International Journal on Computer Vision manuscript No.
(will be inserted by the editor)

Unstructured Light Scanning Robust to Indirect Illumination and Depth Discontinuities

Vincent Couture · Nicolas Martin · Sébastien Roy

Received: date / Accepted: date

Abstract Reconstruction from structured light can be greatly affected by indirect illumination such as interreflections between surfaces in the scene and sub-surface scattering. This paper introduces band-pass white noise patterns designed specifically to reduce the effects of indirect illumination, and still be robust to standard challenges in scanning systems such as scene depth discontinuities, defocus and low camera-projector pixel ratio. While this approach uses *unstructured* light patterns that increase the number of required projected images, it is up to our knowledge the first method that is able to recover scene disparities in the presence of both indirect illumination and scene discontinuities. Furthermore, the method does not require calibration (geometric nor photometric) or post-processing such as phase unwrapping or interpolation from sparse correspondences. We show results for a few challenging scenes and compare them to correspondences obtained with the Phase-shift method and the recently introduced method by Gupta *et al.*, designed specifically to handle indirect illumination.

Keywords Active reconstruction · global illumination · indirect illumination · depth discontinuities

1 Introduction

Scene reconstruction from structured light is the process of projecting a known pattern onto a scene, and use a camera to observe the deformation of the

Vincent Couture, Nicolas Martin and Sébastien Roy
Département d'Informatique et recherche opérationnelle
Université de Montréal
Montral (Québec), Canada
H3T 1J4
Tel.: (514) 343-6111 (ext.:1657)
Fax: (514) 343-5834
E-mail: {chapdelv,martinic,roys}@iro.umontreal.ca

pattern to calculate surface information. The term "structure" comes from the fact that a unique code (a finite set of patterns) is associated to each projector pixel, based on its position in the pattern. Camera-projector pixel correspondence (see Fig. 1) can then directly be established and triangulated to estimate scene depths. Results produced by structured light scanning systems greatly depend on the scene and the patterns used. In particular, it was shown in [23] that low frequency patterns create interreflections in scene concavities that cannot be removed. Another issue comes from scene depth discontinuities, where smoothness of the observed pattern can no longer be assumed.

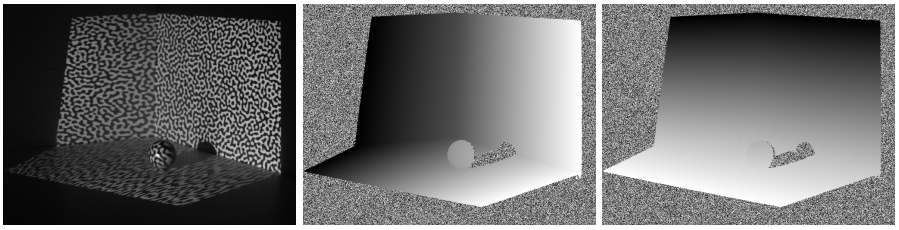


Fig. 1 Example of a scene (left) with one unstructured band-pass pattern projected on it. Several of these patterns are used to recover the x (center) and y (right) correspondence maps between the camera and the projector.

In this paper, we propose the use of band-pass white noise patterns that are specifically designed to reduce the effects of indirect illumination[1] while still being able to handle depth discontinuities. These patterns follow the basic idea of *unstructured* light patterns [20,9,31] that do not directly encode pixel position in the projector. Their only restriction is that the accumulation of such patterns uniquely identifies every projector pixels. Therefore, the correspondence of a camera pixel is no longer computed directly from the observed pattern sequence, and has to be found using an iterative high-dimensional matching algorithm. The matching method we present here is not limited to epipolar lines to avoid the need to geometrically calibrate any of the device in order to recover correspondence.

The spatial frequency of these patterns can be adjusted, making them robust to defocus (due to small depth of field, for instance) or low camera-projector pixel ratio[2]. Also, the method is designed to be independent of photometric properties (such as gamma correction) of both the projector and the camera.

The method was first presented in [8] specifically to address the problem of interreflections. Here, we include new results to show that the method also works for other types of indirect illumination such as translucency and subsurface scattering. We also compare our results with those of other methods,

[1] In the literature, indirect illumination is sometimes called *global* illumination.

[2] The camera-projector pixel ratio is defined as one camera pixel over the number of projector pixels it can see.

namely Phase-shift and a recently introduced method by Gupta *et al.* [15,16] to handle indirect illumination.

The layout of this paper is as follows. We begin in Sec. 2 by briefly reviewing prior works related to structured light patterns. We then expose in Sec. 3 common problems that may arise in structured light setups, namely indirect illumination, scene depth discontinuities and a low camera-projector pixel ratio. In Sec. 4, we introduce unstructured band-pass white noise patterns and discuss their properties. Using these patterns, matching between projector and camera pixels requires a high-dimensional match algorithm, namely locally sensitive hashing, which we describe in Sec. 5. In Sec. 6, the Gupta *et al.* method that also handles indirect illumination is reviewed. Finally, we compute in Sec. 7 camera-projector correspondence maps and reconstructions using our unstructured light patterns and compare results produced by other methods for different challenging scenes. We conclude in Sec. 8.

2 Previous work

Several sets of structured light patterns were previously proposed to perform active 3D surface reconstruction. Structured light reconstruction are often classified based on the type of encoding used in the patterns: temporal, spatial or direct [28]. Here, we also emphasize the amount of supplemental information needed by the method to work effectively. For instance, prior photometric or geometric calibration is often required.

Temporal methods multiplex codes into pattern sequence [25,29,18,14]. For instance, a pixel position is encoded in [22,25] by its binary code, representated by a concatenation of binary coded patterns. One variation introduces Gray code patterns [18] that are designed to minimize the effect of bit errors by ensuring that neighboring pixels have a code difference of only one bit. Temporal methods require a high number of patterns and the scene must remain static during the pattern acquisition process. In practice, these methods can give very good results and do not require any kind of calibration. Due to focus issues or low pixel ratio, the lowest significant bits often cannot be recovered. Solutions have been proposed, like in [14] where high frequency patterns are replaced by a shifted version of a pattern to recover the last significant bits. This method (and all variants of binary encoding patterns) also suffers from the significant indirect lighting induced by the lower frequency patterns, as we will see in the next section.

In contrast, spatial methods use the neighborhood of a pixel to recover its code [4,30,27] in order to decrease the number of required patterns. For example, the patterns can be stripes [4], grids [26] or a more complicated encoding such as the popular De Bruijn patterns [30]. Except for grids, it is worth mentioning that these patterns are one-dimensional, and thus require a geometric calibration relating the camera and the projector. Some methods even allow "one-shot" calibration [27] (i.e. only one pattern is used), but they require a very good photometric calibration. The main drawback of these methods is

that they assume spatial continuity of the scene, which does not hold at depth discontinuities. Furthermore, those methods produce sparse results, as the correspondence can be recovered only at stripe transitions of the pattern. In [34], high quality reconstructions of static scenes are computed using a multi-pass dynamic programming edge matching algorithm. The pattern is shifted over time to compensate for the sparseness of De Bruijn patterns. The number of patterns required is still a lot less than in the case of temporal methods. However, the method requires both photometric and geometric calibration.

Direct coding methods use the intensity measured by the camera to directly estimate the corresponding projector pixel. Similarly to temporal methods, no spatial neighborhood is required to obtain correspondence. Direct methods need only a few patterns, typically three patterns. Because patterns can be embedded in a single color image, one image is theoretically sufficient to recover depth. The work of [32] introduced the so-called "three phase-shift" method which relies on the projection of three dephased sinusoidal patterns. This method was modified in [35] to project only two sinusoidal patterns and a neutral image used as a texture. These methods often require the estimation of the gamma coefficient (for both the projector and the camera) and, because they are one-dimensional, a geometric calibration as well. More patterns can also be used to modulate the signal in 2D and reduce the effects of noise and gamma factors [7]. Furthermore, matching using these patterns is ambiguous due to their periodic nature. In practice, phase unwrapping is used to overcome this issue, but high frequency patterns remain ambiguous for scenes with large depth discontinuities.

