern has all four extreme corners in the "on" state. The resulting transform corrects camera distortion, the scene perspective, and matches the image data to the position and scale fixed in the subsequent processing steps. A small two-dimensional high-pass filter is also applied. The resulting images are cropped and stored for processing in the final analysis program.

Figure 12 show the original reference image, the geometric transform and crop with the grid overlay, and the final image used as reference data.



E:\Projects\Navy Video\RangeImaging\1 meter screen 6.5mm lens\Background.png

☐ Smooth
☐ Laplace
☐ Vertical Flip
☑ Show Grid

X:452, Y:1 Value: 18

(a)

Image Correction Parameters

Distortion 0,0	-2.2942670250425
Distortion 1,0	-7.04388403582534
Distortion 2,0	0.000460453661838249
Distortion 3,0	-0.00457595534006641
Distortion 4,0	-93.7807305824199
Distortion 5,0	0.841882223325948
Distortion 6,0	-8.35449633883131
Distortion 7,0	-389.398726372189
Transform 0,0	1.056
Transform 0,1	0.0147385971418264
Transform 0,2	-150.79216889847
Transform 1,0	-0.0101845137516513
Transform 1,1	1.08726201416796
Transform 1,2	9.30332055645641
Transform 2,0	-1.15340779320056E-06
Transform 2,1	4.5756417389927E-06
Transform 2,2	1

(b)

E:\Projects\Navy Video\RangeImaging\1 meter screen 6.5mm lens\Background.png

☐ Smooth
☐ Laplace
☐ Vertical Flip
☐ Show Grid

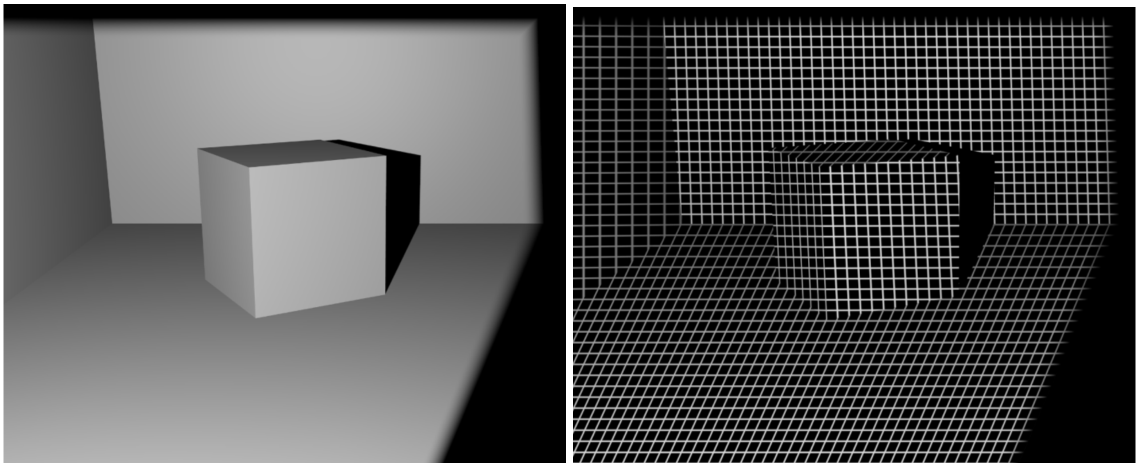
X:168, Y:4 Value: 17

(c)

Figure 12. (a) Corrected geometry image, (b), coefficient editor, and (c), reference data ready for processing.

4.3 Dot detection

To reduce processing load and false dot identification, the search for dots is guided by expected locations. It is easier to track the pattern when scanning horizontally. To illustrate, consider the scene shown in figure 13. Variations in range cause the dots to shift on the axis between the source and the camera (horizontal). On that axis, the dots will shift phase but will remain on the original line and generally remain in sequence. On the perpendicular (i.e., vertical) axis, the dots have a known vertical position but an unknown horizontal shift. This horizontal shift can span many dot periods and is easily aliased. A discontinuous pattern coupled with an unknown shift makes it more difficult to track data vertically than horizontally.



