An Adaptive Correspondence Algorithm for Modeling Scenes with Strong Inter-reflections

Yi Xu and Daniel G. Aliaga

Abstract— Modeling real-world scenes, beyond diffuse objects, plays an important role in computer graphics, virtual reality, and other commercial applications. One active approach is projecting binary patterns in order to obtain correspondence 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. When a scene has abundant indirect light, this process is especially difficult. In this paper, we present a robust 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 but not in the other. Our method performs better than standard method due to the fact that it avoids gross errors during decoding process caused by strong inter-reflections. For the remaining uncertain pixels, we apply an iterative algorithm to reduce the inter-reflection within the scene. Thus, more points can be decoded and reconstructed after each iteration. Moreover, the iterative algorithm is carried out in an adaptive fashion for fast convergence.

Index Terms— Three Dimensional Graphics and Realism, Digitization and Image Capture, Geometric Modeling.

1 Introduction

Creating high-quality models for real-world scenes, such as shiny ornaments and artifacts, plays an important role in computer graphics, virtual reality, and other commercial applications. Active methods, e.g. structured-light systems, add light into the scene and thus, are more robust than passive methods (e.g. stereo vision [26]) that only receive energy from the scene. One popular option among structured-light systems is a camera-projector system. A naïve system will turn on one or a cluster of projector pixels at a time and search for the single dot being projected onto the scene. To reduce the total number of projections, many different codification strategies have been proposed [22]. Pairs of camera and projector pixels that see the same codeword are corresponded and triangulated to obtain 3D scene samples allowing for point-based modeling and rendering [10], and other graphics applications, such as surface reconstruction [8]. Among the many strategies, time-multiplexed codes, e.g. Gray codes [9], yield robust and high resolution 3D information with a relatively small number of patterns. To achieve greater precision, phase shifting patterns, such as sine waves [25], can also be applied.

1.1 Problem Statement

A fundamental assumption of all these methods is that the direct illumination falling onto an observed pixel sample is greater than its total indirect (or global) illumination (e.g. illumination resulting from inter-reflection and subsurface scattering). When this condition is not met, a correct decoding of all camera pixels is very difficult to achieve. This problematic scenario occurs even for diffuse surface materials because of strong diffuse inter-reflection. In general, the unexpected illumination might cause distant camera pixels to see the same codeword, the stripe boundaries of binary patterns to shift from their true positions and/or the phase of sinusoidal patterns to be disturbed. The result is incorrect correspondences and either a bad reconstruction or a large loss of samples. Consider the following two 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 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.

To achieve more accurate decoding and greater number of samples in the presence of complex lighting effects, previous methods attempt to project binary patterns and their inverses [25][30], to use several camera exposure times [25], to project multiple patterns with different intensities [27], or to adapt the pattern intensity locally [11]. Nevertheless, most of the binary pattern methods assume that a scene point is brighter when it is directly illuminated; e.g., a directly illuminated pixel is classified as "1" and "0" otherwise. However, this assumption only holds when a scene point has a relatively weak indirect light component. Furthermore, methods using phase shifting patterns assume a linear image formation process when

- *Yi Xu is a Ph.D. candidate in the Department of Computer Science, Purdue University, 305 N. University St., West Lafayette, IN, 47907. E-mail: xu43@cs.purdue.edu.*
- *Daniel G. Aliaga is an Assistant Professor of Computer Science at Purdue University, 305 N. University St., West Lafayette, IN, 47907. E-mail: aliaga@cs.purdue.edu.*

Manuscript received (December 12, 2007). Final acceptance June, 2008.

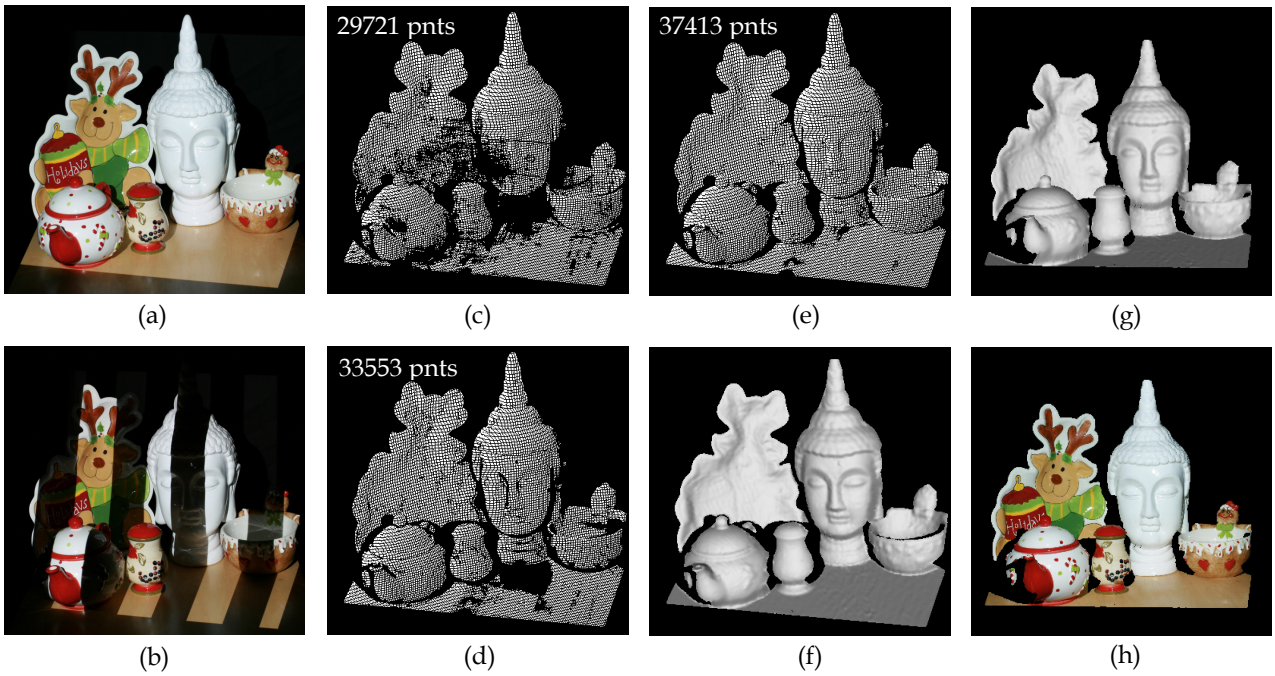


Fig. 1. **Modeling Scenes with Strong Inter-reflections.** a) A picture of the scene. b) The same scene under the illumination of a structured light pattern using one projector. The indirect light is abundant in the scene. c) 3D point cloud reconstructed using standard pixel classification during decoding. d) 3D point cloud reconstructed using our robust pixel classification algorithm for one step. e) After 22 iterations, almost all surfaces visible from the camera's viewpoint are reconstructed. f) A synthetically-shaded model of the scene from the same viewpoint. g) The same synthetically-shaded model of the scene from a different and novel viewpoint. h) A texture mapped triangular mesh is rendered from the same novel viewpoint as that of g).

projecting patterns (e.g. sinusoid) with different phases. The direct component of a scene point will scale linearly according to the phase of the pattern. However, the indirect component depends on the complex interaction of scene geometry, scene reflectance, and pattern geometry; thus, it is non-linear. This makes such methods fragile. While the total direct and total indirect components of a *fully* lit scene can be separated without knowledge of scene geometry (e.g., [20]), the indirect component for *arbitrary patterns* depends on the pattern, the scene geometry and the scene reflectance properties. This produces the chicken-and-egg problem of needing to know scene geometry and reflectance before identifying scene illumination properties and needing to know scene illumination properties in order to perform robust structured-light scene acquisition.

1.2 Observations

The two key observations of our method are (1) we can use a divide-and-conquer approach to iteratively and adaptively discover a set of binary patterns that progressively reduce the indirect light but keep the direct light of some pixels constant and (2) we can estimate tight intensity value bounds for when a pixel is on and for when it is off under the illumination of arbitrary binary patterns. The combination of these observations enables robustly classifying a pixel when its indirect component for the current pattern is smaller than its direct component (which is independent of the pattern). Pixels that cannot be initially classified will have their indirect component iteratively reduced by the next pattern, thus eventually producing a larger number of robustly reconstructed points.

In general, a solution to the aforementioned fundamental problem is to either increase direct light or decrease indirect light per pixel while keeping the other constant. Both enable more robustly identifying directly illuminated pixels and thus improving the decoding process. On the one hand, simply increasing the amount of direct light is not a viable solution because the indirect light will also increase. On the other hand, we can *decrease* the amount of indirect light while keeping problematic pixels with the *same* amount of direct light. Hence, the directly illuminated pixels can be better decoded. Decreasing indirect light is possible by using a top-down approach of illuminating progressively smaller parts of the scene with adaptive patterns. To be more specific, we decrease ambiguous pixels' indirect components by turning off projector pixels whose codes have already been observed by the cameras. The pixels that are directly illuminated will still receive the full direct illumination but all pixels receive an equal or weaker indirect illumination.

While an alternative bottom-up approach is also possible, it would require providing an initial way to segment the image and would be inefficient for pixel regions that do not suffer from having stronger indirect light than direct light. Rather, a top-down approach and a way to bound pixel intensity values for on/off classification enables only performing additional work in the problematic areas of the scene and thus extending a standard structured-light method only in the necessary regions.

