

# Robust Pixel Classification for 3D Modeling with Structured Light

Yi Xu

Daniel G. Aliaga

Department of Computer Science  
Purdue University  
{xu43|aliaga}@cs.purdue.edu

## ABSTRACT

Modeling 3D objects and scenes is an important part of computer graphics. One approach to modeling is projecting binary patterns onto the scene in order to obtain correspondences and reconstruct a densely sampled 3D model. In such structured light systems, determining whether a pixel is directly illuminated by the projector is essential to decoding the patterns. In this paper, we introduce a robust, efficient, and easy to implement pixel classification algorithm for this purpose. Our method correctly establishes the lower and upper bounds of the possible intensity values of an illuminated pixel and of a non-illuminated pixel. Based on the two intervals, our method classifies a pixel by determining whether its intensity is within one interval and not in the other. Experiments show that our method improves both the quantity of decoded pixels and the quality of the final reconstruction producing a dense set of 3D points, *inclusively* for complex scenes with indirect lighting effects. Furthermore, our method does not require newly designed patterns; therefore, it can be easily applied to previously captured data.

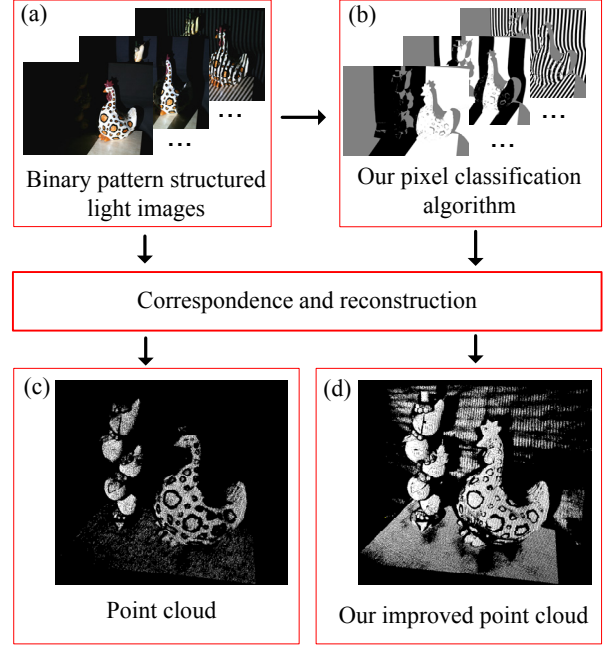
**CR Categories:** I.3 [Computer Graphics], I.3.7 [Three-Dimensional Graphics and Realism], I.4 [Image Processing and Computer Vision], I.4.1 [Digitization and Image Capture].

**Keywords:** structured light, direct and global separation, 3D reconstruction.

## 1 INTRODUCTION

Modeling real-world scenes plays an important role in computer graphics, virtual reality, *historical site* preservation, and other commercial applications. One option is to use structured light systems that make use of lighting devices (e.g., digital projector) to encode scene points by projecting illumination patterns and taking images of the scene under each pattern. Pixels with the same codeword are corresponded and *triangulated* to obtain dense 3D scene samples allowing for point-based modeling, rendering, and other graphics applications [2]. One commonly used illumination pattern is the binary pattern, which is an image of *interleaving* black and white stripes. Each illumination pattern sets one bit of the codeword of a pixel according to whether the corresponding scene point is directly illuminated or not.

Determining whether a pixel is on or off under an illumination pattern is an important and challenging problem in structured light systems. Ideally, applying a threshold to a captured image yields a simple binary image for pixel classification. However in practice, the unknown and potentially complex surface and illumination properties of the scene make such a simple method prone to many classification errors. This leads to incorrect correspondences and thus either to a bad reconstruction or to many samples being lost. To achieve more accurate classification and greater number of



**Figure 1. System Pipeline.** a) We show the binary pattern structured light images. b) Pixels are classified using our algorithm. White/black/gray means on/off/uncertain. c) Points are reconstructed using a standard classification method. d) In comparison, our algorithm robustly produces more points: 201528 points vs. 57464 points.

samples, previous methods attempt to adaptively compute a pixel threshold value, to project pattern images and their inverses, to use several camera exposure times, to project multiple patterns with different intensities, or to use post-processing to clean-up classification (e.g., [5][6][8]).

Most of these methods assume that a scene point is brighter when it is illuminated. However, this assumption only holds when a scene point has a weak indirect light component. Consider the following counter-intuitive examples. (1) If a scene point is in shadow, it should have zero intensity under any illumination pattern. However, due to inter-reflection from other surface patches, the point might have large illumination intensity and thus be classified incorrectly. (2) A scene point may appear dark despite being directly illuminated if the part of the scene from which it would normally receive a significant amount of indirect light is currently not lit. Yet, projecting a slightly different pattern might illuminate the source of the indirect light and make the same point appear very bright even if it is now not directly illuminated. While the total direct and total indirect illumination components of a *fully* lit scene can be separated without knowledge of scene geometry (e.g., [3]), the indirect illumination component for arbitrary projected patterns depends both on the pattern and on the scene geometry. This produces the chicken-

and-egg problem of needing to know scene geometry before identifying scene illumination properties and needing to know scene illumination properties in order to perform robust structured-light scene acquisition.

Our key observation is that for each pixel, we can estimate tight intensity value bounds for when the pixel is on and for when it is off. Knowing these intervals, we can accurately classify a pixel when its intensity value falls into one interval but not in the other.

Our method bounds the two intervals of a pixel based on the following facts: a pixel is on only if its intensity includes the direct component; and, the indirect component of a pixel under a black-and-white stripe pattern is smaller than the total indirect component of the same pixel. This last assumption is easily true since only some of the projector’s pixels are on during any of the binary patterns. Our algorithm produces a significantly more robust and accurate classification.