(a) **(b)**

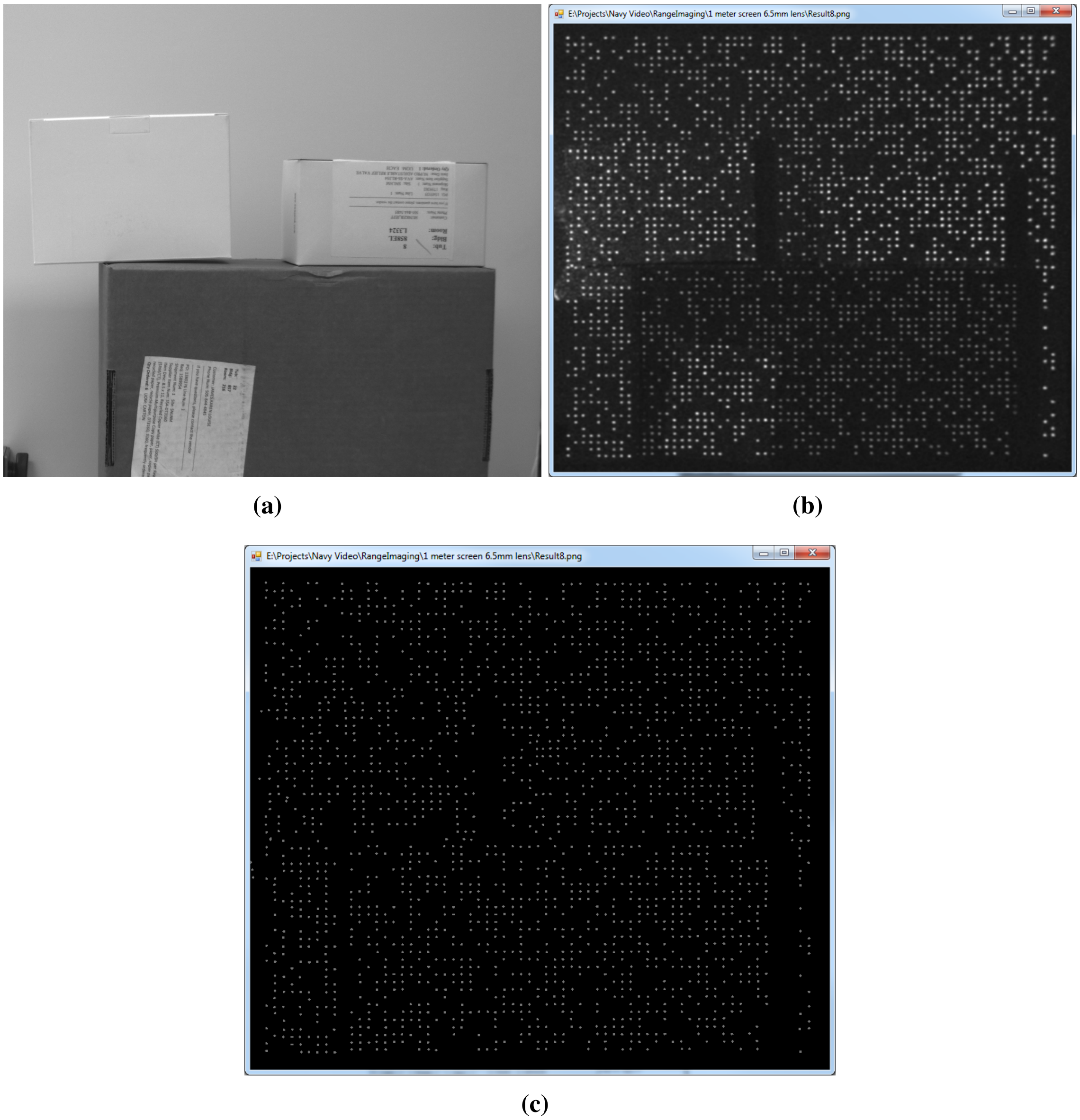
Figure 13. (a) Simulated projector/camera system showing a cube, and (b), simulated projector/camera system showing a cube with a projected pattern of orthogonal lines incident upon the cube.

Dot locations are identified by searching the horizontal line where the dots are expected. The search includes the region one-quarter vertical dot spacing above and below the expected location. As the search point progresses along the horizontal line, the image intensity is tracked. If the intensity has been increasing and then a significant drop occurs ($>$ 5 gray levels) the local region is searched for a peak. That peak is compared to a background sample (1/2 line vertical shift). If the peak exceeds the background by %150 + 5 gray levels, it is assumed to be an illuminated dot. Peak detection should be dynamically optimized, but this method proved sufficient for indoor tests.

After a row of "on" dots is identified, blank regions are back-filled with "off" dots at the nominal frequency. This is a potential source of error and could be improved, preferably by avoiding having to interpolate where "off" pixels are.

The detection process results in a list of dots for each row. The list is in order by horizontal position. Each entry includes the “on” dot’s location or the expected “off” dot’s estimated location. The rows of sensor data are now ready to be compared to the reference data.

The reference data is specified by interpreting an image of a constant range background. The pattern is defined by doing localized searches at deterministic locations. The locations of the found dots are used instead of the calculated locations to help reduce the influence of distortions or other unknowns. Images demonstrating this process are shown in figure 14.



(a) **(b)**

(c)

Figure 14. (a) Test scene under natural illumination, (b) pre-processed data image, and (c), dots found by the algorithm. Note the reflectivity differences in the objects within the scene, and the corresponding found dot patterns.