1.3 Summary

We present an approach using a structured-light based method with Gray code binary patterns which iteratively obtains an increased number of accurately reconstructed

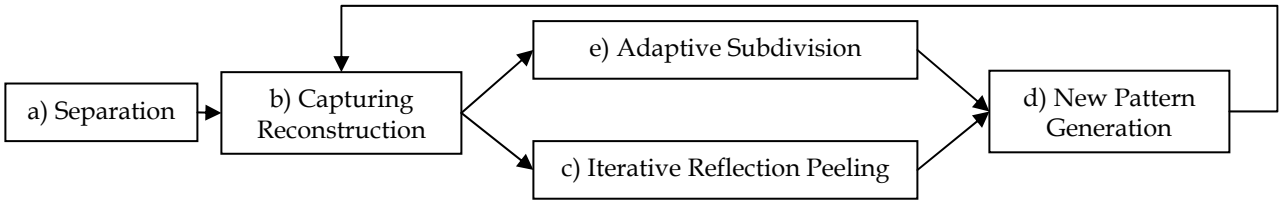


Fig.2. **Pipeline of Our Adaptive Structured-Light Algorithm.**

scene points as compared to the conventional approach (Fig. 1). Our algorithm adapts the structured-light patterns according to the partially reconstructed scene geometry and reflectance property of the scene so as to reduce the amount of indirect illumination, while keeping the same amount of direct illumination for the remaining non-decoded pixels. Pixels are attempted to be corresponded using a robust decoding algorithm based on estimated tight intensity value bounds. A projector-space subdivision is used whenever the number of additionally decoded pixels for an iteration is too small. Parallelism is automatically exploited during subdivision in order to simultaneously decode as many non-conflicting pixel subsets as possible and, consequently, reduce acquisition time. Fig. 2 summarizes our entire algorithm pipeline. After separation of the scene under the illumination of a fully lit projector into its direct and indirect components (Fig. 2a), we capture images and reconstruct 3D points (Fig. 2b). Based on the reconstruction result, we reduce the inter-reflection within the scene (Fig. 2c) and generate new projector patterns (Fig. 2d). Then we iterate the capturing and reconstruction again. If the gain for an iteration is too small, we adaptively subdivide the projector image in order to further reduce inter-reflection (Fig. 2e). After each iteration, the reconstruction of the scene is more complete. The algorithm continues until no more points can be decoded and reconstructed. The end result is a significantly improved model of the scene including objects with strong inter-reflections at the cost of more photographs compared to a standard structured-light method. We have used our approach to capture several datasets of non-diffuse (e.g., glossy) 3D objects. Our method is consistently able to increase the number of correctly reconstructed points by almost 2x and to decrease the number of incorrectly classified pixels by about 10x.

Our major contributions can be summarized as

- a robust method to classify a pixel as directly illuminated or not, based on the intensity intervals of a pixel under the illumination of a binary pattern,
- an adaptive structured-light algorithm which iteratively reduces the per-pixel indirect component and thus allows better reconstruction, and
- an acquisition system which can acquire complex scenes with strong inter-reflections.

The work introduced in this paper significantly extends our previous conference publication [33] by presenting an iterative and adaptive structured-light method enabling a complete reconstruction of complex scenes.

2 Related Work

As related work, we first review various other approaches to modeling optically-challenging objects, and then focus on difficulties with using structured light to acquire such objects, provide an outline of adaptive structured-light systems and finally a summary of methods for separating illumination components. Altogether, our research builds upon these works to create an adaptive structured-light system for objects with strong inter-reflections by using binary patterns and by using insight provided by direct and indirect component separation.

2.1 Modeling Optically-Challenging Objects

Methods have been proposed for acquiring objects in the presence of inter-reflection, as well as other phenomena, such as specular reflection, subsurface scattering, and transparency. Some of the early works focus on passive approaches that attempt to handle inter-reflection during acquisition. For example, Nayar et al. [18][19] presented a pioneering approach using an iterative algorithm to estimate scene geometry and reflectance in the presence of inter-reflections. This algorithm builds upon a shape-from-intensity method and estimates shape and inter-reflection in an alternating fashion for Lambertian surfaces. Wada et al. [32] tackled the problem of recovering the shape of an unfolded book using the image obtained by a typical flatbed image scanner. Their method is also based on a formulation from shape-from-shading with inter-reflections and from moving a proximate light source. A piecewise polynomial model is used to approximate the geometry of the unfolded book. Yang et al. [34] uses inter-reflection as a constraint to uniquely determine the shape of simple concave polyhedron. Chandraker et al. [3] incorporate inter-reflection modeling into Lambertian-surface photometric stereo. Since inter-reflection is not preserved under general bas-relief (GBR) transformation, it has been used to successfully resolve the GBR ambiguity inherent in un-calibrated photometric stereo [1]. In general, these shape-from-shading and photometric-stereo methods do not produce robust and accurate 3D positional samples as compared to triangulation-based structured-light approaches. Moreover, they are only applied to simple geometrical structures where analytic inter-reflection models may be faithfully constructed. In contrast, our method does not rely on the accurate modeling of inter-reflection. Rather, we use an adaptive method that seeks to physically reduce the inter-reflection component for ambiguous pixels, to increase correspondence robustness, and to produce an improved 3D model.

Active methods, such as laser-scanning, have also been applied to optically-challenging objects. Curless and Levoy [7] proposed a robust space-time processing method to reduce the distortion in the acquired range data caused by non-uniform illumination. However, inter-reflection is not considered in this work. Clark et al. [5] presented a method that uses polarized incident laser light to model metal objects. By predicting the polarization state of the direct reflection, they can successfully distinguish between true laser stripe and spurious inter-reflections. Park and Kak [21] use a multi-peak range imaging method to capture complex geometry of specular objects. They store multiple samples per candidate object point and obtain the correct shape by applying a number of local and global consistency tests. Unlike these approaches, our structured-light method is based on a time-multiplexed Gray code, which in general requires relatively fewer images obtained using readily-available digital hardware and does not require any special polarization filters for the cameras. Furthermore, instead of relying on detecting false measurements in 3D space after triangulation, our method attempts to rigorously reject potentially ambiguous camera pixels before triangulation. This leads to more robust results.

Other interesting optical characteristics can also create problems for 3D modeling. For instance, Chen et al. [4] capture translucent objects using a combination of phase-shifting patterns and polarization-difference imaging based on the fact that subsurface scattering depolarizes the incident light. However, specular reflection changes the polarization direction instead of depolarizing the incident light. Unless the exact number of inter-reflections at each scene point is known, a polarization based method is not suitable for objects with strong inter-reflections. Other 3D geometry modeling approaches include, but are not limited to, acquiring transparent objects using polarization analysis [16], light-path triangulation [12], and environment matting [15]. In particular, Tarini et al. [29] use environment matting to model mirroring objects using a shape-from-distortion approach. However, while these works address optically-challenging objects, they do not focus, or in some cases even handle, inter-reflections. Our observation is that the global illumination phenomena of inter-reflections is very common (and important) and even occurs in scenes where the attempt is to be mostly diffuse [6]. Thus, our work is particularly targeted at scenes with strong inter-reflections.

2.2 Binary Pattern Structured Light

Coded structured light systems project illumination patterns onto the scene while assuming the scene is mostly diffuse and the effects of inter-reflection can be ignored. The patterns are used to generate a correspondence between one or more projectors and cameras. The coding strategies can be classified as temporal coding, spatial coding, and direct coding [22]. From among these, temporal time-multiplexed 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) 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 multiple gray level coding methods.

Accurately classifying pixels located within the black-and-white stripes is a crucial step for both diffuse and non-diffuse scenes. 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 [30] presented a way to threshold the images by using an adaptive threshold for each 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 in [25]. 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. In this article, we propose a method that produces better classification in such scenarios and also recognizes when a pixel cannot be robustly classified.

As a side note, some previous works achieve higher accuracy by using different exposure times [25] or multiple intensity illumination images [27]. Our improvement to the pixel classification process can also be used with such methods.

2.3 Adaptive Structured Light System

Our approach iteratively adapts the pattern images according to the partially reconstructed scene geometry and scene reflectance properties. There are related efforts which also aim to take the environment into account when designing code patterns. Caspi et al. [2] was among the first methods to explicitly model the transformation from projected color to observed color of the camera. Their adaptive n-color codes can achieve the same accuracy as binary Gray codes but using fewer patterns. Koninckx et al. [11] introduced an adaptive algorithm in which the pattern intensity is adjusted to avoid over- and under-exposure based on a calibrated camera-projector chain. The geometry of the patterns is also adjusted to avoid aliasing caused by the foreshortened patterns being projected on the scene. Although these methods solved some of the recognized problems in time-multiplexing structured light (e.g. large number of patterns, specularities, and aliasing), they assumed that the impact of global light phenomena is small. In contrast, our method explicitly deals with the problem of strong inter-reflection during structured-light acquisition. Moreover, since we are using more robust binary codes, only cameras need to be calibrated; projector calibration is not strictly necessary.

Adaptive structured illumination can also be applied to estimate the full light transport matrix from a projector to a camera [24]. This matrix encodes the relation between each camera pixel and each projector pixel; it can be estimated by turning on one projector pixel at a time. In [24] an adaptive subdivision scheme is proposed to speed up computing the light transport matrix. Although the full matrix is computed, there is no knowledge about the direct correspondence between camera and projector rays --

thus, no geometry is reconstructed. The focus of our paper is to create detailed and accurate 3D geometric models for complex scenes with strong inter-reflections.

2.4 Direct and Indirect Illumination Components

Finally, to acquire scenes with strong inter-reflections we build upon methods for decomposing the intensity of a pixel into the direct component and indirect (or global) component. The direct component is due to a single reflection. The indirect component is due to multiple reflections (e.g., inter-reflection, refraction, and subsurface scattering). Seitz et al. [23] proposed an inverse light transport theory to estimate the inter-reflection component for Lambertian surfaces. This method requires a very large number of images to compute matrices used in an inter-reflection cancellation process.

More recently, Nayar et al. [20] 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. In theory, a high frequency pattern and its inverse are enough to do the separation. In practice, more pattern images, such as shifting chessboard patterns, are used to compensate for the low resolution of the projector. As pointed out by the authors, the high frequency images among the structured-light patterns can also be used to do the separation. Therefore, our pixel classification method precludes the need for additional capturing and thus can be applied to previously acquired datasets.