We present in Sec. 4 a novel temporal method that use *unstructured* light patterns that are not dependent on projector pixel position. Similar work has been presented in [20] where scanning is performed using a sequence of photographs or a sequence of random noise patterns for flexibility purposes. Contrary to [20] however, we designed the unstructured patterns specifically to minimize the effects of indirect illumination. Another method was recently introduced in [15] to address the problem of indirect illumination using a combination of high frequency patterns, band-pass patterns and standard Gray codes. We will compare this method with our approach in Sec. 6. Our method will also address typical challenges that may arise in structured light setups. We review these in the following section.

3 Problems of structured light systems

This section reviews the problems that may arise in typical structured light setups, such as indirect lighting, varying camera-projector pixel ratios, and scene depth discontinuities. It also discusses strengths and weaknesses of the methods reviewed in Sec. 2.

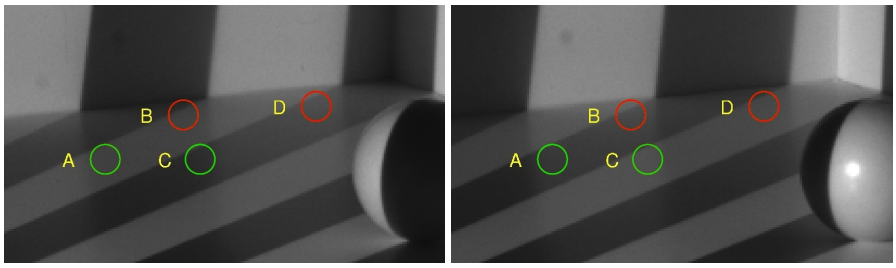


Fig. 2 A stripe pattern (left) and its inverse (right) are displayed. Measured intensities at points (A, B, C, D) are $(56, 56, 35, 71)$ and $(46, 66, 72, 65)$ in the left and right images respectively. Points B and D are incorrectly classified because of interreflection.

3.1 Indirect illumination

When a scene is lit, the radiance measured by the camera has two components, namely direct illumination due to direct lighting from the projector and indirect illumination caused by light reflected from or scattered by other points in the scene for instance[23]. It is generally assumed that when projecting a Gray (or binary) code pattern followed by its inverse, a camera pixel is lighter when observing a white stripe [28]. This is not always the case however, especially in the presence of indirect illumination, as illustrated in Figure 2 by points B and D. This situation severely deteriorates the quality of the recovered codes.

Nayar *et al.* presented in [23] a method to separate direct and indirect components of illumination. They showed that indirect illumination becomes a constant gray intensity when the pattern frequency is high enough, i.e. that geometry, reflectance map and direct illumination are smooth with respect to the frequency of the illumination pattern. Separation is done by subtracting the image of a single high frequency binary pattern and its complement, or by subtracting the minimum from the maximum intensities measured over a few patterns.

Structured light methods that use only high frequency patterns could potentially remove the effects of indirect lighting to improve performance. Phase-shift methods are good examples, but increasing the frequency also increases signal periodicity, which makes the subsequent phase unwrapping step hard if not impossible to accomplish. Therefore, lower frequency patterns tend to be used in practice [28].

For low frequency patterns, it is much harder to remove the effects of indirect illumination. A few methods were proposed to partially achieve this by modulating low frequency patterns with high frequency patterns [7,13,15]. Indirect lighting could also be estimated using a light transport matrix [24, 21,12] which relates every pixel of the projector to every pixel of the camera. However, this matrix is huge and very time consuming to measure and process. For illustration purposes, we computed this matrix, which was then transposed and remapped from projector to camera using our matching results. Figure 3

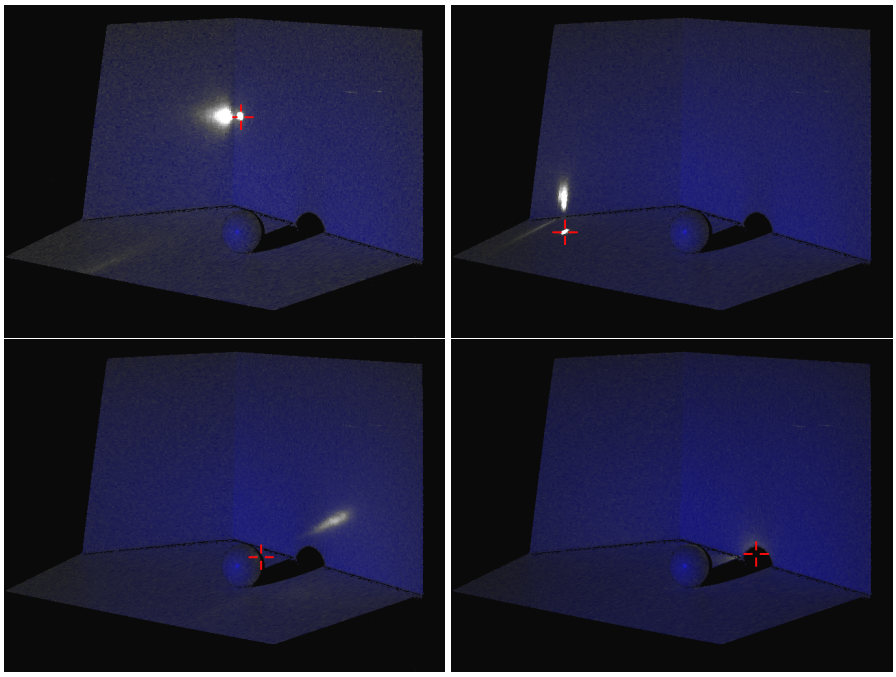


Fig. 3 Illumination contribution for selected pixels, indicated by red crosshairs. The blue color is added artificially to provide a scene reference. Top has direct lighting with inter-reflections. Bottom left feature indirect lighting. Bottom right is a pure shadow.

shows how different regions in the scene contribute to the intensity measured at selected camera pixels by creating indirect lighting. As in [23], we argue that if the pattern spatial frequency is high enough, then these contributing areas always include an equal mixture of black and white, thereby making indirect lighting near constant.

3.2 Depth discontinuities

Spatial methods such as De Bruijn patterns require a neighborhood around a pixel to estimate its code. This allows a reduction in the number of patterns, but creates problems near depth discontinuities where the camera observes a mixture of at least two projector pattern regions. This makes decoding unstable. For this reason, spatial methods require a post-processing step to remove wrong matches near discontinuities, usually a dynamic-programming minimization to add smoothness constraints on the correspondence map [34].

For temporal and direct methods, which do not require any spatial neighborhood, correspondence errors can occur when two codes at different depths are both seen by the same camera pixel. This blends two unrelated codes and affects direct methods such as Phase-shift which rely on the measured intensity to estimate correspondence.

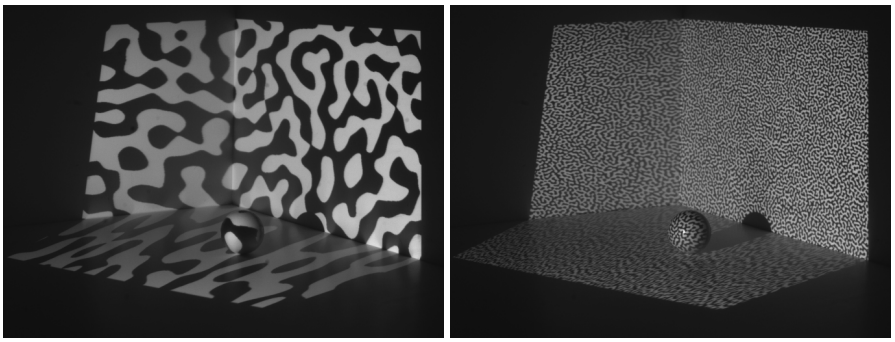


Fig. 4 Synthetic patterns are generated in the Fourier domain by randomizing phase within an octave. Here, two patterns are shown projected on a scene. Spatial frequencies used are (left) 8 to 16 cycles per frame and (right) 64 to 128 cycles per frame.