In this paper, we present a pixel classification algorithm for structured light systems using binary patterns (Figure 1). First, the higher frequency binary patterns are used to separate the direct and indirect components of each pixel, as well as to encode the pixel codeword. Then, our algorithm computes each pixel’s illuminated interval and the non-illuminated interval. A pixel is classified according to the two intervals. Pixels that cannot be classified are labeled as uncertain (Figure 1b). The classification results are fed to a reconstruction engine to demonstrate the quality of the classification (Figure 1d). We have captured and applied our algorithm to several real-world scenes. Moreover, since our method only uses the same binary patterns as in a standard structured light system, it can be easily applied to previously captured data. Results show that, as compared to naïve methods, our algorithm has two significant benefits: it increases the total number of decoded pixels and it improves the quality of the reconstruction. On average, our algorithm reconstructs 2-7 times more points for a variety of objects and surface types.

Our major contributions are

- a method to determine the intensity intervals of a pixel when it is illuminated or non-illuminated during structured-light acquisition, and
- a pixel classification algorithm which allows structured light systems to work better in complex scenes undergoing significant indirect lighting effects.

## 2 RELATED WORK

Our research improves structured light systems using binary patterns with insight provided by direct and indirect illumination component separation. As related work, we present a summary of work in coded structured light reconstruction with focus on binary patterns and in separating direct and global illumination.

### 2.1 Coded Structured Light Systems

Coded structured light systems project illumination patterns onto the scene in order to obtain a corresponded set of pixels. The correspondence may be performed between the projector and a camera or between two or more cameras. The coding strategies used by such systems can be classified as temporal coding, spatial coding, and direct coding [4]. From among these, temporal time-multiplexing coding is widely used. In such systems, a set of patterns are projected onto the scene while the cameras are taking images successively. Binary patterns (e.g., black and white) [1] use only the values 0 and 1 as the basis of the codeword; therefore, it is easy to decode but requires more pattern images as compared to other methods.

Accurately classifying pixels located within the black-and-white stripes is a crucial step. Even though the process is conceptually simple, it is difficult to achieve robust classification in real-world scenes containing complex surface-light interactions

including strong indirect lighting effects. Trobina [8] presented a way to threshold the images by using a single threshold for every pixel. The per-pixel threshold is computed by taking images under all-white and all-black patterns and averaging the two. The author demonstrated that using a pattern and its inverse yields more accurate results. The same strategy is also used by Scharstein and Szeliski [5]. Each pixel is classified based on whether the pixel or its inverse is brighter. These standard methods will not work well when the scene has strong indirect lighting effects. On the other hand, our method produces better classification in such scenarios and also recognizes when a pixel cannot be robustly classified.

To achieve higher accuracy, previous methods also utilize different exposure times [5], multiple intensity illumination images [6] and post-processing to clean up classification results [5]. In this paper, our focus is on improving the pixel classification and structured-light acquisition early in the capture process. This way, we produce better samples sooner and more robustly. Nevertheless, our algorithm also works with more sophisticated methods such as multiple exposures, multiple intensities images, and optional post-processing steps.

### 2.2 Direct and Indirect Illumination Components

The intensity of a pixel in a photograph can be decomposed into the direct component and indirect (or global) component. The direct component is due to light bouncing off the surface in a single reflection. The indirect component is due to multiple reflections (e.g., inter-reflections, subsurface scattering, etc.). Seitz et al. proposed an inverse light transport theory [7] to estimate the inter-reflection component for Lambertian surfaces. This method requires a large number of images to compute matrices used in an inter-reflection cancellation process.

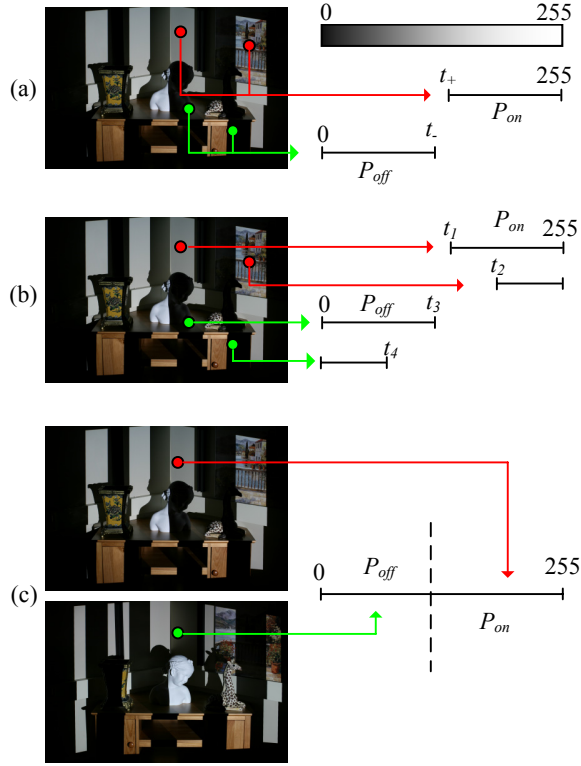
Nayar et al. presented a fast method to separate the direct and global components of a scene lit by a single light source using high frequency illumination patterns [3]. In theory, a high frequency pattern and its inverse are enough to do the separation. In practice, more pattern images, such as a shifting chess board patterns, are used to compensate for the low resolution of the projector. As pointed out by the authors, the higher frequency images of the structured light patterns can also be used to do the separation. Therefore, our method precludes the need for additional capturing and thus can be applied to previously acquired datasets. Furthermore, instead of explicitly seeking to remove the indirect component from arbitrarily illuminated photographs, our method directly attempts to establish bounds of the indirect component and direct component during structured-light acquisition and uses the bounds during classification.

## 3 ROBUST PIXEL CLASSIFICATION ALGORITHM

A pixel classification algorithm in binary-pattern structured-light acquisition uses a set of rules to decide whether a pixel is capturing an illuminated or non-illuminated surface point. Pixels corresponding to surface points visible from the camera but not from the projector should be labeled as uncertain. In the following, we describe our pixel intensity intervals and classification rules for using one or two binary patterns per bit of the codeword.

### 3.1 Pixel Intensity Intervals

To help with classification, we define a pixel’s potential intensity interval. For example, for an 8-bit per channel camera, its value can span at most 0 to 255. This interval can be further subdivided into  $P_{on}$  for when the pixel is directly illuminated and  $P_{off}$  for when the pixel is not directly illuminated. Pixel classification methods generally establish the lower and upper bounds of the two intervals (either explicitly or implicitly). Then, if intensity  $p$  is



**Figure 2. Pixel Intervals.** a) A simple method classifies pixel using two user-defined thresholds. b) An adaptive method computes the albedo for each pixel and uses that for classification. c) A more expensive method classifies a pixel according to whether the pixel or its inverse is brighter.

within one interval but not in the other, the pixel belongs to that category. Otherwise, the pixel is labeled as uncertain.