To actually classify a pixel as on or off, we need to know the direct and indirect component under the illumination of structured-light patterns. Although the direct component is independent of the pattern shape, the indirect component is difficult to compute. Instead of seeking these values explicitly, our method attempts to establish bounds of the indirect and direct component and uses the bounds for classifying pixels as on or off.

3 Robust Pixel Classification

We first present our robust pixel classification algorithm for arbitrary binary patterns. Pixels that cannot be classified will be resolved via our adaptive process described in the next section. A structured-light method 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 it is not. Pixel classification methods generally establish the lower and upper bounds of the two intervals (either explicitly or implicitly). Then, if intensity p is within one interval but not in the other, the pixel belongs to that category. Otherwise, the pixel is labeled as uncertain.

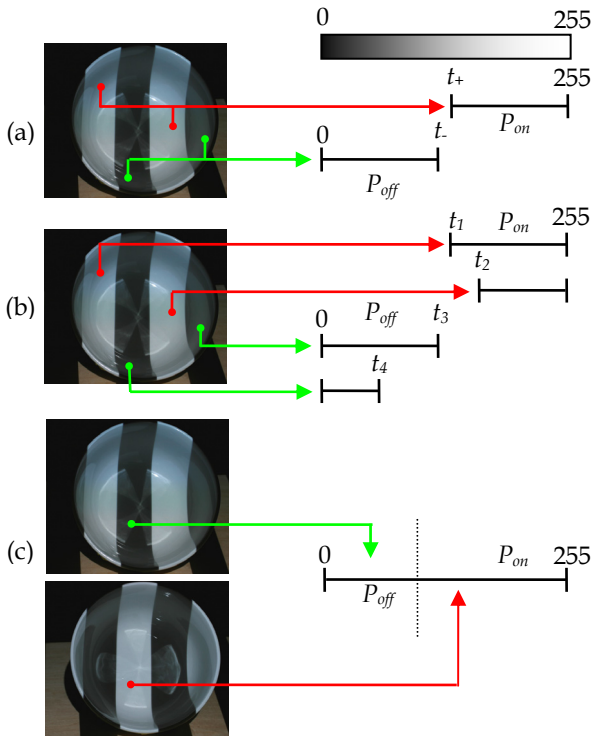


Fig. 3. **Intensity Intervals**. a) A simple method classifies pixel using two user-defined thresholds. b) An adaptive method uses a fixed but different threshold for each pixel. Each of the four pixels has a different threshold. c) A more expensive method classifies a pixel according to whether the pixel or its inverse is brighter.

For example, 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 Fig. 3a. A more accurate method uses a fixed but different value for each pixel as the classification threshold t [30]. Each of the four pixels in Fig. 3b has a different binary threshold t_i and thus has different P_{on} and P_{off} intervals. The threshold 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 assume the two intervals are non-overlapping, i.e. the lower bound of P_{on} is larger than the upper bound of P_{off} [25]. In this case, a single comparison between the pixel and its inverse decides which interval the pixel falls into without explicitly computing t (Fig. 3c).

These methods assume that the two intervals are non-overlapping. However, this is not 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 and robustly. Furthermore, our algorithm can robustly reject pixels that are invisible from the projector or problematic due to excessively strong inter-reflection. This is important because incorrect classification leads to inaccurate decoding and then to bad reconstruction.

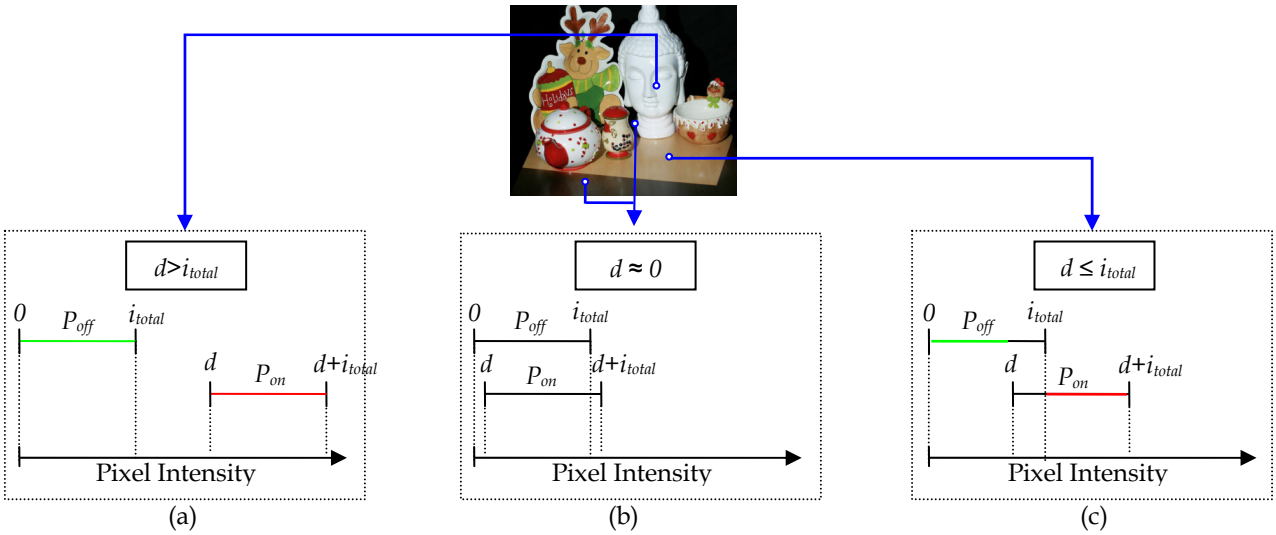


Fig. 4. **Pixel Classification Scenarios**. a) The intervals are completely separated when $d > i_{total}$. b) The two intervals are indistinguishable when $d \approx 0$ (shadow and surface outside of the projector's view frustum). c) The two intervals overlap when $d \leq i_{total}$.

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. In contrast, 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 calculate. For an indirectly illuminated pixel, its intensity p only contains the indirect component i_{off}. In summary,

$$p = d + i_{on} \quad \text{if pixel is on} \tag{1}$$
$$p = i_{off} \quad \text{if pixel is off} \tag{2}$$

Since the direct component d of an illuminated pixel is invariant to the illumination pattern, we can compute d for each pixel using the separation method introduced by Nayar et al. [20]. 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 turned on. As intensity values, they are also larger than or equal to zero. Thus,

$$i_{on} \in [0, i_{total}] \tag{3}$$
$$i_{off} \in [0, i_{total}] \tag{4}$$

From (1), (2), (3) and (4), we establish the lower and upper bounds for intervals P_{on} and P_{off}:

$$P_{on} \subseteq [d, d + i_{total}] \tag{5}$$
$$P_{off} \subseteq [0, i_{total}] \tag{6}$$

As shown in Fig. 4a, 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:

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

The two intervals are very similar to each other when d is close to zero (as in Fig. 4b). 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 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.

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

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 Fig. 4c. 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 *uncertain*
$p < \min(d, i_{total}) \rightarrow$ pixel is *off*
$p > \max(d, i_{total}) \rightarrow$ pixel is *on*
otherwise $\rightarrow$ pixel is *uncertain*

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 $\overline{p}$, 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 *uncertain*
$d > i_{total}$ & $p > \overline{p} \rightarrow$ pixel is *on*
$d > i_{total}$ & $p < \overline{p} \rightarrow$ pixel is *off*
$p < d$ & $\overline{p} > i_{total} \rightarrow$ pixel is *off*
$p > i_{total}$ & $\overline{p} < d \rightarrow$ pixel is *on*
otherwise $\rightarrow$ pixel is *uncertain*

It is worth noting that when $d > i_{total}$, the two intervals are separated. Hence, the mapping from a pixel and its inverse to the two intervals is one-to-one. Thus, 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. [25]).

3.4 Tight Intervals under a Projector

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 finding a larger lower bound of P_{on} or a smaller upper bound of P_{off}. However, given the limitation of not knowing the scene geometry *a priori*, these bounds are already tight.

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, when a scene is under the illumination of a subset of pixels from a fully lit projector, the presented lower bound of P_{on} and upper bound of P_{off} are already tight.

4. ADAPTIVE STRUCTURED LIGHT ALGORITHM

Our adaptive algorithm exploits the fact that the size of the ambiguous region can be reduced when the scene is illuminated by binary patterns. In general, shrinking the size of the ambiguous region requires either increasing the lower bound of P_{on} or decreasing the upper bound of P_{off}. On the one hand, projecting higher illumination intensity for the "on" bits will increase the lower bound of P_{on} but the indirect component will increase as well. If the higher intensity is still from the same projector, all intensities will be scaled by a constant factor so that no new information is gained. If multiple calibrated projectors are used to increase the direct illumination, due to the different configurations between the scene and different projectors, it is uncertain whether the ambiguity will decrease. On the other hand, decreasing the upper bound of P_{off}, i.e. the total indirect component i_{total}, can be achieved by turning off part of the projector pixels. Iterations can be performed to adaptively disable projector pixels and obtain an efficient acquisition.

4.1 Iterative Reflection Peeling

To improve reconstruction, our method decreases the upper bound of P_{off} (i_{total}) while keeping the lower bound of P_{on} (d) unchanged for ambiguous pixels. Our observation is that a pixel's indirect component decreases monotonically to the number of enabled projector pixels regardless of whether the pixel is directly illuminated or not. Consider the following scenario: the total indirect component of camera pixel p under a fully lit projector is defined as i_{total}; if any one projector pixel is turned off, the new indirect component i' of pixel p will decrease if the disabled projector pixel contributed to pixel p's indirect component or will remain unchanged otherwise. There-

fore:

$$i' \leq i_{total} \quad (7)$$

Disable more projector pixels will make the indirect component decrease monotonically. This gives us the ability to reduce ambiguous pixels' indirect components by simply turning off those projector pixels whose codes have already been observed by the cameras, since they are not useful anyway.