3.3 Pixel Ratio

Because of the relative geometry and resolution of the camera and projector, it is often the case that a single camera pixel captures a linear combination of two or more adjacent projector pixels. This situation often occurs in multi-projector setups, where the total resolution of the projectors is far greater than the camera resolution. This is known as having a low camera-projector pixel ratio.

The Gray code method degrades gracefully with pixel ratio, as low significant bits become too blurred to be recovered and are simply discarded. Other methods, such as De Bruijn or Phase-shift, are robust to this as long as their pattern frequencies are low enough.

4 Unstructured light patterns

This section presents our *unstructured light* method, featuring band-pass white noise patterns that are designed to be robust to indirect illumination by avoiding large black or white pattern regions.

In this paper, we consider surfaces that are mostly diffuse. If we can make one full period of our pattern smaller than the diffusion, then the effect of this diffusion is near constant for any pattern with the same frequency [23].

We limit the amplitude spectrum to a single octave, ranging from frequency f to $2f$, where a frequency refers to the number of cycles per frame. For each spatial frequency, the amplitude is set to 1 and the phase is randomized, subject to the conjugacy constraint [5], namely that $\hat{I}(f_x, f_y) = \overline{\hat{I}(-f_x, -f_y)}$.

The second step is to take the inverse 2D Fourier transform of $\hat{I}(f_x, f_y)$, yielding a periodic pattern image $I(x, y)$. To avoid periodicity, we generate a pattern larger than the desired width (say 110% larger) and then cut the extra borders. The pattern intensities are then rescaled to have values ranging

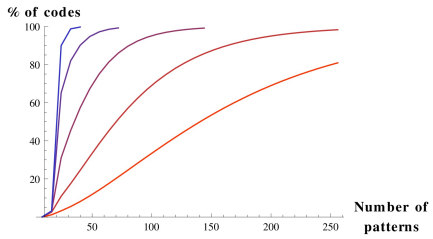


Fig. 5 For HD images (1920 × 1080 pixel resolution), the percentage of pixels having unique codes while increasing the number of patterns. The curves correspond to f ranging from 8 to 128, with steep curves corresponding to patterns of higher frequencies. Curves stop being drawn if they reached 99%.

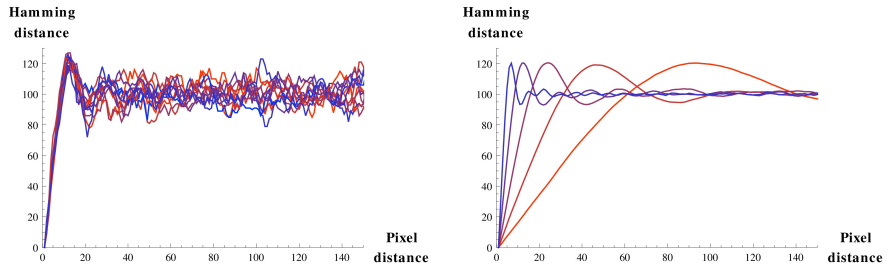


Fig. 6 Hamming distance between a randomly selected pixel and its neighbors with increasing distance, for a code length of 200 patterns. Distances are shown (a) for a few selected pixels ($f = 64$) (b) as the average over many selected pixels for different frequencies f ranging from 8 to 128, with steeper curves corresponding to patterns of higher frequencies. Each curve follows a sharp increase before decreasing to a constant that is half the number of patterns. Patterns of higher frequencies are not as correlated spatially (steeper increase).

in [0:255]. Each pattern is finally binarized with a threshold at intensity 127 to make pixels either black (≤ 127) or white (> 127).

Hence, the patterns are parametrized by frequency f and limited to a single octave of variation to control the amount of spatial correlation (see Fig. 4). More spatial correlation increases code similarity locally, but also increases the number of required patterns to guaranty code uniqueness. We next discuss these two aspects.

4.1 Reducing code ambiguity

In this section, we analyze the relationship between frequency f and the number of patterns required to identify projector pixels uniquely with a code sequence of black and white values. Note that the pattern sequence is uncorrelated temporally to ensure that all bit in a code are independent.

In Fig. 5, we measure the number of patterns required to disambiguate at least 99% of all pixels as frequency f is varied. We consider HD projectors having 1920×1080 pixels. One can see that low frequency noise requires more

patterns. Moreover, low frequency patterns often cause interreflections when large white pattern regions are projected in surface concavities and/or highly reflective materials.

Finally, we observed that this 1% of code duplicates usually correspond to small groups of neighboring pixels that have yet to be disambiguated. High frequency patterns, however, tend to quickly produce unique codes locally but have duplicates elsewhere.

One interesting question is whether 1D patterns could reduce considerably the number of required patterns. These 1D patterns could be used, for instance, in a calibrated setup for which epipolar geometry is known. Ignoring the fact that long 1D stripes do create more interreflection, using 1D stripes does reduce the number of patterns, but not considerably. The reason is that faraway codes usually get disambiguated after only a few patterns (in 1D or 2D), but local disambiguation takes a lot more patterns. For some fixed frequency, 1D disambiguation is faster than in 2D, but only by a factor or about 60% (data not shown). If two sets of 1D patterns are used (horizontally and vertically), then more patterns are actually required ($2 \times 60\%$) than a single 2D set.

4.2 Keeping neighbors similar

One important property of our patterns is the similarity between neighboring codes. Fig. 6 presents the hamming code difference with respect to the distance between two neighboring pixels. Regardless of the frequency used, the hamming difference increases gradually with distance until it reaches a negatively correlated maximum before decreasing to a constant level. The standard deviation around this plateau is that of a Binomial distribution and is equal to about 7.07 bits, that is $\frac{\sqrt{N}}{2}$ for $N = 200$ bits.

This correlation between neighboring codes makes it easier for mismatch to happen between neighbors. However, it provides great robustness to pixel ratio variations, since the averaging of a group of neighboring codes is still highly correlated to each original blended codes. Also, this provides robustness to various local imaging problems like out of focus areas because of small depth of field.

Moreover, the lack of correlation between far pixels helps provide very high robustness to scene discontinuities. When a camera pixel observes a scene discontinuity, its intensity is a blend of two uncorrelated codes. Thus, about 50% of the bits are the same in both codes and will be accurately recovered. The remaining bits belong to either code, thereby ensuring that the matching code is composed of at least 75% of all bits of these two codes. This makes them and their neighbors much more likely to match than any other distant code. In contrast, if the recovered bits of two blended Gray code patterns are not all from the same code, then the resulting code may be completely unrelated to the two blended codes.

5 Establishing pixel correspondence

This section deals with efficiently establishing the correspondence between camera and projector pixels. We designed our matching method so that it does not require any form of prior calibration. By not using any epipolar constraint, matching becomes more difficult but much more flexible. For example, the camera could be a non single view point fisheye and the projector illumination could be bouncing off a convex surface. These cases are common in multi-projection setups and are not easily calibrated [19].

A number of random unstructured light patterns are generated with a preselected band-pass frequency interval. Those patterns are projected one at a time while a camera observes the scene. N patterns are projected, captured by the camera, and then matched.

First, the gray images captured by the camera are converted into binary images for matching. The conversion is simply obtained by measuring if a pixel is above or below the average of previous patterns over time. Let $\Phi_{xy}(i)$ be a monotonic function modeling photometric distortion[3], the average image $\bar{I}_c$ in the camera, computed from all the distorted intensities in the camera, remains a good delimiter because it is well within $\Phi_{xy}(\text{black})$ and $\Phi_{xy}(\text{white})$ when, for a camera pixel, the amount of black and white values is reasonably balanced. Furthermore, the average works well because band-pass noise patterns should not produce big changes in indirect lighting.