Consider the following two examples. A simple threshold method assumes  $P_{off}$  belongs to  $[0, t_-]$  and  $P_{on}$  belongs to  $[t_+, 255]$ , where  $t_-$  and  $t_+$  are two user-defined threshold values which may or may not be the same. Pixels can be classified by comparing their intensities against the thresholds as shown in Figure 2a. A more accurate method uses the albedo of a pixel as the classification threshold  $t$ . Each pixel has a different threshold  $t$  and thus yields a different  $P_{on}$  and  $P_{off}$  per pixel as illustrated in Figure 2b [8]. The albedo can be computed by taking two images under all-white and all-black illuminations and averaging the two. Methods that project a pattern and its inverse assuming the two intervals are non-overlapping, i.e. the lower bound of  $P_{on}$  is larger than the upper bound of  $P_{off}$  [5]. In this case, a single comparison between the pixel and its inverse decides which interval the pixel falls into without explicitly computing  $t$  (Figure 2c).

All these methods assume that the two intervals are non-overlapping. However, this is not necessarily true if the scene point is undergoing strong indirect illumination. Our method overcomes the problem by correctly establishing the lower bounds and upper bounds for  $P_{on}$  and  $P_{off}$ . With these intervals, our algorithm classifies pixels as on/off accurately.

Furthermore, our algorithm can robustly reject pixels that are not visible from the projector or problematic due to excessively strong indirect lighting. This is important because incorrect classification leads to inaccurate decoding and then to bad reconstruction.

### 3.2 Single Pattern Classification Rules

We first derive the decision rules for classification using a single pattern per bit of the codeword. The classification rules involve a sequence of comparisons. For a directly illuminated pixel, its intensity  $p$  can be decomposed into two components: direct component  $d$  and indirect component  $i_{on}$ . The direct component is the response to the direct light from the projector; therefore,  $d$  is invariant under different illumination patterns. On the other hand, the indirect component  $i_{on}$  depends on the bidirectional reflection distribution function (BRDF) at the scene point, the radiance of every surface patch in the direction of the scene point, the relative geometric configurations between the point and other surface patches, and set of the surface patches that are lit. Without detailed scene information, this global component is difficult to compute. For an indirectly illuminated pixel, its intensity  $p$  only contains the indirect component  $i_{off}$ . In summary,

$$p = \begin{cases} d + i_{on} & \text{If pixel is on} \\ i_{off} & \text{If pixel is off} \end{cases} \quad (1)$$

Since the direct component  $d$  of an illuminated pixel is invariant to the illumination pattern used, we can compute  $d$  for each pixel using the separation method introduced by Nayar et al. [3]. Their algorithm estimates the per-pixel direct component  $d$ , and total indirect component  $i_{total}$  for a scene lit by all projector pixels. **Note that indirect component  $i_{on}$  and  $i_{off}$  depend on the illumination pattern and the scene geometry; thus, they are different from  $i_{total}$ .**

After  $d$  is computed, determining the intervals  $P_{on}$  and  $P_{off}$  becomes a problem of finding the lower bounds and upper bounds for  $i_{on}$  and  $i_{off}$ . Both  $i_{on}$  and  $i_{off}$  are indirect components of the pixel when about half of the projector pixels are on. Therefore, they are smaller than or equal to the total indirect component  $i_{total}$  because a scene point receives more indirect light when all projector pixels are on. As intensity values, they are also larger than or equal to zero. Thus,

$$i_{on} \in [0, i_{total}] \quad (2)$$

$$i_{off} \in [0, i_{total}] \quad (3)$$

From (1), (2), and (3), we establish the lower and upper bounds for interval  $P_{on}$  and  $P_{off}$ :

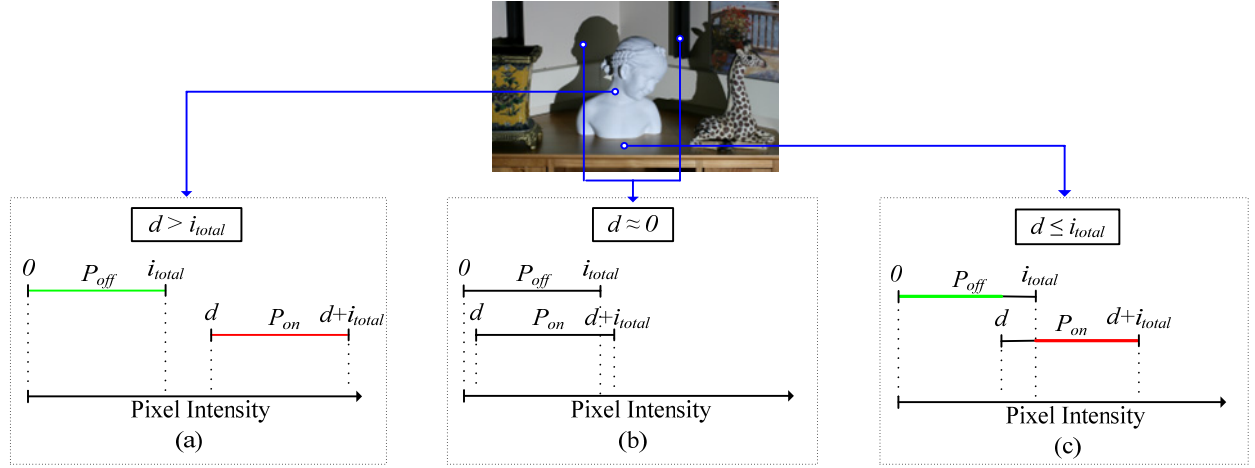
$$P_{on} \subseteq [d, d + i_{total}]$$

$$P_{off} \subseteq [0, i_{total}]$$

As shown in Figure 3a, when  $d > i_{total}$ , i.e. the scene point has a stronger direct component, the two intervals are completely separated. In this case, the decision rules are as follows:

$$\begin{aligned} \text{Rule 1: } & p < i_{total} \rightarrow \text{pixel is off} \\ & p > d \rightarrow \text{pixel is on} \\ & \text{otherwise} \rightarrow \text{pixel is uncertain} \end{aligned} \quad (d > i_{total})$$

The two intervals are very similar to each other when  $d$  is close to zero (as in Figure 3b). This situation can happen when the surface point is not visible from the projector, i.e., it is in shadow. **Thus, the pixel should be discarded from reconstruction. This situation can also occur for a visible pixel with a very small direct illumination component. In this case, the indirect light from other parts of the scene has a huge impact on its observed intensity. We do not have sufficient information to robustly know why the pixel is brighter and hence the pixel should be discarded.** Our algorithm detects these situations and classifies the pixel as uncertain when  $d$  is smaller than a predefined minimum threshold  $m$ .