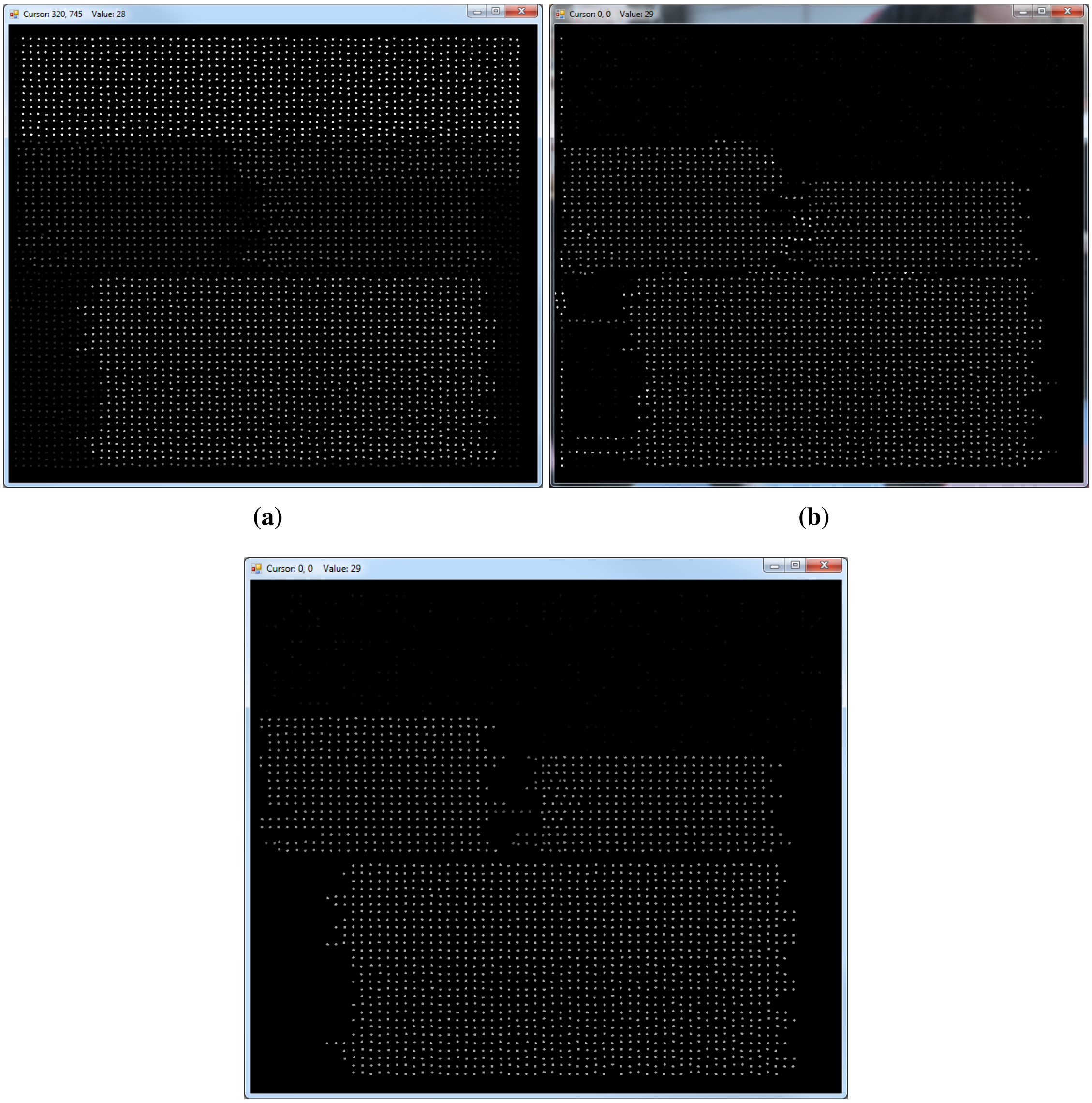
4.4 Pattern correlation

With a fixed physical configuration, it would be unusual for there to be any vertical dislocation of the dot pattern. Therefore, the rows can be assumed to match up and only need to be correlated with the expected reference data row.

A data line is correlated with a reference line by comparing the logical state of each dot to the reference information. Matching regions are identified and all locations within the match are assigned a value based on the length of the matching group. This process is repeated for each possible shift. Then, the set of shift/match/accumulate results is searched for the greatest match at each point. The amount of shift in the data containing that maximum indicates the number of dot locations each dot is shifted in the data image. This creates an association between data image dots and reference dots.

Each association includes the template location, strength of the association (i.e., length of matching sequence), and the actual pixel offset for range calculation. Errors are indicated when a template dot has no associated data image dot or multiple data image dots claim the same template dot. Figure 15 shows an example of this process.

The current software provides an image pixel offset between the data dots and the reference dots. The viewer scales this value and uses it for position and/or color to assist in interpretation. It does not apply physical scene parameters to generate world coordinates. All measurements are in image pixel space.



(a) **(b)**

(c)

Figure 15. (a) template match strengths for a simple test scenario, (b) dot shift values for the corresponding scene, and (c), dot shift with confidence threshold.

4.5 Error reduction and potential improvements

There are many potential sources for error in the reconstruction. Most of them involve missing or occasionally extra dots in the measured data. Fortunately, there is additional information available to detect and attempt to correct some of the errors after the initial processing. Enforcing known constraints on the data should produce results with a better match to actual scene. These are generally “healing” type operations that will provide the greatest benefit on point defects and boundaries. They are potentials and have not been implemented at this time.

Several situations could cause more than one data dot to be assigned to a single template dot. If an illumination dot is incident on an edge in the scene, two dots could be found in the data image. If the found dots are adjacent and each fit with their local (one sided) pattern, they could both be considered valid data. Otherwise, only one can be a true match. In that case, the strength of the signal, the size of the match, and the location of the surrounding matches should be used to select the best matching template dot.

The intent of the pseudo random tiled illumination pattern is to provide unique regions to match in the image. In regions where the horizontal pattern match length is short (weak match), comparing data across rows could help reduce ambiguity.

This configuration of the equipment does not normally result in vertical displacement of template dots. Pattern uniqueness is not required in the vertical direction. This allows for a smaller number of unique tiles. A smaller number of unique tiles means that the tiles can be smaller and offer finer matching region resolution. While the tiles only need to be unique in the horizontal direction, they should be unique to single dot shifts, not just full tile offsets.

“Off” dot locations are not measured, they are estimated from surrounding information. They do not provide as much information as features found in the image. The number of “off” dots should be minimized in the template. Also, long runs of any single state reduce the effectiveness of correlation measurements at small scales.

It may be useful to retain the top few correlation matches at each dot in order to use them as initial candidates for alternate correlation solutions. Extending the comparison in the vertical direction offers additional points to match, but there is potential confusion due to possible changes in range along the vertical axis.

4.6 Future direction: Differential template pattern

A clean data set makes the image reconstruction fairly straight forward. It is the dot identification errors that make the recognition challenging. One of the easiest ways to improve results would be to improve the dot detection. Using differential signals can significantly improve detection in a noisy background. Differential signals would also avoid the need to estimate the locations of the “off” pixels. These two benefits would significantly improve reconstruction fidelity.

One way to achieve a differential signal would be to reverse the contrast in the illuminator and take a second image. A difference of the two images produces a stronger signal at all template locations and significantly reduces the background noise. An important weakness of this approach is that it is sensitive to motion in the image. If the template pattern is dynamic, the reconstruction frame rate will be halved as each reconstruction requires two frames of data. A static template does not reduce the frame rate as any sequential pair of images provides a differential set.