Thus, as codes from *unstructured* light patterns no longer have any correlation to projector pixel position, pixel correspondences have to be found by matching two sets of high dimensional vectors to one another. Using N patterns, we obtain a N-dimensional binary vector for each pixel of both the camera and the projector image. For HD images, each set has around $1920 \times 1080 \approx 2$ million N-dimensional vectors. For the remainder of the section, we assume that camera pixels are matched to projector pixels, although matching can be performed the other way around (or even both ways simultaneously), which can be useful, for instance, in multi-projector systems [19] to remove the need to inverse the correspondence maps.

Efficient matching is achieved using a high-dimensional search method based on hashing of binary vectors as described in [2,10,3]. Algorithm 1 shows a pseudo-code of the matching algorithm. All vectors are hashed by selecting b-bits (hopefully noise free) out of the N code bits. We use a key size b that should cover at least the number S of pixels in the projector such that expected number of codes hashed by a single key is around 1. In practice, we use $b = \lceil \log S \rceil$. While the codes should ideally match exactly (i.e. have the same key), there is some level of noise in practice. Thus, the method proceeds in k iterations, and selects a different set of bits for each iteration.

For a given pixel, the probability P that it is matched correctly after k iterations, in other words, that its hashing key has no bit error, can be modeled

[3] Photometric distortion includes gamma factors, scene albedo and aperture [6].

as

$$P = 1 - (1 - (1 - \rho)^b)^k \tag{1}$$

where ρ is the probability that one bit is erroneous. The number of iterations required to get a match within confidence P can be computed as

$$k = \frac{\log(1 - P)}{\log(1 - (1 - \rho)^b)} \quad . \tag{2}$$

Several factors can increase the ρ value such as very low contrast and aliasing which becomes worse for higher frequency patterns and lower camera-projector pixel ratios. Thus, ρ can vary locally in the camera image, as scene albedo may change contrast for parts of the scene only. The pixel ratio may also change, in the presence of slanted surfaces for instance. Estimating ρ would yield an indicator of how many iterations are required, given the desired probability of a correct match P. However, Sec. 5.1 will introduce heuristics that improve convergence and thus, make the number of iterations predicted by ρ very pessimistic. Other termination criteria are discussed in Sec. 5.2.

Fig. 8(a) shows how adding code errors affects the convergence. We generated $N = 200$ patterns and applied a noise according to various ρ values. For instance, the best match should have an average optimal error of 20 bits for $\rho = 0.1$. One can see that convergence is still achieved for $\rho \leq 0.1$, but that it becomes much slower for higher ρ values. Since the number of iterations grows exponentially with ρ, a value larger than about 0.3 will result in no convergence.

Matching heuristics (see Sec. 5.1) can improve convergence considerably (see Fig. 8(b)). However, optimal matches do not guaranty quality matches. For instance, when $\rho = 0.3$ is used, the distribution of errors for good matches is not well separated from random codes ($\rho = 0.5$), distributed around half the number of bits $\frac{N}{2}$. We will discuss these distributions again in Sec. 5.1, in particular Fig. 9.

During an iteration, the hash table can be unbalanced, i.e. more that one code hashes in a single bin. The search for the closest code in each bin can increase significantly the matching time. In practice, the codes hashing to the same bin could be stored in a data structure accelerating the search. Instead, we select the first hashed code. Even if this strategy does not choose the best code, the time gained can be used to perform another matching iteration. Typically, the execution time for one iteration on a laptop with an Intel dual core 2.2 Ghz CPU with 2GB of RAM is around one second when matching an HD camera to an HD projector, and the iteration time is doubled when applying the heuristics.

Algorithm 1 Pseudo-code of the basic matching algorithm.

```
{assuming that the projector resolution is WxH}
k ← ceil(log(W * H)) {compute the hashing key size for a hash table of size N}
N ← 200 {number of projected patterns}
for all camera pixels i do
   match_cost[i] ← inf {init match costs to infinity}
end for
{keep matching until some criterias are met (see text)}
repeat
   mask ← RandomMaskSelect(k, N) {select k bits out of N}
   proj_hash_table.init()
   for all projectors codes P[i] do
      proj_hash ← hash(P[i], mask)
      proj_hash_table.add(proj_hash, P[i])
   end for
   for all camera codes C[i] do
      cam_hash ← hash(C[i], mask)
      P[j] ← proj_hash_table.query(cam_hash) {closest projector code to C[i]}
      cost ← HammingDistance(P[j], C[i])
      if cost < match_cost[i] then
         match[i] ← P[j]
         match_cost[i] ← cost
      end if
   end for
until some criteria is met
```

5.1 Matching heuristics

Usually, reconstruction methods take advantage of *a priori* knowledge about the scene in order to improve the results. One common assumption is that neighboring pixels have similar correspondences, thereby suggesting some form of local smoothing. Unfortunately, smoothing can introduce errors at discontinuities or wherever the assumption does not hold. In our case, we propose two simple heuristics that take advantage of scene smoothness to get a dramatic speedup in convergence. Their great advantage is that they improve the convergence time without any degradation of the final result.

The heuristics are illustrated in Fig. 7. *Forward matching* tests if a camera pixel can find a better match in the neighborhood of its current match in the projector. This heuristic refines matches that lie within the area of locally correlated region where cost increases with distance w.r.t. the best match ($\leq$ 15 pixels in Fig. 6(a)). *Backward matching* tests the neighbors of a camera pixel to check if they could also match its corresponding projector pixel. This heuristic tends to create new matches, i.e. it improve current matches with potentially uncorrelated matches ($>$ 15 pixels in Fig. 6(a)). The speedup is shown in Fig. 8, where the convergence is plotted as a function of the number of iterations needed with and without the use of the heuristics.

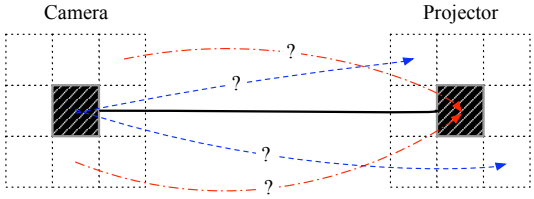


Fig. 7 When a match is found (black solid line), two simple matching heuristics can be used : *forward matching* (blue dashed lines) attempts to improve an existing match and *backward matching* (red dot-dashed lines) attempts to create neighborhood matches.

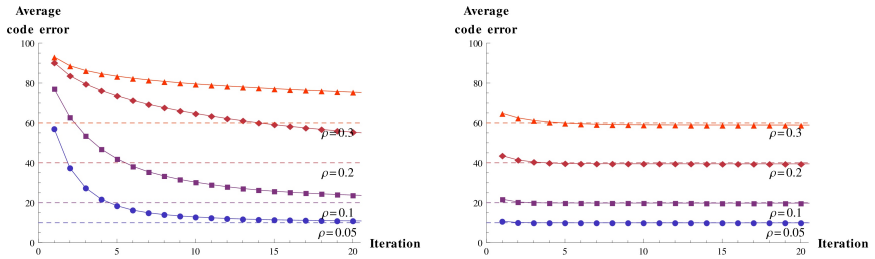


Fig. 8 For increasing noise levels ρ, convergence of the hashing method (a) without heuristics (b) with heuristics. The dashed lines represent the theoretical lowest average code error. Convergence is much faster when applying the heuristics.