Our *iterative reflection peeling* algorithm starts from an all-white projector mask image I_0 (Fig. 5). First we estimate the per-pixel direct component d and total indirect component i_{total} under the illumination of I_0 using [20]. Then, Gray code patterns and their inverses are projected onto the scene. Our robust pixel classification algorithm, described in the previous section, is used to decode the pattern images. Then, we identify the projector pixels, which correspond to the codes that have already been recovered, and disable them. This produces a new projector mask image I_1 with a reduced number of white pixels as compared to I_0. We perform an AND operation between I_1 and each of the Gray code patterns to obtain a set of new patterns to be projected in the next iteration.

To estimate the tight intensity value bounds for P_{off} and P_{on} under the newly generated structured-light patterns, we first compute the per-pixel direct component d and indirect component i_1 under the illumination of projector mask image I_1. The direct component is again invariant under different patterns as long as the scene point is directly illuminated. Therefore, the intensity p of a pixel under the illumination of I_1 can take on either one of the following two values:

$$p = d + i_1 \quad \text{if directly illuminated by } I_1 \quad (8)$$
$$p = i_1 \quad \text{if not directly illuminated by } I_1 \quad (9)$$

The camera pixels not directly illuminated by I_1 are not important because they either have already been assigned a correct codeword in a previous iteration or are in shadow and cannot be reconstructed at all. Thus, the per-pixel indirect component i_1 can be computed by simply subtracting the total direct image (per-pixel d) from the image captured under the illumination of I_1. We assume the camera intensity response is linear. Since our algorithm estimates the intensity range for pixel classification instead of computing the exact camera intensity, a small error can be handled by conservatively adjusting the boundaries. The resulting image of the subtraction has the value i_1 for the unresolved camera pixels. Then, the new i_1 ($\leq i_{total}$) is the upper bound of P_{off}. More pixels are assigned a correct code due to the fact that the ambiguous region is reduced by *peeling* away part of the global inter-reflection. Afterwards, a new projector mask image I_2 is generated and the iterations continue until convergence. The pipeline of the iterative reflection peeling algorithm is illustrated in Fig. 5.

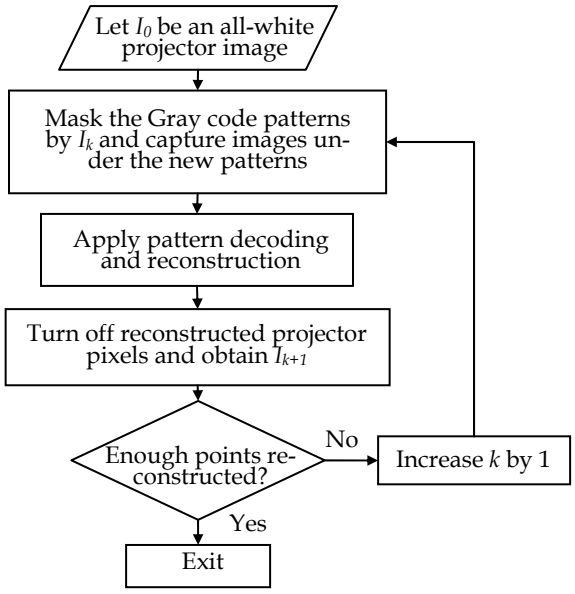


Fig.5. **Iterative Reflection Peeling Algorithm**.

The algorithm converges when almost all the codes are observed and the reconstruction is almost complete. Nevertheless, it can also happen when the indirect component is too strong everywhere for all the unresolved pixels such that no additional pixels can be decoded. Thus, the divide-and-conquer method described in following section is used to tackle this problem.

4.2 Divide-and-Conquer Scheme

It has already been shown that a projector image with less enabled pixels results in decreased indirect component for every camera pixel. This principle can also be applied when the indirect component is strong for all the remaining unresolved camera pixels. The key idea is to subdivide the projector image I_k into n smaller patches and perform the iterative reflection peeling algorithm on each patch separately. Since the indirect component for every camera pixel is decreased, the ambiguous region is reduced allowing more decoding and reconstruction. If necessary, each patch can be subdivided further in order to obtain a complete 3D model. The choice of subdivision logic is less important since the indirect component of every camera pixel is guaranteed to decrease or remain unchanged after subdivision. We choose a quadtree recursive subdivision (i.e. subdividing regions into four rectangles) due to its suitability for covering the entire 2D projector image space.

A naïve subdivision would sequentially execute the reflection peeling algorithm using each one of the subdivided projector patches. However, this might be redundant. Consider the following case. Each of a set of patches (e.g., 1, 2, 3, and 4) illuminates a group of camera pixels, either directly or indirectly. If the set of illuminated camera pixels of projector patch 1 does not overlap with those of patch 2, patch 1 and 2 should be combined and the reflection peeling algorithm should be performed only once using projector patch 1 and 2 together as one input mask. We call them non-conflicting projector patches. The fact that rays from one projector patch do not contribute to the indirect components of camera pixels corresponding to the other projector patch enables us to capture less images for a reconstruction.

4.3 Capture Acceleration

We exploit the parallelism due to non-conflicting projector patches by using an adaptive subdivision algorithm (Fig. 6). Similar to that in the work of [24], the goal of adaptive subdivision is to reduce total capturing time. However, the overall objective of [24] is to identify the set of camera pixels illuminated by each projector pixel (either *directly* or *indirectly*), while our algorithm tries to find the set of camera pixels that are *directly* illuminated by one projector pixel with the help of time-multiplexed coding. In general, our algorithm requires much less projection-capture cycles than that of [24] and will produce a 3D model.

4.3.1 Adaptive Subdivision

For clarity, we discuss the algorithm using the example illustrated in Fig. 6. The algorithm starts with a single projector patch (Fig. 6a). The projector mask (Fig. 6b) consisting of only one patch is used to start the first iterative reflection peeling. After it converges, the projector patch (Fig. 6a) is uniformly subdivided into four patches (Fig. 6c, patch 1-4). If any of the projector patches is almost black (i.e. the number of un-resolved projector pixels is less than a small threshold), we discard that patch. Then, we project the patches (patch 1-4 in Fig. 6c) sequentially onto the scene and capture images under the illumination of them. We discard any projector patch that produces a dark camera image, since it projects to a part of the scene not visible to the camera (patch 2 in this example). Then, to maximize parallelism, we seek to group the projector patches that are non-conflicting. If any two captured images have non-overlapping sets of pixels that are illuminated either directly or indirectly, the corresponding projector patches are non-conflicting (patch 1 and 4 in this example). Therefore, we can group the patches into one mask. In this example, after the subdivision, we obtain two masks (Fig. 6d): one consists of only one patch (3) and the other consists of two patches (1, 4). Each mask will be used to start a new iterative reflection peeling step.

When these two iterative algorithms converge, we need to subdivide the projector patches further, resulting in 12 smaller projector patches (Fig. 6e patch 1-12). From the knowledge of the upper level, for example, we know patch 1 and 9 are non-conflicting. They can already be grouped together. Therefore, we only need to project and capture eight images to identify non-conflicting projector patches: (1, 9), (2, 10), (3, 11), (4, 12), (5), (6), (7), and (8). By comparing the captured images, these sets can be further merged. For example, (1, 9), (4, 12) and (8) are black projector patches, (2, 10) and (5) do not conflict, and (6) results in a black camera image. Thus, only three masks are needed in this example: (7), (2, 5, 10), and (3, 11) (Fig. 6f) and three new iterative peeling procedures are carried out using the three masks as input. The divide-and-conquer continues until all iterations converge or the projector patches are too small.

4.3.2 Direct Illumination-Based Adaptation

Since we are only interested in *directly* illuminated camera pixels for correspondence purpose, we can easily discard projector patches that only generate indirect illumination on the scene. This can be done by subtracting the total direct image from the captured images under the illumination of each projector patch P. Given the direct component d and the indirect component i_p, the intensity p of a pixel under the illumination of projector patch P can take on either one of the following two values:

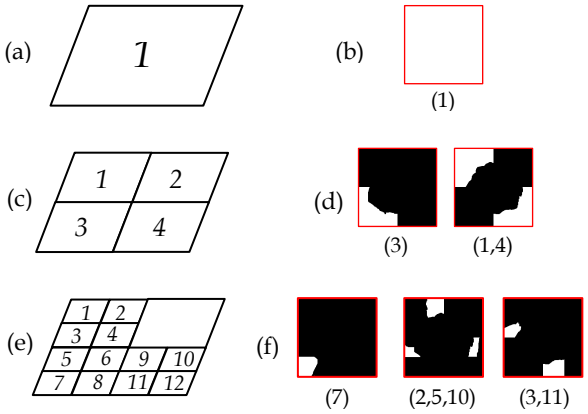


Fig.6. **Adaptive Subdivision**. Subdivision scheme for three levels (left) and corresponding binary images that are used to mask the Gray code patterns for each level (right).

$$p = d + i_p \quad \text{if directly illuminated by } P \tag{10}$$
$$p = i_p \quad \text{if not directly illuminated by } P \tag{11}$$

After subtracting the total direct intensity d from each pixel, we obtain the difference intensity p':

$$p' = i_p \quad \text{if directly illuminated by } P \tag{12}$$
$$p' = i_p - d \quad \text{if not directly illuminated by } P \tag{13}$$

If p' is negative, the pixel is not directly illuminated by the patch since i_p is non-negative. If every pixel of the difference image is negative, the projector patch does not produce any direct illumination on the scene. Thus, the patch can be safely discarded from further processing. In practice, we disable a patch by testing whether a difference image is too dark (i.e., below a small intensity threshold).

By subtracting the total direct component, the intensity of the camera pixels that are already decoded will become negative since there is no direct light coming to that pixel. Therefore, these pixels can be easily detected and will be ignored when computing non-conflicting patches. By discarding projector pixels that only produce indirect light and removing the impact of already decoded camera pixels during non-conflicting set test, our method significantly speeds up the original adaptive subdivision.