Another approach to a differential template would be to increase the resolution of the illuminator and use dot pairs to represent each data point. In this context, a vertical pair would be appropriate as few scene conditions affect vertical scale. Whether the upper or lower of the pair is “on” determines the state of the dot. This approach places additional demands on the resolution of the illuminator and camera, but does not increase the system sensitivity to motion in the scene. A state and position for each logical dot in the template can be recovered in a single frame.

5 Optimized projected patterns and homographies

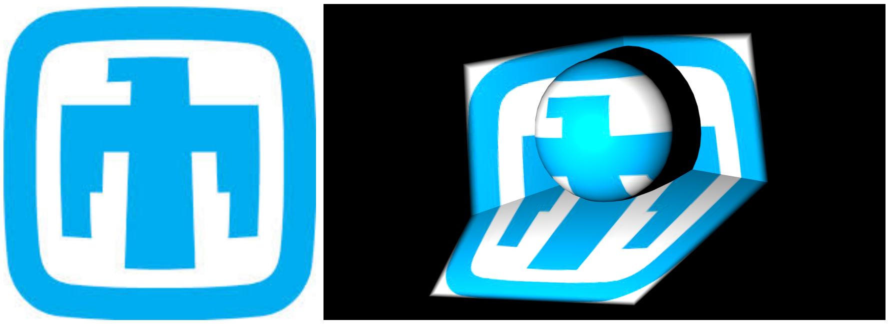
Though PRBA DOEs appear to have great promise, they only project a single fixed pattern. Work was performed to understand potential methods to extract maximum information from a scene given the ability to alter the projected pattern. An immediate problem that appears in this scenario is the variability of scene complexity and the potential for losing significant regions of data due to occlusions or shadows.

When images are projected onto a geometrically complex scene, varying depth and discontinuities typically cause the projection the appear skewed. These distortions can be corrected through a camera-projector system, in which a correspondence is established between the projector and camera such that images can be pre-skewed to account for these geometric variables a priori. When projected onto the scene, these pre-skewed images appear rectilinear to the camera, and thus can enable maximum information extraction.

5.1 Problem Description

Perspective distortion is a phenomenon in which an object or image appears different when viewed from different perspectives. In regards to a projector, images appear rectilinear only from the point of view of the projector itself. Even on planar surfaces, such as a projection screen, an image will be slightly skewed depending on the angle and position of the viewer. The results of such effects can be seen in Figure 16. The original image appears in the proper size, shape, and orientation only when viewed from directly behind.

When projected onto a nonplanar scene, objects, such as the sphere, can cause discontinuities in the perceived image. Areas behind these object in which there is no light from the projector are called shadows. These cannot be rectified without the use of a second projector. Conversely, areas which are illuminated by the projector but not captured by the camera are called occlusions. The discontinuous nature of these objects can cause perspective correction around the boundaries of these areas to be much more difficult to correct.



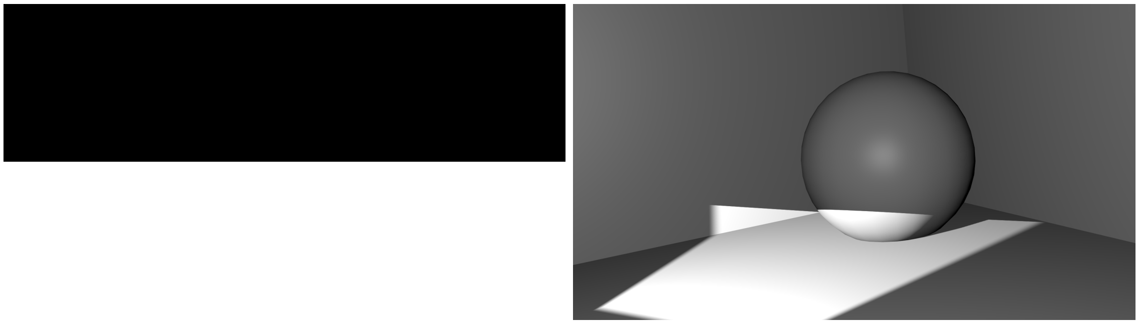
(a) Original Image **(b)** Image with perspective distorition

Figure 16. Ordinary projector system.

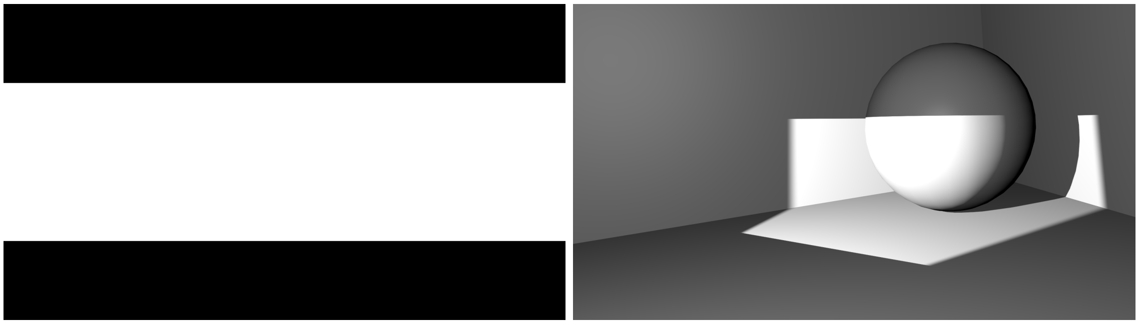
5.2 Development of corresponding points

To project an image onto a complex scene that appears rectilinear to the camera, the correspondence between the projector and camera must first be established. This unique correspondence can be used to pre-warp the image such that the geometric distortions are corrected a priori. The correspondence problem can be solved through a variety of methods.