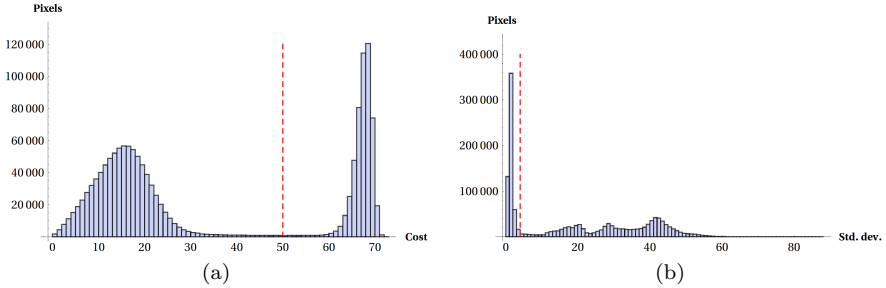


Fig. 9 For a typical scene, (a) a histogram of match costs has two distributions centered at ρN and at a value a bit below $\frac{N}{2}$ (see text for details). (b) a histogram of standard deviation of intensities has a high peak corresponding to unlit camera pixels or low contrast regions. A threshold (indicated here by the red dashed line) cannot completely separate the long tails of the distributions.

5.2 Match confidence and termination criteria

This section discusses a termination criteria to decide when to stop matching iterations. This is not a trivial problem due to the probabilistic nature of the algorithm. For instance, it can often happen that hashing improves a few matches even after there was no improvement for several iterations.

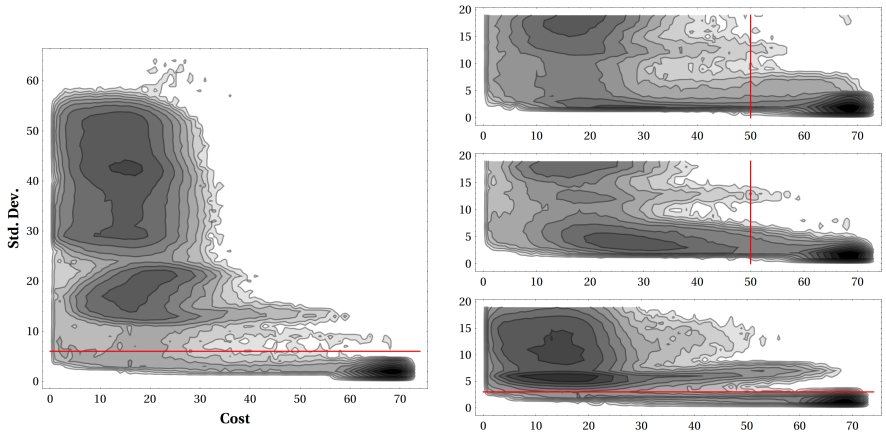


Fig. 10 2D log histograms of matching costs and standard deviations of intensity for the 4 scenes presented in the experimental results, namely (left) `Ball` and (right, top to bottom) `Games`, `Grapes & Peppers` and `Corner`. The red lines show the thresholds to remove unlit camera pixels.

Camera pixels that see a surface area not directly illuminated by the projector should be excluded from the matching process because they produce random codes that depend on camera noise. The matching process would keep improving these matches, making a termination criteria more difficult to establish. Looking at the matching costs or standard deviations of intensity could be a good strategy to detect most of the unlit camera pixels. Fig.9(a) shows a histogram of the matching costs for a typical scene after 50 iterations. The matching costs are distributed in two well separated Binomial-like distributions, namely one centered at ρN and one centered below $\frac{N}{2}$ (in Fig.9, $N = 200$ and $\rho \approx 0.1$). The first distribution corresponds to correctly matched camera pixels. The second distribution corresponds to unlit pixels; its mean is lower than $\frac{N}{2}$, because only the minimum matching code is kept at each iteration. Fig.9(b) shows a histogram of the standard deviations of pixel intensities. The distribution is roughly bimodal, with the highest peak corresponding to mostly unlit pixels. This narrow peak illustrates well the fact that all the patterns produce near constant indirect illumination for a given scene. Gray codes do not feature this property. The rest of the distribution is composed of lit pixels, modulated by the scene reflectance.

However, this peak also contains pixels corresponding to dark scene objects. Because of this ambiguity, we consider both criteria, as illustrated in Fig. 10. Because of the long tails of the distributions, there is usually no single threshold which can separate all good matches from wrong matches. For most scenes, either criteria works. For scenes with dark objects, saturated or noisy imaging conditions, one criteria might work better than the other. The red lines illustrates the thresholds we used for the different scenes. In practice, both criteria could be used at the same time.

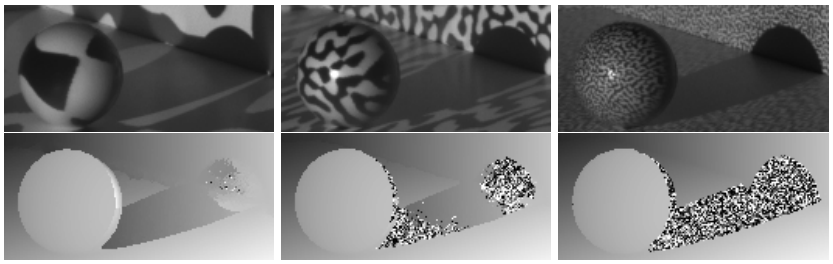


Fig. 11 Correspondence from unstructured patterns at frequencies 8 (left), 32 (middle) and 128 (right). The effects of using higher frequency patterns are exposed on the edge of the ball and its shadow.

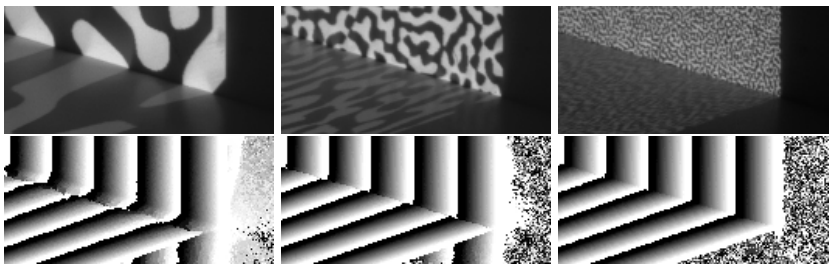


Fig. 12 Correspondence from unstructured patterns at frequencies 8 (left), 32 (middle) and 128 (right). The effects of using higher frequency patterns are exposed at the corner of the wall and the ground.

Once the unlit camera pixels are discarded, we can iterate until only a small number of pixels are updated (say 5 pixels) for a few iterations (say 5 iterations). Very few match errors may remain, usually less than 0.01% of all pixels (20 or 30 pixels). These are typically located where strong interreflection remains, such as the intersection of two walls. There, the high code errors makes the heuristics inefficient. An exhaustive search is then performed for all matches that are not smooth with respect to their neighbors, in the hope of finding a better match. Smoothness for a camera pixel is simply checked by considering the average match of its neighbors, and verifying that it is within a threshold distance τ (we use τ=1.5). Note that this smoothness condition will also select all depth discontinuities as potential match errors, thereby subjecting them to an exhaustive search. This search is repeated until no further updates are made.

5.3 First results

In this section, we present the first results of our method on a real scene composed of two walls, a floor and a ball (see Fig. 1). The scene contains significant interreflections, depth discontinuities and out of focus regions. A more detailed comparison with other methods will be presented in Sec. 7.

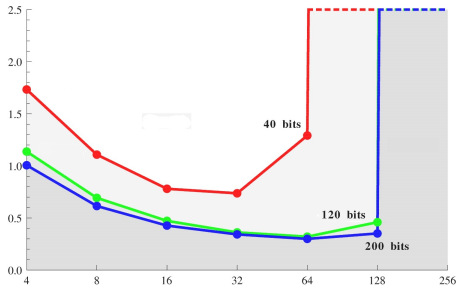


Fig. 13 Average correspondence cost as a function of pattern frequency (4,8,...,256), for various code lengths (40,120 and 200 bits). Observe that more bits give lower errors. Low frequency patterns give slightly larger average errors because they required even more than 200 bits to disambiguate all pixels locally. High frequency patterns suffer from aliasing which makes convergence harder to achieve.