**Figure 3. Pixel Classification Scenarios.** a) The two intervals are completely separated when  $d > i_{total}$ . b) The two intervals are indistinguishable when  $d \approx 0$ . c) The two intervals overlap when  $d \leq i_{total}$ .

Rule 2:  $d < m \rightarrow$  pixel is *uncertain* ( $d \approx 0$ )

When  $d \leq i_{total}$ , the pixel has a relatively stronger indirect component and the two intervals overlap near the middle range. This is shown in Figure 3c. The pixel can be labeled as on/off only if its intensity  $p$  is smaller than the lower bound of  $P_{on}$  or larger than the upper bound of  $P_{off}$ . Closer values of  $d$  and  $i_{total}$  produce larger classifiable intervals. Therefore, we have the following decision rules:

Rule 3:  $p < d \rightarrow$  pixel is *off*  
 $p > i_{total} \rightarrow$  pixel is *on*  
 otherwise  $\rightarrow$  pixel is *uncertain* ( $d \leq i_{total}$ )

Combining the rules for the three different cases together, we derive the following single pattern classification rules:

Table 1. Single pattern classification rules.

$d < m \rightarrow$ pixel is <i>uncertain</i>
$p < \min(d, i_{total}) \rightarrow$ pixel is <i>off</i>
$p > \max(d, i_{total}) \rightarrow$ pixel is <i>on</i>
otherwise $\rightarrow$ pixel is <i>uncertain</i>

### 3.3 Dual Pattern Classification Rules

Projecting the code pattern and its inverse yields two values for each pixel which can be used to improve robustness. Both pixel values,  $p$  and  $\bar{p}$ , obey the same single pattern classification rules. The single pattern rules can be combined and extended to form dual pattern classification rules (see Table 2). In this way, our algorithm performs an on/off classification only when a pixel and its inverse exhibit consistent behaviors.

Table 2. Dual pattern classification rules.

$d < m \rightarrow$ pixel is <i>uncertain</i>
$d > i_{total} \wedge p > \bar{p} \rightarrow$ pixel is <i>on</i>
$d > i_{total} \wedge p < \bar{p} \rightarrow$ pixel is <i>off</i>
$p < d \wedge \bar{p} > i_{total} \rightarrow$ pixel is <i>off</i>
$p > i_{total} \wedge \bar{p} < d \rightarrow$ pixel is <i>on</i>
otherwise $\rightarrow$ pixel is <i>uncertain</i>

It is worth noting that when  $d > i_{total}$ , the two intervals are completely separated. Hence, the mapping from a pixel and its

inverse to the two intervals is one-to-one. The classification rules can be simplified as the brighter one among the two must be directly illuminated. In other words,  $d > i_{total}$  is a sufficient condition for a brighter pixel among a pixel and its inverse to be directly illuminated, and is the assumption used in some previous methods (e.g., [5]).

### 3.4 Overlapping Pixel Intervals

In order to improve the classifiable regions for the ambiguous case when  $d \leq i_{total}$ , we need to decrease the overlap between the intervals. This could be accomplished by either increasing the lower bound of  $P_{on}$  or decreasing the upper bound of  $P_{off}$ . However, given our limitation of not knowing the scene geometry *a priori*, these bounds are already tight, and thus we are limited to classifying pixels in that range as uncertain.

Consider the following two scenarios regarding an observed point (including its corresponding image pixel) and a surface patch elsewhere in the scene. The scene is such that the point receives indirect light only from the surface patch. The patch itself does not receive any indirect light. With an illumination pattern, it might be the case that the point is “on” and the surface patch is “off”. In this case, the patch does not provide any indirect light for the point. The intensity of the point’s corresponding image pixel only contains its direct component  $d$ . This is a minimum condition for when the intensity of an illuminated pixel reaches the lower bound  $d$  of interval  $P_{on}$ .

Next, consider the case of when a different illumination pattern makes the point “off” and the patch “on”. The point’s only source of illumination is the indirect light from the single patch. This implies the point’s  $i_{total}$  is only a function of the light from the patch. Since the patch does not receive any indirect light, its irradiance is a result of the direct light it receives and thus is constant when lit. Therefore, the light the patch gives to the point is also constant and does not change as long as the patch is lit. This is precisely the definition of  $i_{total}$  and hence the intensity of point’s corresponding pixel equals to  $i_{total}$ . This is the condition when the intensity of a non-illuminated pixel reaches the upper bound  $i_{total}$  of interval  $P_{off}$ . Without knowing the geometry, our lower bound of  $P_{on}$  and upper bound of  $P_{off}$  is already tight.

## 4 RECONSTRUCTION AND RENDERING

To show the classification results of our algorithm, we implemented a reconstruction and rendering engine, which uses two mutually calibrated cameras, a digital uncalibrated projector, and a standard graphics card. For each camera pixel, we



concatenate the bits from all our binary classification images, ignoring any pixel with uncertain bits, and then correspond, reconstruct, and render the scene.

Establishing correspondences implies identifying surface points observed by both cameras. Pixel classification produces a set of candidate camera pixels for each projector pixel illuminating the scene. In practice, digital cameras are often higher resolution than projectors and thus several nearby camera pixels decode the same projector pixel codeword. We group the pixels and use the center as the overall position. To ignore gross misclassifications, a simple image-space culling method removes same-code pixel clusters that span too much image area.