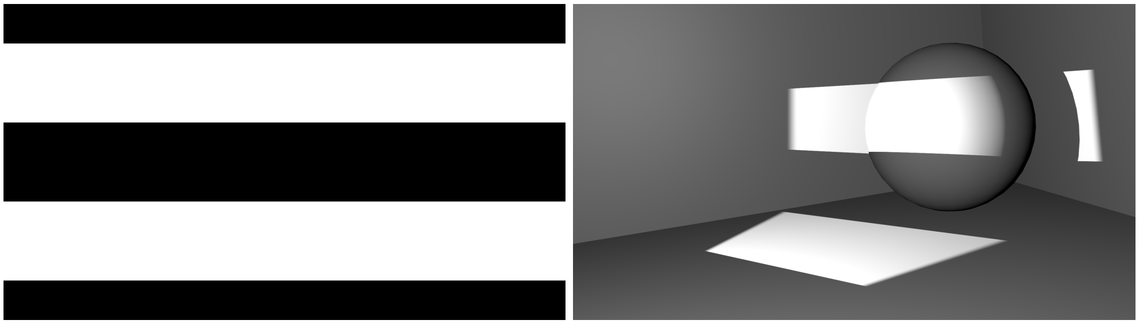
For this system, a series of gray-coded images were used to uniquely code each pixel. This method is effective for detecting encoding errors since neighboring pixels differ only by a single bit. Additionally, the edges between dark and white in a previous image will become the center of either a dark or white stripe in the next image. This ensures that areas which are prone to errors in the previous image are moved to the area with the highest contrast in the next. This method minimizes the effect of noise on the image, which is critical for implementing a real-world system. A series of gray code images can be seen in Figure 17.



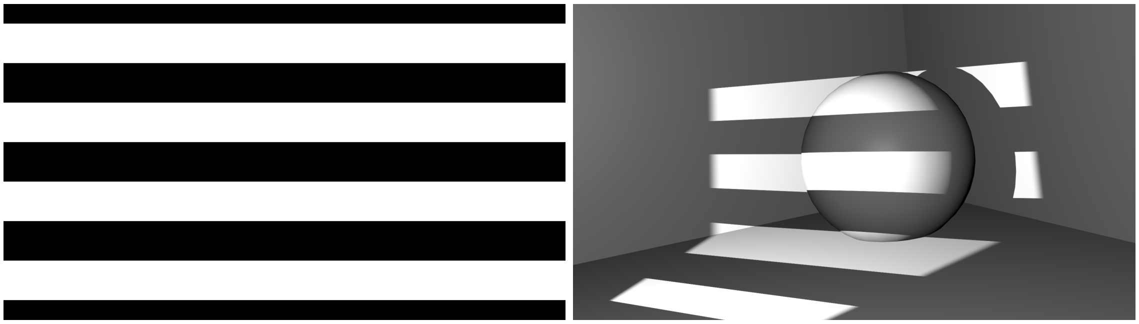
(a) Projected Gray Code Layer 0 **(b)** Captured Grey Code Layer 0



(c) Projected Gray Code Layer 1 **(d)** Captured Grey Code Layer 1



(e) Projected Gray Code Layer 2 **(f)** Captured Grey Code Layer 2



(g) Projected Gray Code Layer 3 **(h)** Captured Grey Code Layer 3

Figure 17. Several layers of gray codes and the resulting projected images.

To account for varying albedo and nonuniform lighting, each gray code image was projected, inverted, then projected again. The images were then compared against each other to search for the regions which became brighter due to the white strips. The use differential lighting to determine light and dark coding regions within the gray code, while taking twice as long, can account for a large variety of environmental conditions. Even with this technique of double projection, it is still possible for pixels which exist in areas with low SNR, such as the edge of shadows or object with poor reflectance, to be improperly encoded. An example of such regions can be seen in the fringes of the shadow region in Figure 18.

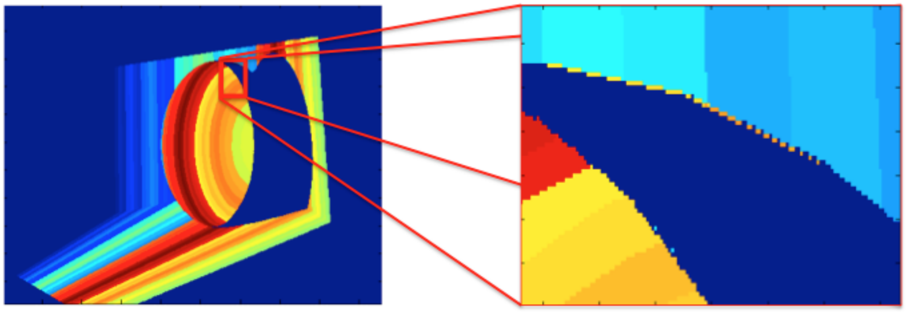
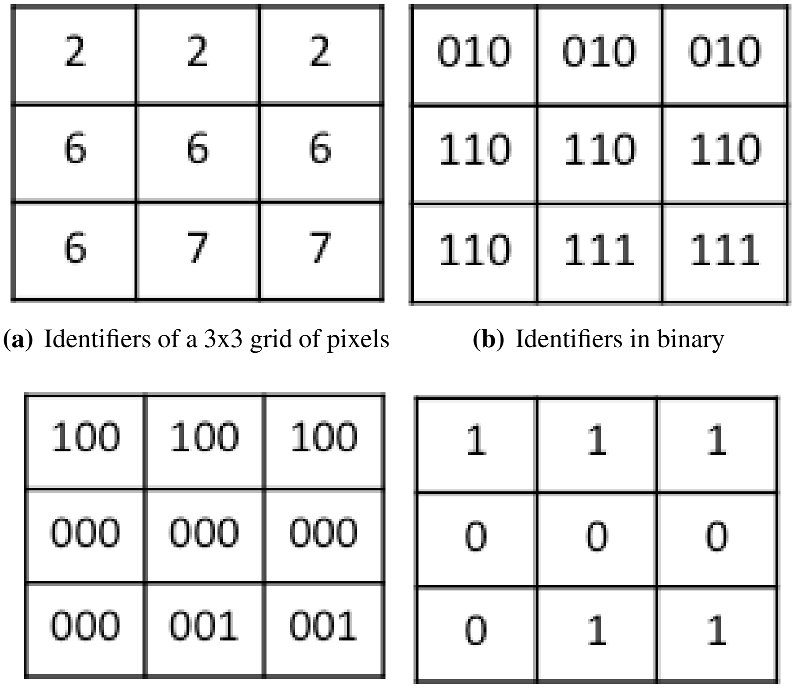


Figure 18. Certain pixel IDs are not accurately recorded on some edges.