Our method gives x and y correspondence maps, as illustrated in Fig. 1. A frequency f of 128 cycles per image was used. Furthermore, we tested our method over a range of unstructured pattern frequencies. The x correspondences for selected regions are shown in Figs. 11 and 12. Notice that for regions not lighted directly, random codes are expected. This is observed behind the ball (Fig. 11 (right)). High-frequency patterns also improve matching on the floor near the wall.

Finally, using the best results of our method as a reference, we measured errors by varying pattern frequencies and the number of patterns used. Fig. 13 shows that errors are smaller with more patterns and middle frequencies. Low frequencies are unsuitable to reduce the effects of indirect lighting, and more patterns are required to disambiguate codes locally. The fact that middle frequency patterns (here 32 and 64 cycles per frame) perform better than very high frequency patterns shows a tradeoff in the choice of frequency. While very high frequencies (here 256 cycles per frame) would be ideal to make indirect illumination near constant, they suffer from the problem of camera aliasing, i.e. the camera resolution needs to be is sufficiently high to resolve the signal. They are also more prone to loss of SNR due to local blurring effects such as sub-surface scattering and defocus.

6 Comparison with the Gupta *et al.* method

This section compares our method to the method recently introduced in Gupta *et al.* [15] to address indirect illumination. Their method uses four set of codes, standard Gray codes and three other sets optimized for different illumination effects.

First, they address what they classify as long-range illumination (diffuse and specular interreflections) with the use of high-frequency patterns, generated by combining a chosen high-frequency base pattern with standard Gray codes through the XOR operation. From the captured images, the original

Gray code patterns can be recovered by performing the XOR operation again with the same chosen pattern. Although this pattern could be any high-frequency pattern, Gupta *et al.* use the two highest Gray code patterns to generate two sets of patterns, namely XOR-2 and XOR-4 patterns (2 and 4 correspond to the maximum stripe width in both sets). Note that this choice produces narrow but very long stripes, which is not the case in our patterns. Effects of indirect illumination could probably be reduced further by choosing a base pattern that limits the stripes in both directions.

Second, they address short-range effects (sub-surface scattering and defocus) that can severely blur the high-frequency patterns, leading to a lot of code errors during the binarization process. To avoid this, Gupta *et al.* use a set of patterns called min-SW Gray codes [11], featuring stripe widths between 8 and 32 respectively.

Note that the XOR-2 and XOR-4 patterns do not maintain the basic Gray code property, namely that a code and any one of its neighbors differ only by one bit. This property ensures that if a camera pixel observes a mixture of two neighboring codes, then the dominant code is chosen from the one black/white transition (i.e. one bit difference). But if more than one transition exists between two codes, then there is no guarantee that all dominant bits come from the same code, and the resulting code may then correspond to an unrelated far away pixel position. This is especially true in the presence of interreflections. In contrast, our method ensures local coherence.

In [15], good correspondences are chosen if they match in at least two sets of codes. Otherwise, a camera pixel is flagged as an error. In our implementation of the method, we matched codes in x and y separately and we considered that two matches agreed if their pixel distance was less or equal to 2. As in [16], we applied a 3×3 median filtering on each of the 4 correspondence maps to remove noisy matches due to pixel aliasing. Note that we did not address in this paper the iterative error correction process[33,15] which captures additional patterns that include only unmatched projector pixels. While this process can be effective to decrease indirect illumination given a good error detection criteria, we argue that it should ideally not be required for robust patterns.

7 Experiments

In order to test the performance of our proposed method, we scanned several challenging scenes using a Gige Prosilica 1360 camera and a Samsung P400 projector. The pixel resolution of the camera and the projector were 1360×1024 and 800×600 respectively. We tested four scenes that exhibit different challenges: `Ball`, `Games`, `Grapes & Peppers` and `Corner` (see Fig. 14). We compared correspondence results from our method based on unstructured light (42 patterns), the Gupta *et al.* method using all 4 sets of patterns (42

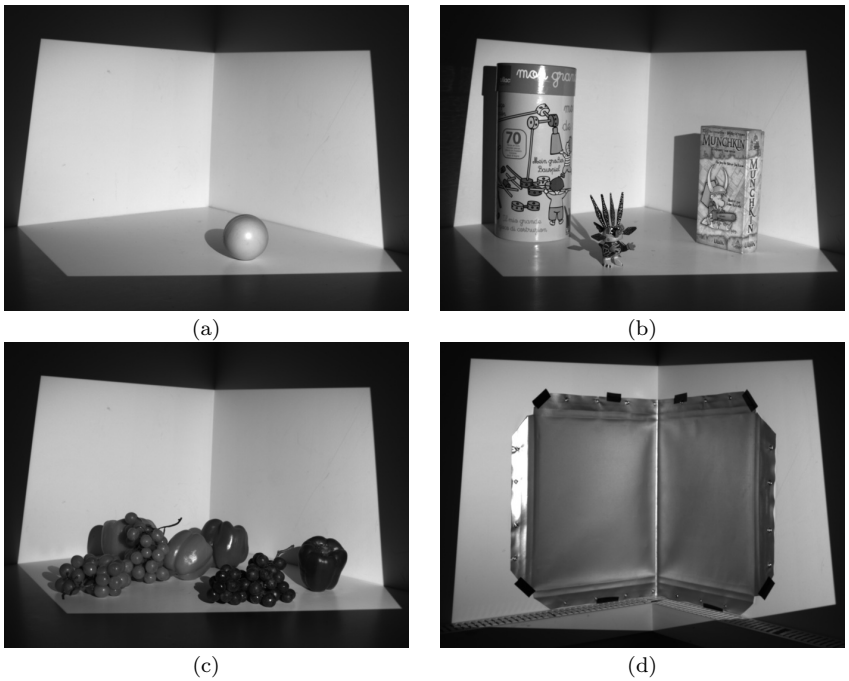


Fig. 14 The four scenes that we tested, namely (a) `Ball`, (b) `Games`, (c) `Grapes & Peppers` and (d) `Corner`.

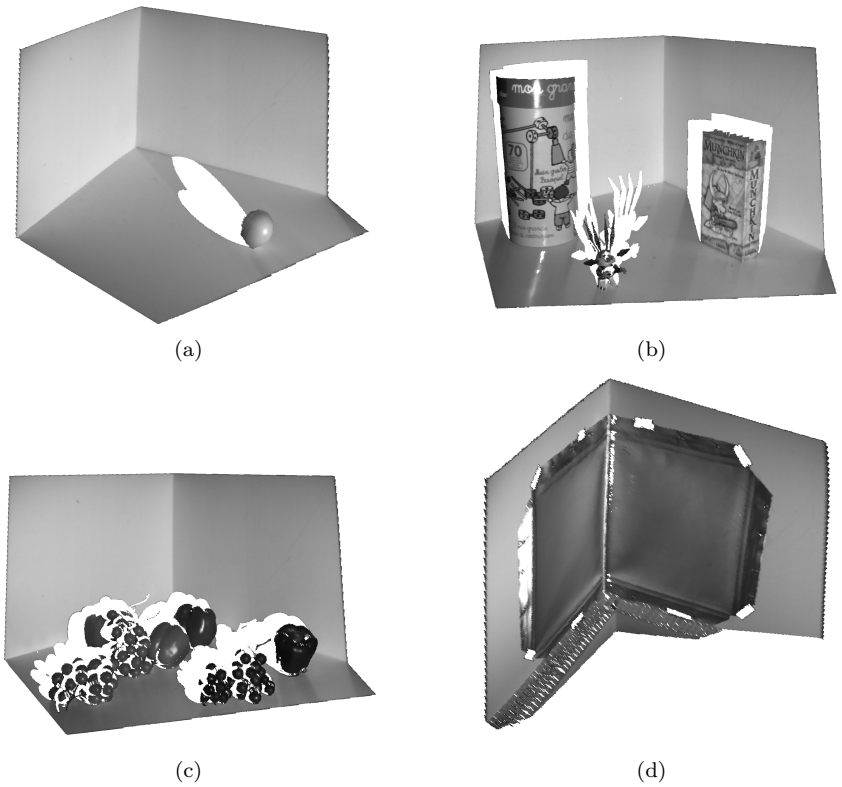
patterns[4]), the Gupta method using XOR-4 patterns only (12 patterns) and Phase-shift (3 patterns). More results are available online at [1], including results produced by Gray codes and by using more unstructured light patterns.