Corresponded pixels are triangulated to obtain the 3D location of a scene point. Triangulation accuracy depends on the baseline and calibration accuracy of the two cameras. In our system, we use high-resolution and carefully calibrated cameras to obtain good triangulation results. Nevertheless, correspondence and calibration errors may cause erroneously reconstructed scene points that are excessively distant from their neighboring scene points – these points are trivially culled from the solution set.

Finally, color renderings are produced by splatting each point with the corresponding color value from the reference image. Each splat consists of an object-space quadrilateral whose size and orientation is determined by triangulating the nearby scene points in image space. In a first visibility pass, the visible surfaces are found by rendering larger than necessary splats to the z-buffer. In a second pass, the smallest splats that still cover the object surfaces are rendered and blended with the color buffer [2][9].

## 5 IMPLEMENTATION DETAILS

We use two Canon Digital XTi SLR cameras, each capturing images at a resolution of 3888 by 2592, and an Optoma DLP projector of resolution 1400 by 1050. During a capturing session, 20 binary Gray code [1] patterns and their inverses are projected onto the scene. Of these patterns, 10 are horizontal stripe patterns and the remaining 10 are vertical stripe patterns. The higher frequency structured light patterns and their inverses (level 8 and level 9 in this paper) are used to separate the direct and indirect components for each pixel. All software is implemented on a Dell PC with 3.0GHz CPU and 2GB memory. Separation using the 4 patterns on a scene takes about 20 seconds for each camera. Classifying all images from each camera takes about 70 seconds.

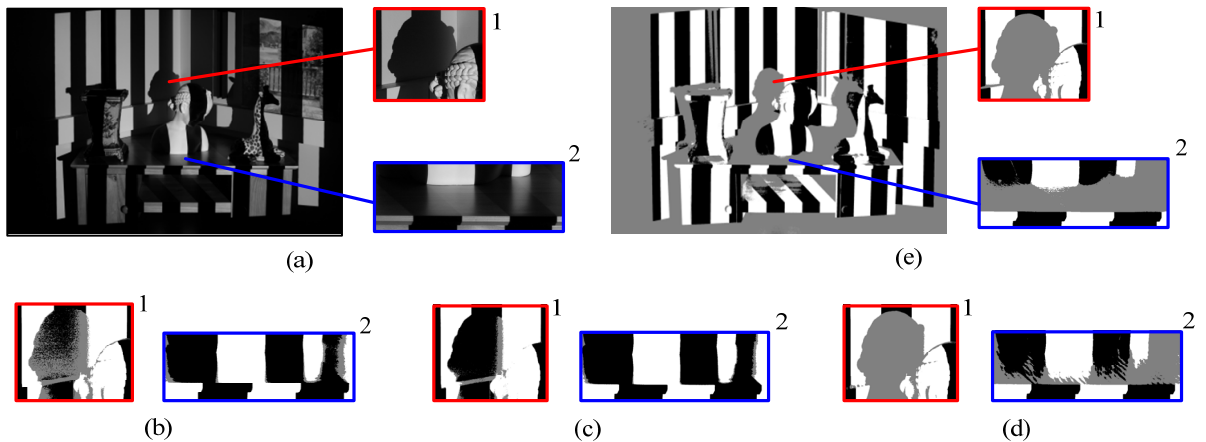
The lower bounds and upper bounds derived in Section 3 are for ideal scenarios. In practice, due to the light coming out from the deactivated projector pixels and “fogging” inside the projector that adds light to the patterns, it is not exact. To compensate this, we use a small  $\epsilon$  to conservatively reject pixels that are close to the interval boundary. The same  $\epsilon$  is also used for standard methods to achieve reliability as used in [5]. In our experiment, we found that a value of 5 (out of 255 gray levels) works well.

## 6 RESULTS AND DISCUSSION

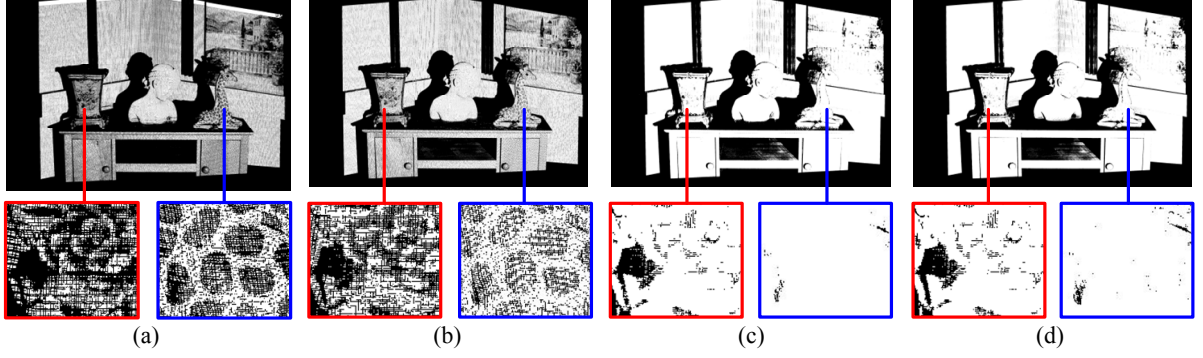
We have tested our algorithm on three different scenes, each of different complexity and each with several objects of diverse materials. To compare the quality of our pixel classification algorithm, we also implemented two standard pixel classification methods for structured light systems. Standard method 1 uses the average of the two images captured with all white and all black patterns as the threshold to classify each pixel. Standard method 2 determines whether a pixel is directly illuminated according to whether the pixel or its inverse is brighter.

Figure 4 shows the classification results for a corner scene. White, black, and gray pixels in the classification images represent on, off, and uncertain, respectively. Figure 4a is the input image of the scene with two zoomed-in areas. Figure 4b and 4c show the classification results using standard methods 1 and 2 on these areas. Figure 4d shows the improved classification using our method and using the same structured light patterns for separating total direct and indirect illumination. Our method not only correctly classifies pixels in the shadow as uncertain (rectangle 1), but also reduces misclassifications on the table top due to strong inter-reflection (rectangle 2).