5.3 Error Detecton

Since gray-coded pixels differ from their neighbors by only a single bit, it is possible to detect errors through an XOR filter. An XOR filter will output high bits if the corresponding input bits differ from each other. Since each neighboring pixel should only vary from its neighbor by a single pixel, if there is more than 1 high bit in the n-bit output of filter, the pixel is determined to have an encoding error and replaced by a null identifier. This excludes the pixel from future calculations.

An example of XOR filtering can be seen in Figure 19. In this example, the center pixel is compared to its neighbors. Since the middle pixel only differs by 1 bit, from its surrounding neighbors, it was correctly encoded.



2	2	2
6	6	6
6	7	7

(a) Identifiers of a 3x3 grid of pixels

010	010	010
110	110	110
110	111	111

(b) Identifiers in binary

100	100	100
000	000	000
000	001	001

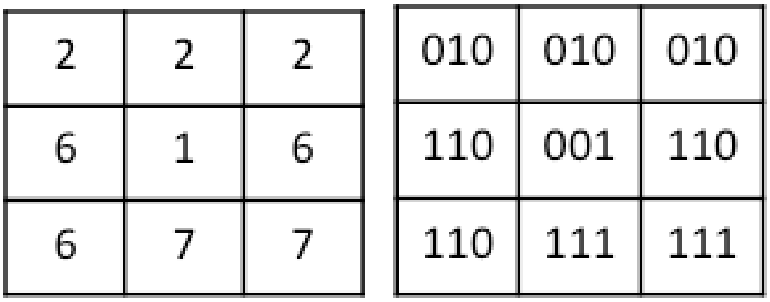
(c) Output of XOR filter

1	1	1
0	0	0
0	1	1

(d) Number of bits different from center

Figure 19. XOR filter process with no errors.

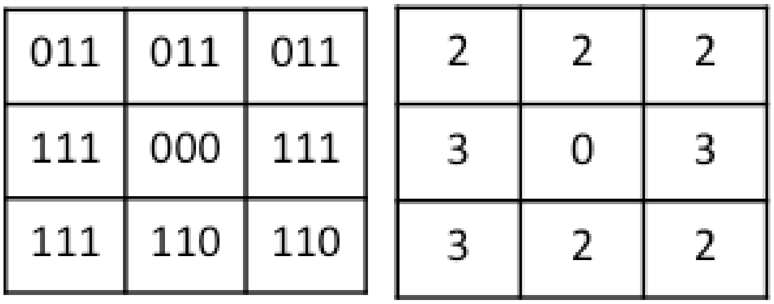
In Figure 20, the XOR filter reveals that the center pixel differs from its neighbors by 3 bits. This means that the pixel's identifier recorded by the camera does not match the identifier attempted to be given by projector. This pixel is then excluded from the interpolation process.



2	2	2
6	1	6
6	7	7

010	010	010
110	001	110
110	111	111

(a) Identifiers of a 3x3 grid of pixels **(b)** Identifiers in binary



011	011	011
111	000	111
111	110	110

2	2	2
3	0	3
3	2	2

(c) Output of XOR filter **(d)** Number of bits different from center

Figure 20. XOR filter process with errors.

It should be noted that other filters other than the XOR were considered, and found to have undesirable results. A mean filter assumes the data is related by numerical proximity, which is not the case for gray code, where it is possible to jump from ID ”10001” to ”00001” in neighboring spaces. A median filter was experimentally shown to degrade image quality. It is best to use a filter that does not alter the original observed IDs and instead detect and simply reject error-prone regions.

5.4 Interpolation

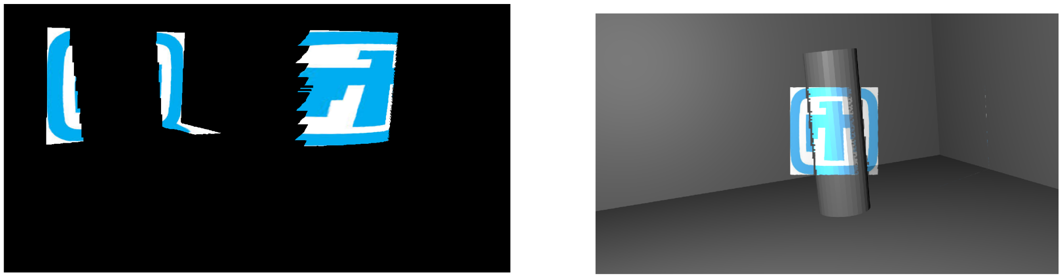
Since a unique one-to-one mapping between the camera and projector is usually not possible, missing or invalid data points must be recovered through interpolation. Two types of interpolation were primarily tested: piecewise homographies and bilinear interpolation.

Piecewise Homographies

Homographies are traditionally used in image processing to relate pixel position between two images. This general geometric transform assumes the scene is planar, which allows for continuous linear transformation of the pixels to exist between images. Since the approximation of planarity is poor in many cases, it is instead possible to develop a piecewise relationship between the camera and projector by dividing the scene into sub-windows, for which this approximation is accurate. A homography can be established for each of these windows, which then allows for projective correction of the pixels within this region. This piecewise network of homographies allows the scene to be approximated as a series of continuous, interconnected planes.