We generated the unstructured patterns using $f = 64$, i.e. with frequencies ranging from 64 to 128 cycles per frame horizontally. We chose this range as it is about 4 times below the Nyquist frequency limit of 400 cycles per frame in both the camera and the projector (the camera-projector pixel ratio is approximatively 1 for our setup). This adds robustness to out of focus regions and in the presence of subsurface scattering. Note that the curvy stripes of the patterns are thus about 4 pixels wide. This is similar to the XOR-4 codes, although the latter also contain stripes as narrow as 2 pixels which are not as robust.

Using these frequencies, the number of patterns required so that each projector pixel has a unique code[5] is about 80. However, we used only 42 patterns for a fairer comparison with the Gupta *et al.* method. These still produce more than 99.9% projector pixels with unique code, but the lower number of

[4] For a 800×600 projector resolution, the Gupta *et al.* method requires 10 patterns for each set of codes, plus an all white and an all black pattern to get a good estimate of the mean gray intensity for decoding purposes [16].

[5] Generating 1D unstructured light patterns reduces the number of required patterns, but the longer vertical strips create more indirect illumination than 2D patterns.



(a) (b)

(c) (d)

Fig. 15 Triangulation from the correspondences given by our method for (a) the `Ball` scene, (b) the `Games` scene, (c) the `Grapes & Peppers` scene and (d) the `Corner` scene.

patterns makes outlier matches more likely and so we apply a 3×3 median filter on the resulting matches. Note that our method finds x and y correspondence maps but that we only display the former for comparison with the other methods.

For each match result, we also computed the pixel difference with an unfiltered *reference* match given by our method using 200 patterns. For visualization purposes, the differences were scaled by 64, i.e. a 1-pixel difference has 64 pixel intensity, a 2-pixel difference has a 128 pixel difference, etc. The quality of the reference match can be seen by looking at the corresponding scene reconstructions by triangulation shown in Fig. 15.

Ball

The `Ball` scene is similar to the scene used in Sec. 5.3. It is composed of two walls, a floor and a ball that creates a highlight and a depth discontinuity at its boundary. Results of all tested methods are shown in Fig. 16. Our method (top row) gives good results with errors at the depth discontinuity and at the intersection of the wall and the ground. Using more patterns increases robustness at these locations (see online results [1]). The Gupta *et al.* method

(2^{nd} row) also performs well, but the voting scheme fails on the ground near the wall where interreflections are higher. Using only XOR-4 patterns (3^{rd} row) performs better there, but performs worse on the out of focus foreground. Phase-shift (last row) gives good results but with errors near the ground/wall intersection[6].

In order to verify that all methods perform similarly when unaffected by indirect illumination, we selected a region where indirect illumination is negligeable, namely the upper left region of the left wall, and compared the matches of all methods. At least 80% of the matches were exactly the same. All the remaining matches were within a distance of one pixel.

Games

Fig. 17 shows results for the `Games` scene, which exhibit a lot of sharp discontinuities. Also observe the curved surface of the cylindrical box, especially the soft edges at the sides where surface normals become perpendicular to the optical axis of the camera. Our method successfully matches all these problematic areas. However, it has problems on the top of the rectangular box. Note that while Phase-shift and the Gupta *et al.* method using only XOR-4 patterns seem to be performing better there, it is in fact light reflected from the wall that is being matched, thus the large error w.r.t. the reference match. The Phase-shift result exhibits wavy patterns due to light bouncing off the cylindrical and rectangular boxes.

Grapes & Peppers

Results for the `Grapes & Peppers` scene are shown in Fig. 18. Grapes are translucent fruits that create subsurface scattering, and peppers have very shiny surfaces. Subsurface scattering is especially challenging to high-frequency patterns because they become blurry. Our method works well but has larger errors on the pepper in the middle of the frame. There, the slanted surface and subsurface scattering make the patterns very blurry and using more patterns would have increased robustness considerably. The standard Gupta *et al.* method actually works better here than using only XOR-4 patterns because the latter are too high frequency. While it is true that one could apply XOR-8 or XOR-16 patterns [16], this underlines that it is not obvious to select the right set of patterns beforehand. Moreover, a set of patterns might give good results on some parts of the scene but not another. This was the reason to use multiple sets of patterns in the Gupta *et al.* method. Phase-shift produces good results but wavy artifacts can be seen on the peppers.

Corner

The `Corner` scene was made using two highly reflective surfaces set at a 90 degree angle. Both our method and the Gupta *et al.* method using only XOR-4

[6] We computed Phase-shift using three 64 cycles per frame patterns and calibrating the nonlinearities related to gamma coefficients of the camera and the projector. We here ignore issues related to the ambiguous periodicity of the signal as we are only interested in how well the phase can be recovered. Thus, we performed phase unwrapping by by looking at the reference match and finding the most likely period for each pixel independently.

patterns perform well except very near to the corner. Phase-shift also performs well but exhibits a periodic error even far away from the corner. The Gupta *et al.* method performed poorly for this scene. Notice that we pruned matches on the black tape holding the reflective material as it has very low reflectance.

8 Conclusion

In this paper, we addressed the problem of indirect illumination in structured light systems by taking advantage of a new approach to active reconstruction that uses patterns unrelated to projector pixel position. The only constraint imposed on these unstructured light patterns is that a sequence of these patterns identifies every projector pixels by a unique code. The proposed band-pass white noise patterns are designed to reduce the effects of indirect illumination and be robust to other issues such as low camera-projector pixel ratios. Because of the high number of patterns, the method is robust to capture errors and the matching algorithm provides very good performance with respect to depth discontinuities. Future works could address the problem of estimating matches at sub-pixel precision, as well as reducing the number of patterns by increasing the amount of new information given by each pattern, while still keeping their basic properties. It would also be interesting to investigate if the method could be used when multiple light sources are used [13] or when the projector is moving [17].

References

1. http://vision3d.iro.umontreal.ca/en/projects/unstructured-light-scanning/
2. Andoni, A.: Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In: IEEE Symposium on Foundations of Computer Science, pp. 459–468. IEEE Computer Society (2006)
3. Andoni, A., Indyk, P.: Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. Communications of the ACM **51**(1), 117–122 (2008)
4. Boyer, K., Kak, A.: Color-encoded structured light for rapid active ranging. IEEE Transactions on Pattern Analysis and Machine Intelligence **PAMI-9**(1), 14–28 (1987)
5. Bracewell, R.N.: The fourier transform and its applications (1965)
6. Caspi, D., Kiryati, N., Shamir, J.: Range imaging with adaptive color structured light. IEEE Transactions on Pattern Analysis and Machine Intelligence (1998)
7. Chen, T., Seidel, H.P., Lensch, H.P.A.: Modulated phase-shifting for 3d scanning. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2008)
8. Couture, V., Martin, N., Roy, S.: Unstructured light scanning to overcome interreflections. IEEE International Conference on Computer Vision (ICCV) ((to appear) Nov. 2011)
9. Davis, J., Nehab, D., Ramamoorthi, R., Rusinkiewicz, S.: Spacetime stereo: A unifying framework for depth from triangulation. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **27**(2), 296–302 (2005)
10. Gionis, A., Indyk, P., Motwani, R.: Similarity search in high dimensions via hashing. In: VLDB '99: Proceedings of the 25th International Conference on Very Large Data Bases, pp. 518–529. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1999)
11. Goddyn, L., Gvozdjak, P.: Binary gray codes with long bit runs. Electronic Journal of Combinatorics **10**, 27 (2003)