Our results also show that, in practice, the additional separation quality obtained by using more images does not produce a significant difference for classification. For example, Figure 4e shows further improved separation results that can be obtained but at the expense of many additional captured images. The images used for total direct/indirect illumination separation in Figure 4e are projected chess board patterns with 4 by 4 black and white blocks shifted one pixel each time and 49 times along both  $X$  and  $Y$  directions. Using the structured light patterns for separation generates slightly more reconstructed points because accurate separation will result in more conservative classification, and then less reconstructed points. Although using additional patterns lead



**Figure 4. Classification Results.** White pixels are classified as on, black pixels are classified as off, and gray pixels are uncertain. a) We show an input image with a pattern projected onto the scene with zoomed-in areas of a shadow (rectangle 1) and of a table top with strong inter-reflection (rectangle 2). b) Classification results on zoomed-in areas using standard method 1. c) Classification results on zoomed-in areas using standard method 2. d) Classification results using our algorithm with structured light pattern separation. e) Classification results using our algorithm with additional pattern images separation.



**Figure 5. Binary Images for Decoded Pixels.** White means a pixel can be decoded without any uncertainty. Images are generated using: a) standard method 1, b) standard method 2, c) our method using structured light pattern images for separation, and d) our method using additional pattern images for separation. More decoded pixels lead to a brighter image.

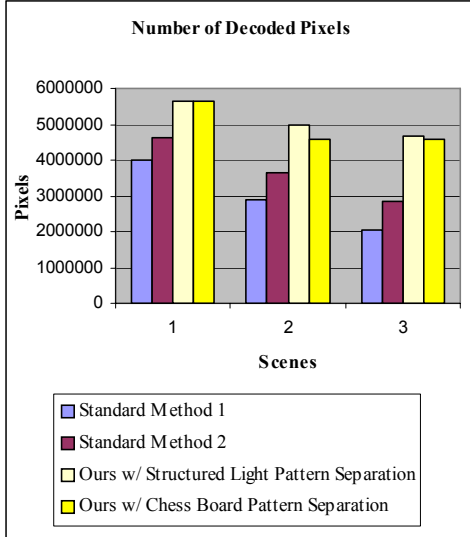
to more accurate reconstruction, we use the higher frequency structured light patterns for separation, which requires significantly less effort and still achieve roughly the same quality.

As compared to other methods, our algorithm may yield relatively more uncertain pixels in one pattern image but over all produces more decoded pixels due to robust classification. Decoded pixels are those that can be fully classified as either on or off in all the images. The quantity improvement is demonstrated in Figure 5 using the same corner scene. It shows binary images with decoded pixels in white and uncertain pixels in black. Therefore, an image with more decoded pixels will appear to be brighter. Our methods (Figure 5c and 5d) clearly produce denser decoded pixels than standard methods (Figure 5a and 5b). Graphs in Figure 6 plot the total numbers of decoded pixels for our three scenes using all four methods. Our improvement is about 20% to 60% depending on the scene.

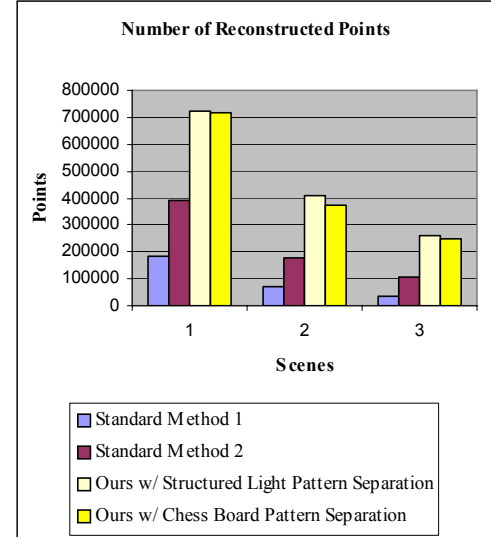
Besides the number of decoded pixels, our method also improves the reconstruction quality. Figure 7 shows the number of reconstructed points after culling away outliers using the same

thresholds for all methods. Each group shows the results of one scene. It is worth noting that although the improvement of the total number of decoded pixels is at most 60%; our algorithm reconstructs 1.8 to 2.5 times the number of points as compared to the better standard method. This implies that the decoding in our method is more accurate because pixels are less likely to be culled away by simple outlier removal.

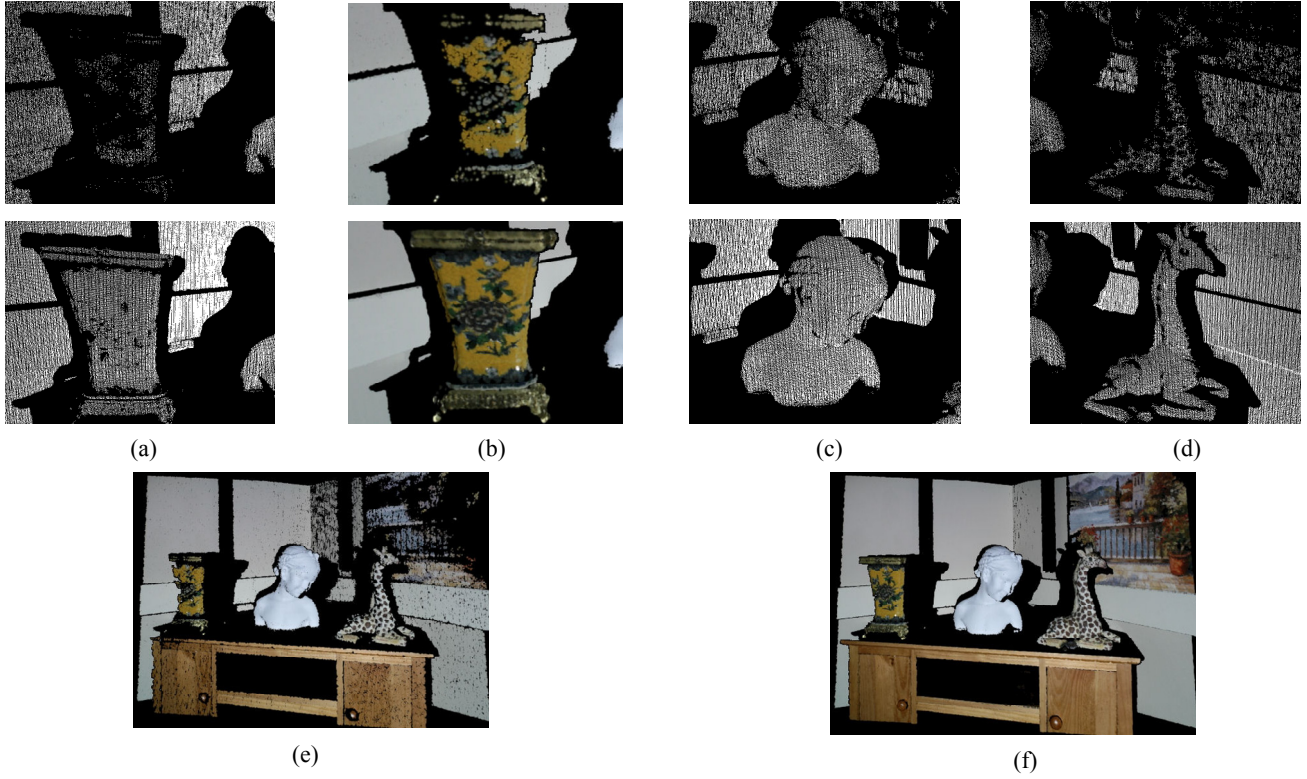
Figures 1, 8, 9, 10 show the quality of the reconstructed 3D point clouds both for our method and for standard method 2. The black and white images among these figures simply show the reconstructed points as white dots. A higher point cloud density leads to a brighter image. The color images are generated using our point-based rendering implementation. Figures 8 and 9 clearly show the higher number of samples we are able to robustly decode and reconstruct for the corner room and objects scene. Moreover, a point-based rendering scheme has the advantage of precluding us from having to worry about difficult 3D triangulations. The denser samples produced by our approach yield better quality imagery than simply increasing the splat size