This approach allows most of the original information to be used from the image, (i.e., the highest quality pre-skewed image are derived from this method). Additionally, a lack of resolution in determining spatial correlation will not degrade image sharpness; only its rectilinearity will be affected. However, this method is computationally expensive. For N gray code images projected, it requires N^2 homographies to be calculated. Additionally, the placement of the sub-windows becomes crucial to post-rectification image integrity. Any corner points that exist within shadow regions prevent the calculation of the homography for the surrounding four sub-windows. Also, any points which lie across discontinuous edges suffer from heavy over-skewing. This approach, therefore, suffers when a large number of discontinuities are present.

An example of piecewise homography interpolation can be seen in Figure 21. The quality of the original image is maintained, and sharp lines between colors is apparent within the image. Additionally, the image is perceived as rectilinear. However, the areas between the pillar and the wall show a darkened line where the scene is no longer continuous.



(a) Pre-skewed image: projector perspective **(b)** Pre-skewed image: camera perspective

Figure 21. Pre-skewed images using piecewise homography interpolation.

Bilinear Interpolation

Bilinear interpolation is a method in which every piece of data is connected to its neighbors by a plane. This allows for the linear connection between all data points. For this case, the pixels which were successfully mapped translate their values to the projected image. From there, the values of the invalid and missing pixels are then able to be obtained.

Bilinear interpolation is fairly robust against discontinuities; instead of relying on rectilinear windows, the bilinear interpolation method uses all pixels which were successfully correlated as a point of reference. This means that sections with errors which were not filtered in the previous stage can potentially localize the effects of the error.

This method is much faster computationally, requiring only one more image in the gray code to be projected to double the resolution of the pre-skewed image. Additionally, this method generally ignores discontinuities. The use of many points around the edges of objects and shadows allow for the smooth transition of the image between discontinuous features. However, this method suffers in image quality. Because it relies on only known data points, if there is a sparsity of information the spaces in between these points get blurred during the interpolation process. This means that the quality of the image is degraded, but rectilinearity is preserved.

An example of bilinear interpolation can be seen in Figure 22. The edges between the sphere and wall and shadow are both smoothly handled, with no obvious disparities between either. However, the interpolation causes ringing and blurring around the edges of the image lines. This is due to the sparsity of information from the original image. Even so, this image is still of sufficient quality, and likewise has the desired rectilinearity.

(a) Pre-skewed image: projector perspective

(b) Pre-skewed image: camera perspective

Figure 22. Pre-skewed images using bilinear interpolation.

5.5 Optimized projected pattern summary

Images were pre-skewed using a camera-projector system such that they appear rectilinear when projected onto a geometrically complex surface. This is accomplished by solving the correspondence problem between the projector and camera, then interpolating between missing data points using a piecewise homography network or bilinear interpolation. This allows an image to be arbitrarily placed on the camera's virtual scene, then translated to the projected image though the corresponding pixels such that, when projected, the camera's perspective of the scene will contain the image where it was placed only virtually before. Pre-skewed image can thus be projected onto a scene appearing rectilinear to the camera in a desired position and orientation.

6 Summary and future investigation

This project investigated the 3D imaging method, illumination design, and software required to create a 3D imaging system capable of real time operation in an outdoor environment typically seen in various physical security scenarios.

Pseudo-random binary array patterns were projected onto complex scenes. Leveraging the mathematical properties of PRBAs and the physical construction of the 3D imaging system, depth points could be accurately extracted from a single image after a system calibration was performed.

The use of narrow band NIR illumination appears a practical solution capable of achieving the SNR needed for pattern extraction in an outdoor environment. Specifically, the 920-990 nanometer spectral absorption region optimizes minimized solar irradiance, low absorption over several hundred meters, and remains within the spectral sensitivity region of silicon detectors

Finally, software was developed to extract depth information from the 3D imaging system, and two methods were developed to construct optimized patterns for projection given a complex scene.

Future investigations and opportunities include investigation into multi-spectral projecting techniques using a DOE, differential dot pair projected patterns without any “off” projected elements, and illumination design capable of rapidly altering the projected pattern.

References

[1] N. Trif, “Model–Based Visual Recognition of 3-D Objects Using Pseudo-Random Grid Encoding,” Ottawa-Carleton Institute for Electrical Engineering, 1993.

[2] L. Kari and Z. Xu, “De Bruijn Sequences Revisited,” in Automata and Formal Languages, 2011, pp. 241–254.

[3] J. Salvi, S. Fernandez, T. Pribanic, and X. Llado, “A state of the art in structured light patterns for surface profilometry,” Pattern Recognit., vol. 43, no. 8, pp. 2666–2680, Aug. 2010.

[4] F. J. MacWilliams and N. J. A. Sloane, “Pseudo-Random Sequences and Arrays,” in Proceedings of the IEEE, 1976, vol. 64, no. 12, pp. 1715–1728.

A Software implementation information

The reconstruction process is implemented in the C# language on a desktop workstation running Microsoft Windows 7. It is not tied to Microsoft products or particularly compute intensive and could be implemented on many platforms. This environment was selected simply due to current familiarity.

A.1 Computing platform

The following platform was available and more than adequate for the task. All work was done off-line.

- Hardware: Dell T5610 workstation
- OS: Microsoft Windows 7/64
- Environment: Microsoft Development Studio / C#
    - A general purpose programming environment, MS Dev Studio provides a comprehensive development environment for C#, a memory-managed language which is well suited to algorithm development and implementation

A.2 Software packages

This implementation utilized several freely available packages that are commonly used to perform fundamental operations in an image processing context.