5. Reconstruction and Rendering

To show the results of our adaptive algorithm, we implemented a reconstruction and rendering engine, which uses two mutually calibrated digital cameras and an uncalibrated digital projector. During each iteration, the decoding process concatenates the bits from all our binary

classification images, ignoring any pixel with uncertain bits. Then pixels seeing the same codes from two cameras are corresponded and triangulated.

Establishing correspondences implies identifying common surface points observed by both cameras. The decoding procedure 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. The decoding results per camera are stored in two camera-resolution size code maps: one for each of the XY coordinates. A pixel value x in the X code map means that the camera pixel sees the projector pixel whose X coordinate is x. Similarly occurs for the Y code maps. In order to improve reconstruction, we apply standard clean up algorithms as suggested in [25]. We first fill in small holes of the code map and interpolate the code values into these regions to enforce smoothness. We then remove small isolated clusters in the code maps to reject outliers.

Although the major decoding process is the same for all iterations, the later iterations can take advantage of the information gathered in the earlier ones. For example, on the one hand, if a camera pixel is successfully decoded and reconstructed in an earlier iteration, it should no longer be considered as a candidate for the remainder of the process. On the other hand, if a projector pixel has been assigned to a reconstructed 3D point, any camera pixel seeing its codeword should be discarded because the projector pixel is already disabled. Thus, outliers are easily detected and rejected.

We group the pixels seeing the same codeword and use the center as the overall position. To ignore misclassifications, an 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, renderings are produced by a projective texture mapping of the reconstructed mesh. The meshing is performed in the camera image plane using 2D Delaunay triangulation. Although the texture contains both diffuse and directional reflections, we use it to shade the scene as it would be seen from the camera's viewpoint. To further separate the diffuse and specular components in the texture images, we could incorporate a polarization based method (e.g. [17][31]), a color-space method (e.g. [14]), or a high-frequency structured-light method (e.g., [13]).

6. 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 used at 1024 by 1024 pixel resolution (a subset of the native resolution of 1400x1050 pixels). During a capturing session, shifting chessboard patterns are initially projected to separate the direct and indirect components for each pixel. The pattern is composed of 5 by 5 small blocks and is shifted 1 pixel a time and 9 times along each of the X and Y directions. The exposure is kept low in order to avoid overexposure (e.g., due to highlights or caustics) and to limit the signal from dark projector pixels. Thus, the ambient term is very small and can be safely ignored in both separation and classification.

During each iteration, 16 binary Gray code patterns and their inverses are projected onto the scene, resulting in 32 images per camera. Of these patterns, 8 are horizontal stripe patterns and the remaining 8 are vertical stripe patterns. These patterns are the AND result between the original Gray code patterns and the current projector mask. When a new projector mask is computed, we apply a mild morphological dilation operator to the projector image in order to ensure the borders of the stripes are captured by the cameras. Whenever a subdivision is necessary, we also capture up to 4^l (l is the level of subdivision) images to perform quadtree subdivision. In practice, due to parallelism, the number of captured images is much smaller than 4^l when $l > 1$ as explained in Section 4.3.

All software is implemented on a Dell PC with 3.2GHz CPU and 2GB memory. Separation on a scene takes about 120 seconds for each camera. Classifying all images from each camera takes about 60 seconds. The lower bounds and upper bounds derived in Section 3 and 4 are for ideal scenarios. In practice, due to the light leaking from the deactivated projector pixels, "fogging" inside the projector that adds light to the patterns, and projector and camera non-linearities, it is not exact. To compensate, we use a small ε to conservatively reject pixels close to the interval boundary. The same ε is also used in standard methods to improve reliability.

TABLE 3. Datasets Statistics

Name	*bowl*	*dinnerware*	*objects1*	*objects 2*	*lady*
# of images captured (standard method needs 64)	682	770	1650	1698	482
Total image acquisition time (min)	55	60	145	146	40
Total modeling time (min)	34	38	90	94	22
# of points using standard classification	7463	25990	25239	29721	58672
# of points using our classification (1 iteration)	9050	28555	33712	33553	57833
# all points reconstructed using our system	14010	30292	38395	37398	60677
# of iterations	8	9	22	23	5
# of subdivisions	1	2	3	3	0

7. RESULTS

We have applied our approach to the capture of five example scenes, ranging from a single concave object to complex scenes with several objects. Table 3 summarizes the statistics of our datasets. The first dataset is a single *bowl*. The *dinnerware* dataset consists of three mugs, one bowl and two Vitrelle™ glass plates. The *objects 1* dataset has two plastic buckets and three porcelain ornaments. The *objects 2* dataset is formed of several shiny objects which generate a lot of inter-reflection. Finally, the *lady* dataset is a case containing strong diffuse inter-reflection. We list the total number of photographs captured, the amount of time spent capturing images and reconstructing the models, the number of iterations and adaptive subdivisions to capture the final model, and the number of points reconstructed using standard classification (the method that compares pattern image and its inverse as in Fig. 3c), our classification for only one iteration, and our classification for all iterations. The number of points reported is obtained by adjusting global culling thresholds until there are no visual outliers. Since our cameras have much higher resolution than the projector, the number of reconstructed points is much less than the number of camera pixels. For *bowl* and *dinnerware*, only areas containing the objects are reconstructed, additional points belonging to walls and tables are discarded. As shown in the table, a significant amount of time was spent capturing images and transferring them from our cameras to the computer using USB cables. In addition, our high-resolution 3CCD camera produces high quality imagery but increases processing time because of the greater number of pixels. Using a lower resolution camera can reduce both acquisition and processing time but at the expense of lower image quality.

The first dataset is a single concave and white porcelain bowl. Fig. 7 shows pictorially the classification results of the bowl. As one can see, the actual images (Fig. 7a) are difficult to classify due to strong inter-reflection. The standard method produces a lot of false positives (Fig. 7b). In contrast, our method avoids misclassifications by declaring these difficult pixels as uncertain (Fig. 7c). The unclassified pixels are resolved during later iterations of our algorithm (Fig. 7d). In the presence of highlights and caustics, the smoothness constraint assumed by the separation method does not hold [20]. Therefore, the estimated direct and indirect components are inaccurate. In this case, our method does generate false labeling for the pixels belonging to such regions (circular structure in Fig. 7d). However, we found that the post-processing of the correspondence data will easily reject these outliers. Our final classification is very close to the ground truth (Fig. 7e). The ground truth results are generated by manually classifying the pixels in each pattern image.

To understand the behavior better, we graph the number of correctly classified pixels and incorrectly classified pixels for both standard method and our method. A pixel is correctly classified if its on/off decision is the same as that of the ground truth. A pixel is incorrectly classified if

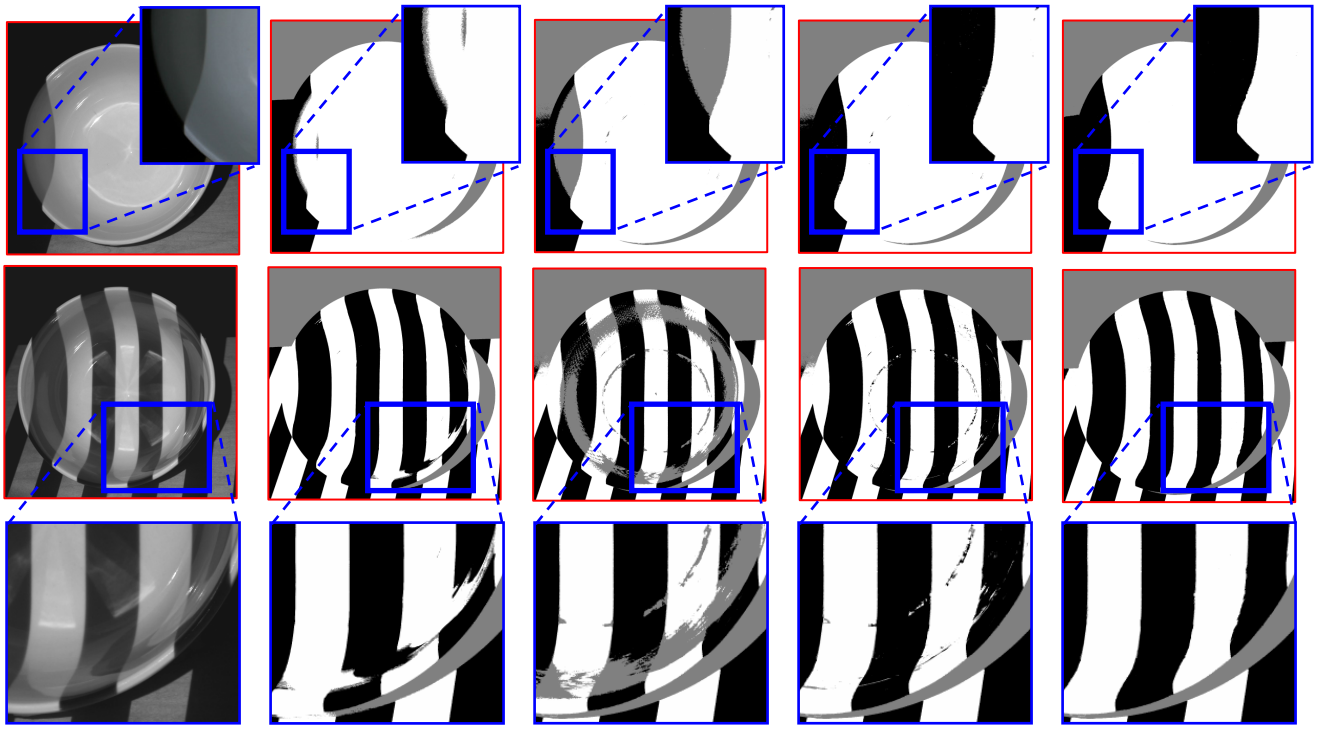


Fig.7. **Classification Results.** a) The first two rows are part of the actual pictures of the bowl under the illumination of two patterns. The third row shows a zoom-in view on a rectangular region of the second pattern. b) The classification results using standard method that uses a pattern and its inverse. A white (black) pixel means on (off). c) Classification results using our robust method for one iteration. A gray pixel means uncertain. d) Classification results after applying our algorithm for all iterations. e) Manually labeled ground truth.