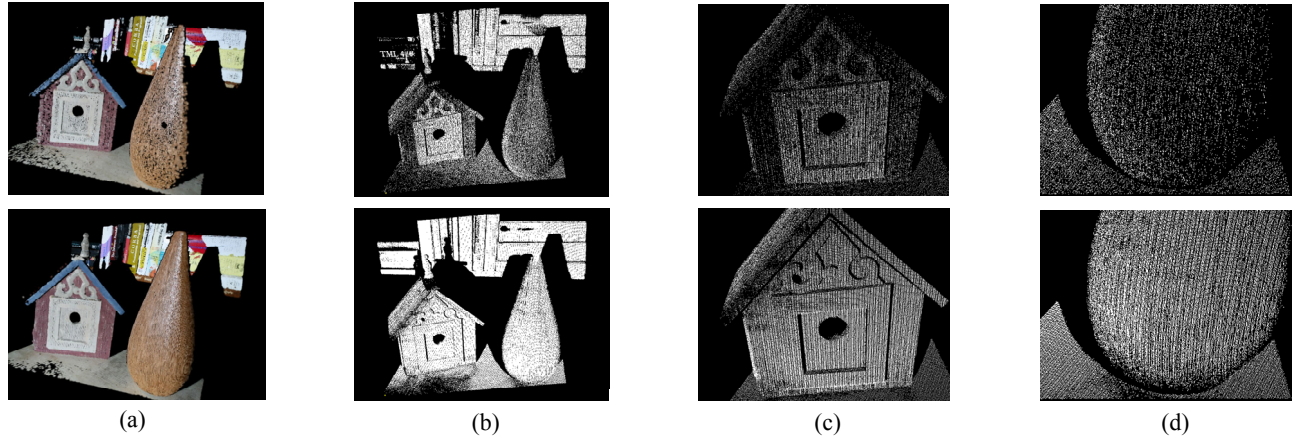
**Figure 6. Decoded Pixels.** Number of decoded pixels in three different scenes using two standard methods and our method with structured light pattern separation and with additional pattern image separation: 1) a corner scene, 2) a table top scene with wooden objects, and 3) a table top scene with two shiny objects. The improvement is about 20%-60%.



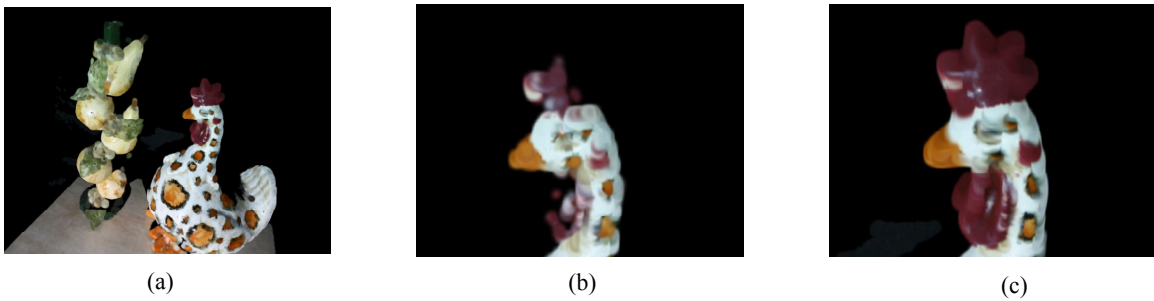
**Figure 7. Reconstructed Points.** Number of points for three different scenes using two standard methods and our method with structured light pattern separation and with additional pattern image separation. The improvement is about 1.8 to 2.5 times. Please note that the numbers of reconstructed points are much smaller than the numbers of decoded pixels due to the limited resolution of the projector.



**Figure 8. Point Clouds.** In (a-d), pictures in the first row are generated using standard method 2. Pictures in second row are generated using our algorithm: a) a close up view of the vase, b) point-based rendering of the vase, c) a close up view of the bust, and d) a close up view of the giraffe. e) The point-based rendering of the entire scene using standard method 2. f) The point-based rendering of the entire scene using our method.



**Figure 9. Objects Scene Point Clouds.** a) Point-based rendering of the scene. b-d) Several views of the reconstructed point clouds. The first row are the results using standard method 2 and the second row are the results using our method.



**Figure 10. Shiny Objects Scene Point Clouds.** a) Point-based rendering of the scene using our method. b) A close up view on the head using standard method 2. c) A close up view on the head using our method.



to compensate for the missing points (see Figure 10).