- OpenCV
    - Image processing library, used to estimate lens distortion and perform distortion correction
- Emgu.CV
    - C# wrapper for OpenCV, provides access to OpenCV in a managed memory programming environment
- OpenGL
    - 3D rendering library, used to visualize the reconstructed image

- OpenTK
    - C# wrapper for OpenGL, provides access to OpenGL in a managed memory programming environment
- Gimp
    - 2D image manipulation program, used to prepare and evaluate simulation images, lab data, and results
- Blender
    - 3D creation program, used for creating simulations and preparing test data

A.3 Code base

The implementation specific software was developed in house and exists in a Microsoft Development Studio solution (RangerReader.sln) which contains three projects:

- Camera Calibration
    - Uses multiple images of a checkerboard to estimate lens distortion coefficients
- PreProcessDoe
    - Performs user interactive lens distortion correction, perspective correction, cropping, and some filtering operations to prepare images for parsing
- RangeReaderDoe
    - Parses image data and compares to a reference template. Results are displayed in both 2D and 3D representations.

B Interfacing Blender with Matlab

To automate image simulations, Matlab was configured to interface with Blender. Python was used to interface the two programs, since Blender communicates directly with Python. The following tutorial was tested on a Mac. Version 2.72 of Blender was used, which has installed python version 3.4.0.

Although specific instruction will vary with the host system's specifications, the general steps to interface Blender and Matlab should remain the same:

1. Configure Blender to run through terminal commands
2. Configure Matlab environmental variables
3. Create scene in Blender
4. Write Python code
5. Write Matlab wrapper code

B.1 Configure Terminal Commands

It is possible to use the "alias" command to run Blender from the command line for a Mac. This allows the user to specify unique commands for the terminal. This section allows the user to call Blender from the terminal.

1. Open a terminal
2. Type the following command into the terminal. This creates the command "blender" and associates it with the executable that opens blender. Edit the path to fit where the program is located on the host system.

   ```
   sudo echo “alias blender=/Applications/Blender/blender.app/Contents/MacOS/blender” >> 7.bash_profile
   ```

3. Close the terminal, then open a new terminal.

   The terminal must be restarted to put the changes into effect.

4. Type "blender" without quotation marks into the terminal.

   If done correctly, this should open a new instance of Blender.

B.2 Configure Matlab Environment

Matlab is able to execute system commands within the terminal by placing an *!* before any command. To run the “blender” command, the Blender environmental variables must be set to the correct paths. If the previous step was implemented correctly, this section should allow the user to open a new instance Blender by typing *!blender* into the command line.

Type the following into the terminal:

```
blenderpath = ':/Applications/Blender/blender.app/Contents/MacOS';
pathstr = getenv('PATH');
if isempty(findstr(pathstr,blenderpath))
setenv('PATH', [getenv('PATH') blenderpath]);
setenv('BLENDER_SYSTEM_SCRIPTS',...
'/Applications/Blender/blender.app/Contents/Resources/2.72/scripts');
setenv('BLENDER_SYSTEM_DATAFILES',...
'/Applications/Blender/blender.app/Contents/Resources/2.72/datafiles');
setenv('BLENDER_SYSTEM_PYTHON',...
'/Applications/Blender/blender.app/Contents/Resources/2.72/python');
end
```

After setting the environmental variables, run the command, *!blender*. The command should open an instance of blender without any errors.

B.3 Create a scene in Blender

This section allows the user to create a scene in blender which is capable of interfacing with Matlab. The instructions include how to create a projector in blender. Images from Matlab can be automatically loaded onto the projector, and the rendered scene can be saved automatically as well.

1. Open Blender and create a new scene and save it as ProjImScene.

   Note, the name can be anything but must be referred to later in the Python and Matlab wrapper code.

2. Model a scene with some simple objects, such as a plane or sphere and save the scene.

   Since rendering can take significant time, it is recommended that materials and textures are not added at this time.

3. Create a spot light source and name it Projector.

4. Under Object Data, open the tab Spot Shape and check the box Square.

   This creates a square light source. Additionally, the size of the projector can be modified in this tab.

5. Under Texture, click New to create a new texture. Name the texture ProjIm.
6. Ensure that type is set to Image or Movie.
7. Open the tab Image and load the image to be projected.
8. Open the tab Mapping and set Coordinates to View.

The scene should be ready to interface with Matlab. It is important to have a unique name for the scene and the texture being projected.

B.4 Python Code

Python can be used to directly modify the Blender environment. Blender also automatically generates a line of python code for every action completed in the GUI, so adding lines of code is relatively simple. To access these auto-generated lines, tutorials are available online.

The code below loads an image to the projector in Blender, renders the scene, then saves the image. These images will be loaded into Matlab later in the wrapper code. Save this file as *Proj_autorender.py*.

```
import bpy
import os

cdir = os.getcwd();

bpy.data.textures["Projim"].image = bpy.data.images["ProjectedImage.png"]
bpy.data.scenes['Scene'].render.filepath = 'RenderedImage.png'
bpy.ops.render.render( write_still=True )
```

B.5 Matlab Code

To pass the images from code to code, images are saved in the local file by one language, then loaded by the second. The function below takes an image *im* and saves it to a local file. It then calls blender and runs the python code from the previous section. It then loads and returns the image rendered by Blender, *renderedim*. Save this function as *renderBlender.m*.

```
function renderedim = renderBlender(im)

imwrite(im,'ProjectedImage.png');
!blender -b ProjImScene.blend -P Proj_autorender.py > dump.txt
renderedim = imread('RenderedImage.png');

end
```

If all the previous steps were successfully completed, it should be possible to simply input an image into the function *renderBlender* in Matlab, have Blender project the image, and return the rendered image of the scene.

DISTRIBUTION

Quantity	Mail Stop	Name	Organization
1	MS0781	Gabriel Birch	6525
1 (electronic copy)	MS0899	Technical Library	9536

Sandia National Laboratories