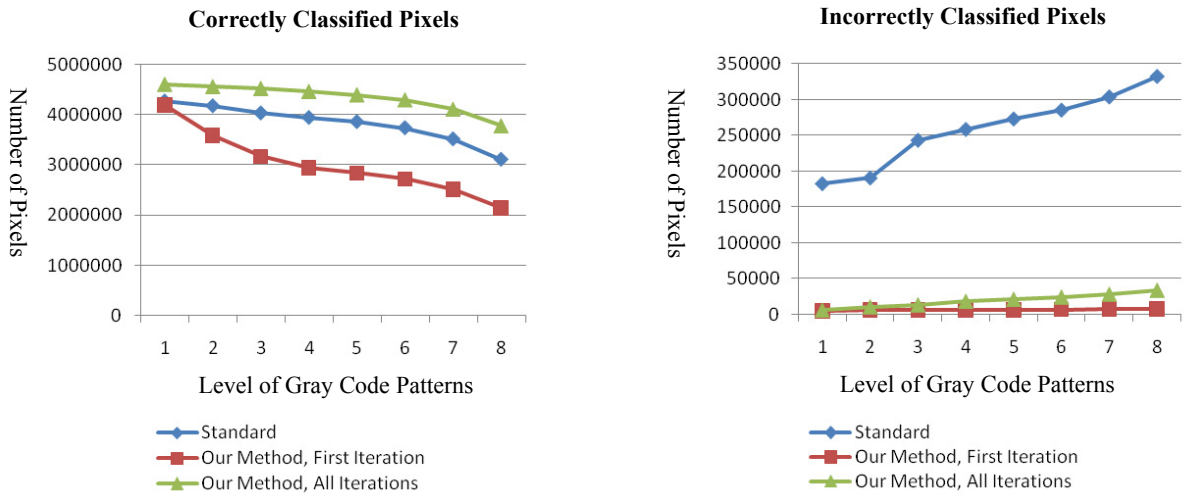


Fig.8. **Number of Correctly and Incorrectly Classified Pixels**. The horizontal axis is the level of Gray code subdivision; higher values correspond to more stripes in the projected pattern. The vertical axis represents the number of correctly/incorrectly classified pixels. As compared to the standard method, our method produces both more correctly classified pixels and less incorrectly classified pixels.

its on/off decision is opposite to that of the ground truth. Fig. 8 shows the two graphs. When computing the number of correctly classified pixels for level l, we only count those that are correctly classified for *all* levels up to l. For incorrectly classified pixels, we count those that are incorrectly classified for *any* level up to l. As can be seen, although standard classification can produce 45% more correct classifications than one step of our method, it also generates 41 times more incorrectly classified pixels. During reconstruction, these false positives cause correctly classified pixels to be culled away. By applying our robust classification for all the iterations we obtained 21% more correct classifications and 10 times less incorrect ones. Thus, the overall modeling quality is much higher. The numbers reported for all iterations includes those of the first iteration and all the later iterations.

Fig. 9 shows the pictures of the results for the *bowl*. The quantitative improvement over standard method is 21% for the first iteration and 88% after 8 iterations. The reconstructed point cloud using standard method is irregular (Fig. 9a). In contrast, our method initially avoids dealing with ambiguous regions and reconstructs the non-ambiguous areas fairly well (Fig. 9b). The final reconstruction after several iterations of our method is close to a complete model (Fig. 9c). The remaining small holes are due to difficult configurations such as caustics, pixel overflow due to specular highlights, and surface patches that are nearly parallel to the viewing or lighting directions.

Since we have manually labeled the pixel-code maps for this dataset, we can apply the same reconstruction engine to produce a ground truth point cloud. Then, points reconstructed using ground truth classification, standard classification, and our classification can all be corresponded through their projector pixel coordinates; this allows us to easily compute the distance between the ground truth model and a reconstructed model without aligning models and/or finding the closest points between the models. To evaluate the *accuracy* and *completeness* of our approach, we use the same metric as in Seitz et al. [28]. We first compute the per-point distance between the ground truth model (G) and the model to be evaluated (R). To evaluate accuracy, we compute the distance d, for which, X% of points of R are within distance d to ground truth G. A more accurate reconstruction tends to have a smaller d value. Fig. 10a shows the results for different X values (plotted along the horizontal axis) using standard method and our method. As can be seen, our method consistently produces more accurate reconstructions than standard method. To measure completeness, we compute the fraction of points of ground truth G that are within a certain distance d to the model R. Fig. 10b shows the results for different d values (plotted along the horizontal axis). Our method produces about 30% more points than standard method. The completeness score for our method is close to 100% when the distance d is sufficiently large. The base of the percentage is different in the two graphs; therefore, the d values are different for the

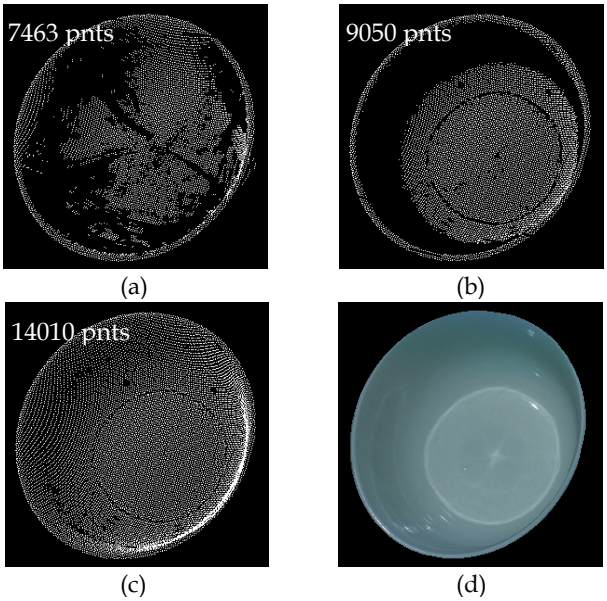


Fig.9. **Modeling the Bowl.** Point cloud reconstructed using a) standard method, b) our robust classification for one iteration, and c) our method for all iterations. d) The final model is rendered from a novel viewpoint using texture mapping.

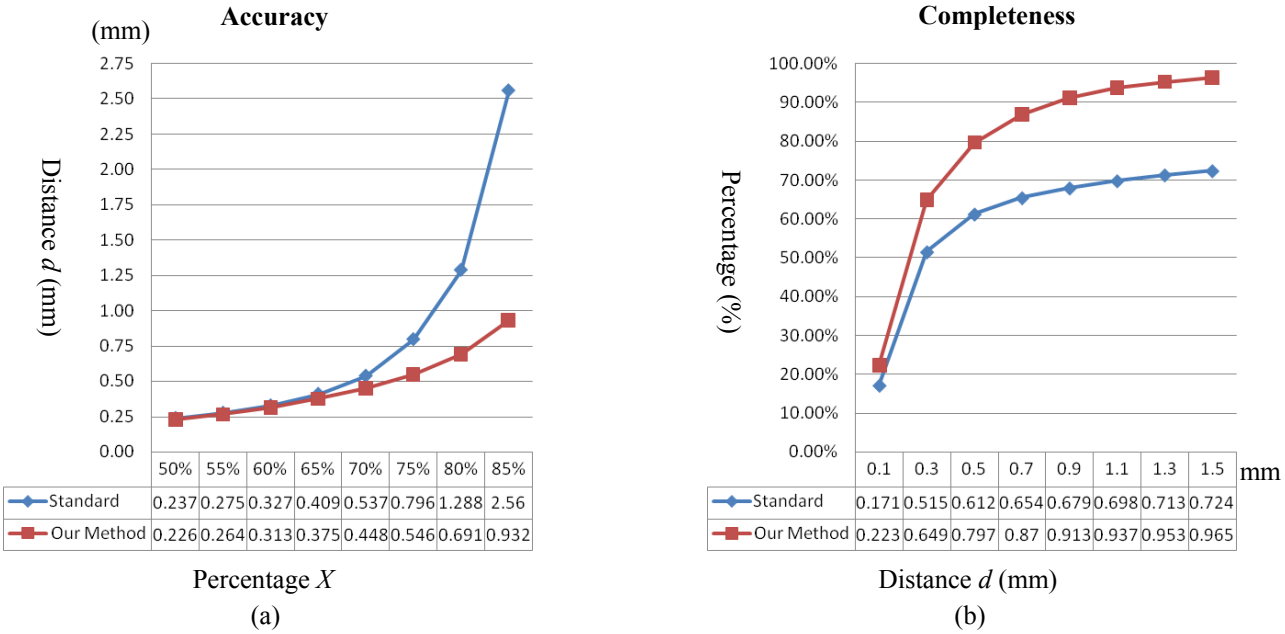


Fig.10. **Reconstruction Accuracy and Completeness.** a) **Accuracy**: The distance *d* (Y axis) when *X*% (X axis) of points of a reconstructed model R (using either standard method or our method) is within distance *d* to the ground truth model G. A more accurate reconstruction tends to have smaller *d* value. b) **Completeness**: The percentage (Y axis) of points of the ground truth model G is within distance *d* (X axis) to the reconstructed model R. A more accurate and complete reconstruction has a higher completeness score.

same X value. We avoided very difficult regions during manual classification for creating the ground truth model. This leads to the fact that the ground truth model has fewer points than the model reconstructed by our method. Hence, the *d* value for the second graph may be smaller even when the percentage number is the same.