Figure 11 shows the percentage of outliers under progressively stricter culling criteria. As can be seen, the outlier percentage of standard method 2 is always higher than that of our method. Moreover, it increases much faster than that of our algorithm: only about 68% of the original points remain using the standard method; while 96% remain using our method.

Finally, our algorithm also works when the scene has environment ambient light. This requires us take an image of the scene with environment light only. Then the ambient component is subtracted from each of the input pattern images. The results are shown in Figure 12.

## 7 CONCLUSIONS AND FUTURE WORK

We have presented an efficient, robust and easy to implement algorithm for classifying a pixel to be on/off/uncertain in structured light systems using binary patterns. Our method correctly classifies pixels and rejects ambiguous pixels based on their intensities. Our experiments show that as compared to naïve methods, our algorithm improves both the quantity and quality of the reconstructed points. This produces dense datasets suitable, for instance, for point-based modeling. Furthermore, since some of the structured light patterns can be used to obtain direct and indirect separation, our method can be applied to previously captured data.

One limitation of our algorithm is being conservative. As long as the pixel intensities are in the ambiguous range, our method classifies the pixel as uncertain. **For an object with strong sub-surface scattering, its indirect component can be much larger than its direct component. Therefore, our algorithm will classify many pixels as uncertain due to a small direct component.** As shown in Figure 13, the teddy bear has a small direct component and a very large indirect component. Our algorithm produces very few decoded pixels while a naïve method can produce at least a very coarse reconstruction. As a matter of fact, if the classification threshold  $\epsilon$  is set to zero, a naïve method will decode every pixel by allowing many misclassifications. Our method always makes conservative but accurate decisions due to the tight bounds of the intervals.

In the future, we would like to extend our idea to structured light systems using  $n$ -ary code patterns (e.g. multi-level gray). This involves establishing bounds for each gray level of the pattern instead of only two. We would also like to find better intensity bounds for a scene. This could be done by taking more images of the scene under structured light patterns using different camera/projector parameters.

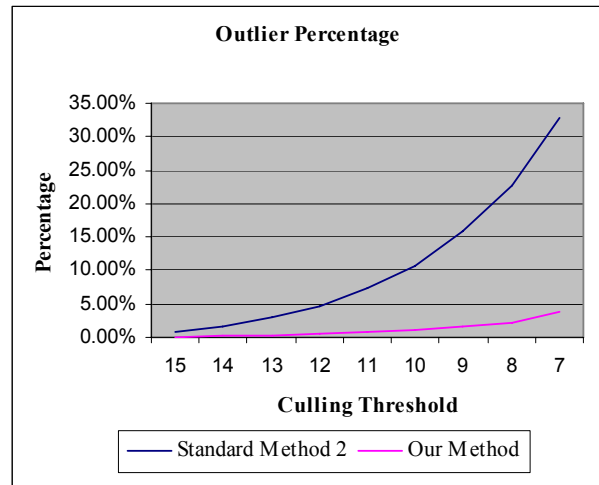
## ACKNOWLEDGEMENTS

We would like to thank the reviewers for their valuable suggestions to improve this paper. This work was supported by NSF CCF 0434398 and by a Purdue Research Foundation grant.

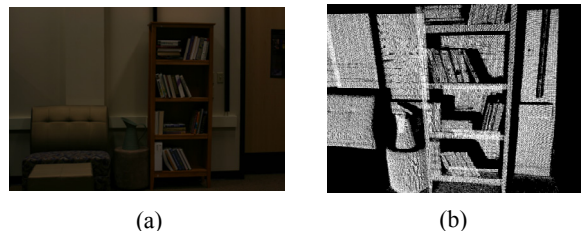
## REFERENCES

- [1] Inokuchi, S., Sato, K., and Matsuda, F. Range imaging system for 3-D object recognition. *Proc. ICPR 1984*, 806-808.
- [2] Kobbelt, L. and Botsch, M. A survey of point-based techniques in computer graphics, *Computers and Graphics*, 2004, 28(6), 801-814.
- [3] Nayar, S., Krishnan, G., Grossberg, M., and Raskar, R. Fast separation of direct and global components of a scene using high frequency illumination. *Proc. of ACM SIGGRAPH 2006*, 935-944.
- [4] Salvi, J., Pages J., and Batlle, J. Pattern Codification strategies in structured light systems. *Pattern Recognition* 37 (2004), 827-849.
- [5] Scharstein, D. and Szeliski, R. High-accuracy stereo depth maps using structured light. *Proc. CVPR 2003*, 195-202.

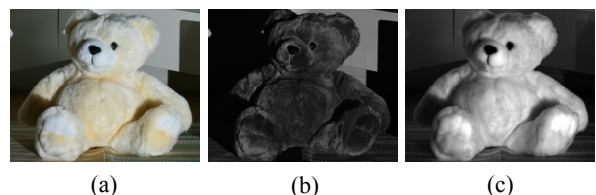
- [6] Skocaj, D. and Leonardis, A. Range image acquisition of objects with non-uniform albedo using structured light range sensor, *Proc. ICPR 2000*, 778-781.
- [7] Seitz, S., Matsushita, Y., and Kutulakos, K. A theory of inverse light transport. *Proc. ICCV 05*, 1440-1447
- [8] Trobina, M. Error model of a coded-light range sensor, *Technique Report, Communication Technology Laboratory, ETH Zentrum, Zurich*, 1995.
- [9] Zwicker, M., Pfister, H., van Baar, J., Gross, M. EWA splatting, *IEEE TVCG*, 2002, 8(3), 223-238.



**Figure 11. Outlier Percentage.** We cull the corner scene outliers using a threshold for both world space culling (in mm) and image space (in pixels). The horizontal axis is the culling threshold. Our method results in much fewer outliers than standard method.



**Figure 12. Ambient Lighting.** For a scene under significant ambient lighting, our method still performs well: a) a picture of the scene taken under ambient light and b) a reconstructed point cloud using our method.



**Figure 13. Limitations.** Our algorithm fails when the indirect illumination is excessively strong: a) a picture of a teddy bear, b) its direct component in grayscale, and c) its indirect component in grayscale.