To evaluate the contributions from different components of our iterative algorithm (i.e. initial pixel classification, iterative reflection peeling, and adaptive subdivision), we compute the accuracy and completeness score of each step for the bowl dataset. For computing accuracy score, we set the *X*% value to be 90%. For completeness score, we set the distance *d* to be 1.0mm. The results are listed in Table 4. The accuracy of our method decreases when using more iterations because the later iterations are eventually dealing with more optically-difficult regions of the scenes; thus are less robust. However, the overall error is still rather small. Our reflection peeling algorithm (step 2) completes the object from 60% to 90%. The gain of adaptive subdivision (step 3 and 4) on this particular dataset is relatively small; in general, it is very dependent on scene geometry and reflectance.

To show the effects of outlier rejection (image- and world-space culling discussed earlier in Section 5), we show synthetically-shaded models without and with outlier rejection using standard method and our method in Fig. 11. The original reconstruction using standard method (Fig 11a) is noisier than that using our method (Fig 11c). After outlier rejection, standard method only reconstructs an incomplete model (Fig 11b). In contrast, our method produces both improved correctness and improved completeness (Fig 11d).

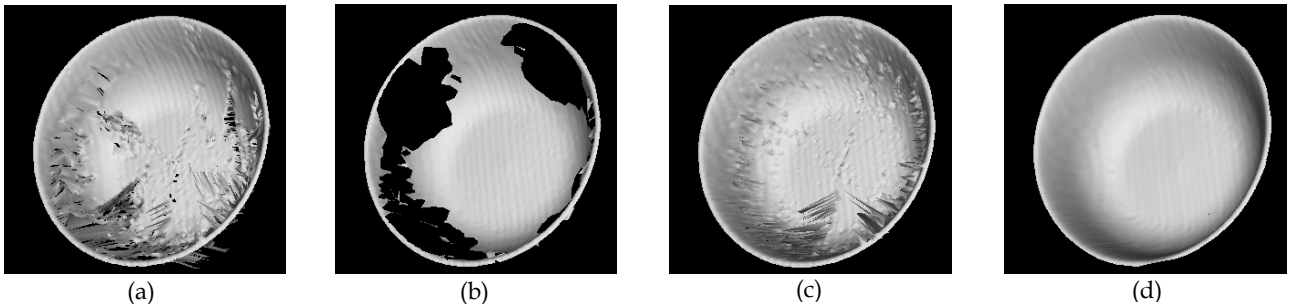
TABLE 4. CONTRIBUTIONS OF DIFFERENT COMPONENTS

Step	1	2	3	4
Accuracy (mm)	0.367	0.710	0.711	0.723
Completeness	60%	90%	92%	95%

Legend:
1 = our pixel classification before reflection peeling
2 = reflection peeling converged before subdivision
3 = after subdivision and reflection peeling only on upper left quarter of projector mask
4 = full algorithm done

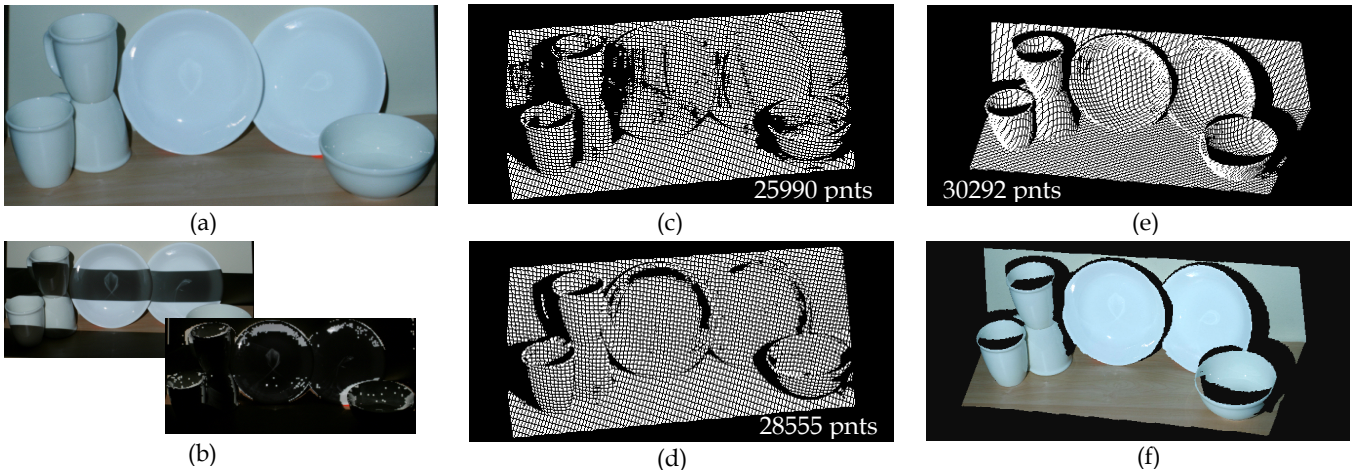
For the more complicated *dinnerware*, *objects 1*, and *objects 2* datasets, our method also performs better than standard method. The improvement of the first step ranges from 10% to 34%. After all iterations, the improvement is 17%-88%. Fig. 12 and Fig. 13(a-d) show the reconstruction results for *dinnerware* and *objects 1*. The results for *objects 2* were shown in Fig. 1. With 8-23 iterations and at most 3 levels of subdivision, our algorithm can reconstruct quite complete models for various types of scenes. It is interesting to note that for a subdivision level 3, the projector image is divided into 8x8 small patches. However, there are only about a little more than 20 iterations in total. This is because parallelism is exploited by efficiently combining non-conflicting projector patches together.

The *lady* statue is white and diffuse; thus a perfect subject for standard structured light. However, when it is placed in a corner, diffuse inter-reflection is prevailing. Although for the entire scene, standard method produces slightly more points than our method with only one iteration (58672 vs. 57833), our method generates more points on the lady statue (8900 vs. 9066). After 5 iterations, our points reconstruct a nearly complete model of 60677 points (9413 points on the statue). Fig. 13(e-h) shows the results.



(a) (b) (c) (d)

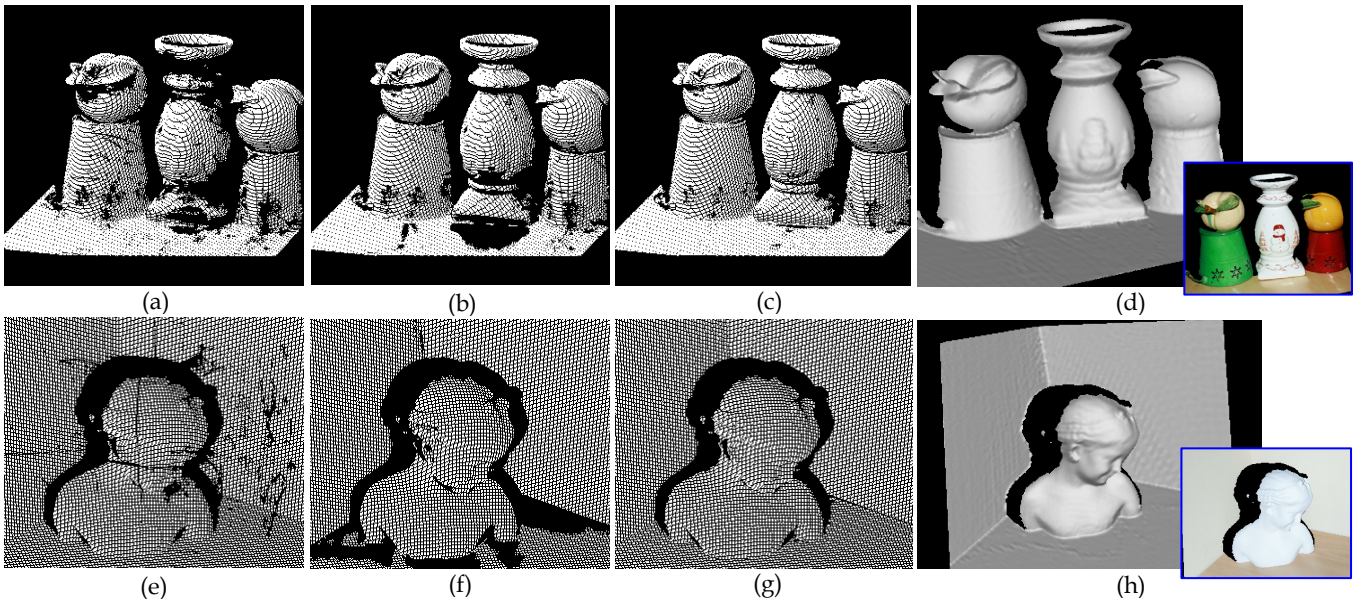
Fig. 11. **Outlier Rejection.** Synthetically-shaded model reconstructed using standard method before outlier rejection (a) and afterwards (b). Synthetically-shaded model reconstructed using our method before outlier rejection (c) and afterwards (d).



(a) (c) (e)

(b) (d) (f)

Fig. 12. **Dinnerware Set.** a) A picture of the scene. b) The same scene under the illumination of a structured-light pattern during first iteration (left) and second iteration (right). c) 3D point cloud reconstructed using standard pixel classification. d) 3D point could using our pixel classification for one step. e) Complete point cloud after 9 iterations. e) Rendering the scene as a texture mapped triangular mesh.



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 13. **Modeling Scenes with Inter-reflection.** a, e) 3D point cloud using standard method. b, f) 3D point could using our one step robust pixel classification. c, g) Complete point cloud after 21 and 9 iterations, respectively. d, h) Rendering the scene as a synthetically-shaded triangular mesh from a novel viewpoint. Each bottom right insert shows texture mapped models of the scene.

8. Conclusions and Future Work

We have presented an adaptive and iterative algorithm for modeling scenes with strong inter-reflections. Our algorithm is based on establishing accurate pixel intensity intervals of a scene during the illumination of a structured-light pattern. By iteratively reducing the inter-reflection within the scene, our algorithm is able to robustly decode more camera pixels in successive iterations. Furthermore, the inter-reflection is reduced in an adaptive manner whereby parallelism is exploited in order to decrease total image capturing time. Our experiments show that as compared to a standard method, our algo-

rithm improves both the quantity and quality of the reconstructed 3D points. This produces dense datasets suitable, for instance, for content creation, virtual reality, and other applications. Our reflection peeling and subdivision approach can also be applied to standard pixel classification case. Standard method tends to produce non-uniform reconstructions due to large number of outliers (e.g. Fig. 9a); the remaining projector pixels distribute the entire projector image space. Our robust pixel classification reconstructs optically-simple regions completely in the earlier stages. Therefore, the projector patches corresponding to these regions can be discarded earlier in the pipeline. This leads to faster convergence than applying reflection peeling directly to standard method.

One limitation of our algorithm is being conservative. If the pixel intensity is in the ambiguous classification interval, our method classifies the pixel as uncertain and relies on later iterations to decode it. For an object with strong subsurface scattering, such as fluffy toys and candles, its direct component can be very weak. Therefore, our algorithm will classify many pixels as uncertain while a standard method can produce at least a very coarse reconstruction.

We are investigating several avenues of future work. First, the current system uses a uniform quadtree subdivision. The remaining pixels on a projector image might be arbitrarily shaped, thus a non-uniform and non-regular subdivision scheme (e.g., an oriented-bounding-box hierarchy) would help produce a tighter-fitting spatial hierarchy and thus further reduce capture time. Second, our system currently ignores the fact that specularities and caustics violate the assumption of the direct/indirect illumination separation algorithm and thus no valid intensity intervals are defined in these areas. In the future, we would like to study how to compute the intensity intervals for these regions. Third, we would like to perform multi-viewpoint captures to which our method can be straightforwardly applied. Finally, we seek to extend our intensity interval idea to structured-light systems using multi-grayscale patterns, multi-color patterns, and multi-parameter imaging. This involves establishing bounds for each parameter instead of the current two (i.e. black and white).

ACKNOWLEDGMENTS

This work was supported by NSF CCF 0434398 and by a Purdue Research Foundation grant. The authors would also like to thank the reviewers for their comments and suggestions.

REFERENCES

[1] P. Belhumeur, D. Kriegman, and A. Yuille, "The Bas-Relief Ambiguity", *Intl. Journal. of Comp. Vision*, vol. 35, no. 1, pp. 33-44, 1999.

[2] D. Caspi, N. Kiryati, and J. Shamir, "Range Imaging with Adaptive Color Structured Light", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, no. 5, pp. 470 – 480, 1998.

[3] M. Chandraker, F. Kahl, and D. Kriegman, "Reflections on the Generalized Bas-relief Ambiguity", *IEEE Conf. on Comp. Vision and Pattern Recognition*, pp. 788-795, 2005.

[4] T. Chen, H. Lensch, C. Fuchs, and H.P. Seidel, "Polarization and Phase-Shifting for 3D Scanning of Translucent Objects," *Proc. of IEEE Comp. Vision and Pattern Recognition*, pp. 1-8, 2007.

[5] J. Clark, E. Trucco, and L. Wolff, "Using Light Polarization in Laser Scanning", *Image and Vision Computing*, vol. 15, no. 2, pp. 107-117, 1997.

[6] M. Cohen and D. Greenberg, "The Hemi-Cube: A Radiosity Solution for Complex Environments", *Proc. of SIGGRAPH*, pp. 31-40, 1985.

[7] B. Curless and M. Levoy, "Better Optical Triangulation Through Spacetime Analysis", *Proc. of Intl. Conf. on Comp. Vision*, pp. 987 - 994 , 1995.

[8] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface Reconstruction from Unorganized Points", *Proc. of ACM SIGGRAPH*, pp. 71-78, 1992.

[9] S. Inokuchi, K. Sato, and F. Matsuda, "Range Imaging System for 3-D Object Recognition", *Proc. Intl. Conf, of Pattern Recognition*, pp. 806-808, 1984.

[10] L. Kobbelt and M. Botsch, "A Survey of Point-based Techniques in Computer Graphics", *Computers and Graphics*, vol. 28, no. 6, pp. 801-814, 2004.

[11] T. Koninckx, P. Peers, P. Dutre, and L. Van Gool, "Scene-Adapted Structured Light", *Proc. IEEE Conf. on Comp. Vision and Pattern Recognition*, pp. 611- 618, 2005.

[12] K. Kutulakos and E. Steger, "A Theory of Refractive and Specular 3D Shape by Light-Path Triangulation", *Proc. of Intl. Conf. on Comp. Vision*, pp. 1448-1455, 2005.

[13] B. Lamond, P. Peers, and P. Debevec, "Fast Image-based Separation of Diffuse and Specular Reflections", *SIGGRAPH Sketch*, 2007.

[14] S. Mallick, T. Zickler, P. Belhumeur, and D. Kriegman, "Specularity Removal in Images and Videos: A PDE Approach," *Proc. of European Conf. of Comp. Vision*, pp. 550-563, 2006.

[15] W. Matusik, H. Pfister, R. Ziegler, A. Ngan, and L. McMillan, "Acquisition and Rendering of Transparent and Refractive Objects", *Proc. of the 13th Eurographics Workshop on Rendering*, pp. 267-278, 2002.

[16] D. Miyazaki and K. Ikeuchi. "Inverse Polarization Raytracing: Estimating Surface Shape of Transparent Objects", *Proc. of IEEE Comp. Vision and Pattern Recognition*, pp. 910-917, 2005.

[17] S. Nayar, X. Fang, and T. Boult, "Separation of Reflection Components Using Color and Polarization", *International Journal of Computer Vision*, vol. 21, no. 3, pp.163-186, 1997.

[18] S. Nayar, "Shape Recovery Using Physical Models of Reflection and Inter-reflection", *Ph.D Thesis*, Carnegie-Mellon University, 1991.

[19] S. Nayar, K. Ikeuchi, and T. Kanade, "Shape From Interreflections", *Proc. of Intl. Conf. on Comp. Vision*, pp. 2-11, 1990.

[20] S. Nayar, G. Krishnan, M. Grossberg, and R. Raskar, "Fast Separation of Direct and Global Components of a Scene using High Frequency Illumination," *Proc. ACM SIGGRAPH*, pp.935-944, 2006.

[21] J. Park and A. Kak, "3D Modeling of Optically Challenging Objects", *IEEE Trans. on Visualization and Comp. Graphics*, vol. 14, no. 2, pp 246-262, March-April 2008.

[22] J. Salvi, J. Pages, and J. Batlle, "Pattern Codification Strategies in Structured Light Systems", *Pattern Recognition*, vol. 37, pp, 827-849, 2004.

[23] S. Seitz, Y. Matsushita, and K. Kutulakos, "A Theory of Inverse Light Transport," *Proc. Intl. Conf. Computer Vision*, pp. 1440-1447, 2005.

[24] P. Sen, B. Chen, G. Garg, S. Marschner, M. Horowitz, M. Levoy and H. Lensch, "Dual Photography", *Proc. ACM SIGGRAPH*, pp. 745-755, 2005.

[25] D. Scharstein and R. Szeliski, "High-accuracy Stereo Depth Maps using Structured Light," *Proc. IEEE Conf. on Comp. Vision*

and Pattern Recognition, pp. 195-202, 2003.

[26] D. Scharstein and R. Szeliski, "A Taxonomy and Evaluation of Dense Two-frame Stereo Correspondence Algorithms," *Intl. J. Computer Vision*, vol. 47 no.1/2/3, pp. 7-42, Apl-Jun 2002.

[27] D. Skocaj and A. Leonardis, "Range Image Acquisition of Objects with Non-uniform Albedo using Structured Light Range Sensor," *Proc. Intl. Conf, of Pattern Recognition*, pp.778–781, 2000.

[28] S. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms", *Proc. of IEEE Conf. on Comp. Vision and Pattern Recognition, pp.* 519-526, 2006.

[29] M. Tarini, H. Lensch, M. Goesele, and H.P. Seidel, "3D Acquisition of Mirroring Objects using Striped Patterns", *Graphical Models*, vol. 67, no. 4, pp. 233-259, 2005.

[30] M. Trobina, "Error Model of a Coded-light Range Sensor," Technique Report, Communication Technology Laboratory, ETH Zentrum, Zurich, 1995.

[31] S. Umeyama and G. Godin, "Separation of Diffuse and Specular Components of Surface Reflection by Use of Polarization and Statistical Analysis of Images", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no. 5, pp.639-647, 2004.

[32] T. Wada, H. Ukida, and T. Matsuyama, "Shape from Shading with Interreflections Under a Proximal Light Source: Distortion-Free Copying of an Unfolded Book", *Intl. Journal of Comp. Vision*, vol. 24, no. 2, 125-135, 1997.

[33] Y. Xu and D. Aliaga, "Robust Pixel Classification for 3D Modeling with Structured Light", *Proc. Graphics Interface*, pp. 233-240, 2007.

[34] J. Yang, N. Ohnishi, D. Zhang, and N. Sugie, "Determining a Polyhedral Shape using Interreflections", *Proc. of IEEE Conf. on Comp. Vision and Pattern Recognition*, pp. 110-115, 1997.

Yi Xu is a Ph.D. student in Computer Science at Purdue University. His research interests are in image-based modeling and rendering, 3D reconstruction, and interactive computer graphics. He obtained a B.Eng. degree from Zhejiang University, China and a M.S. degree from University of Alberta, Canada.

Daniel G. Aliaga is an Assistant Professor of Computer Science at Purdue University. He is a researcher in computer graphics and in particular capturing and rendering large environments. Dr. Aliaga obtained his B.S. degree from Brown University and his M.S. and Ph.D. degree from University of North Carolina. He is a member of ACM SIGGRAPH and an editor for Elsevier Graphical Models.