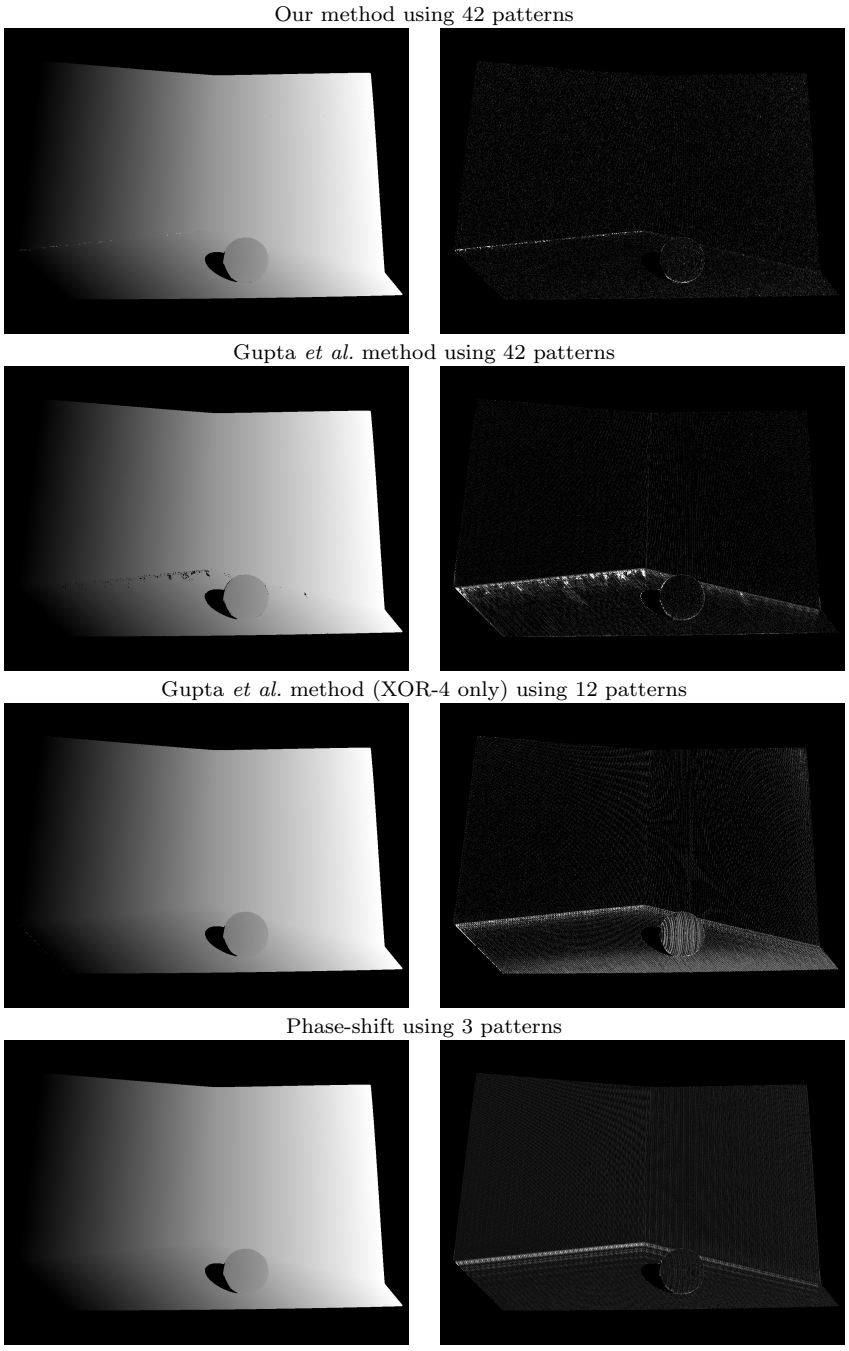


Fig. 16 Results for the `Ball` scene. The left column show the x correspondence map given by the tested methods. The right column shows the pixel difference w.r.t. the correspondences given by our method using 200 patterns.

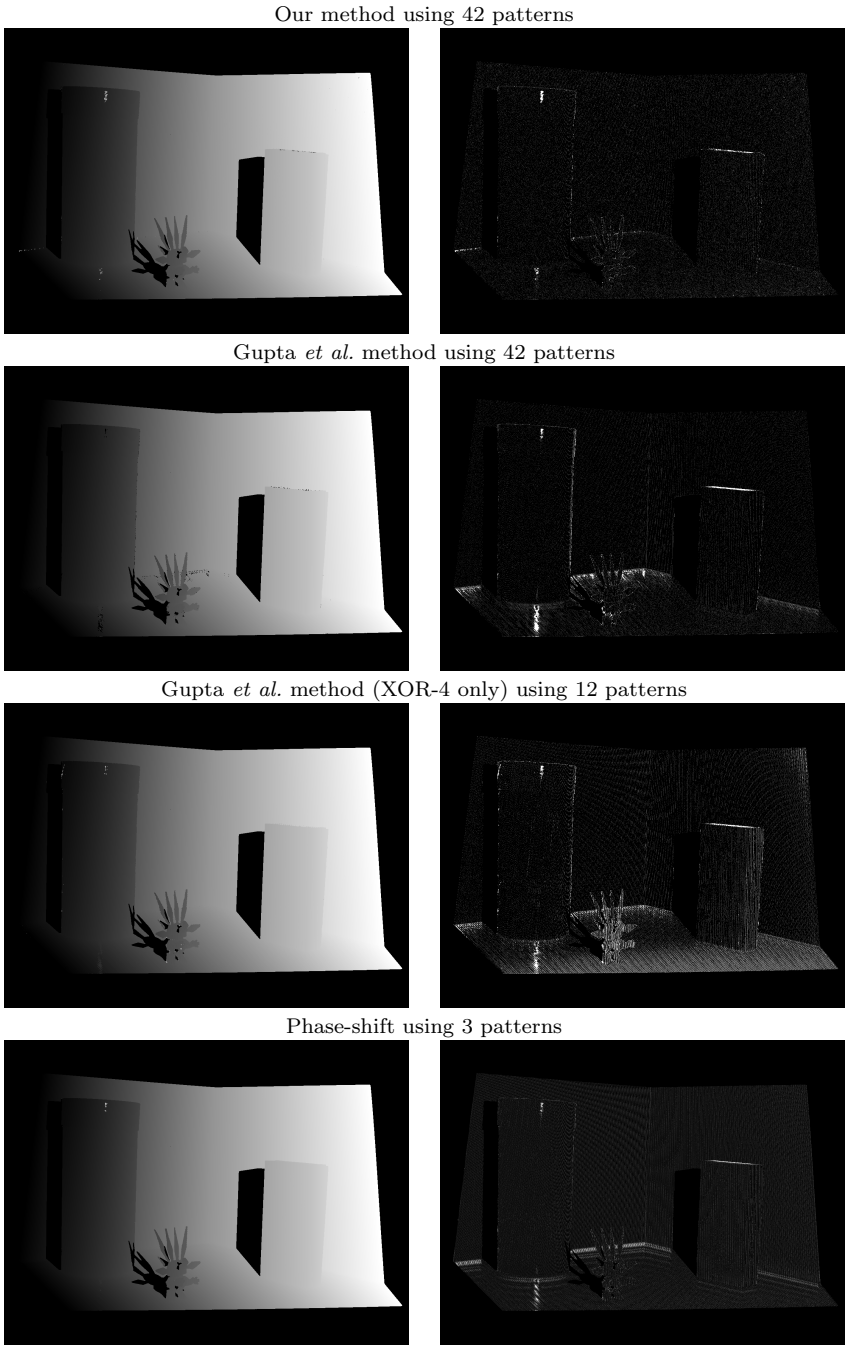


Fig. 17 Results for the `Games` scene. The left column show the x correspondence map given by the tested methods. The right column shows the pixel difference w.r.t. the correspondences given by our method using 200 patterns.

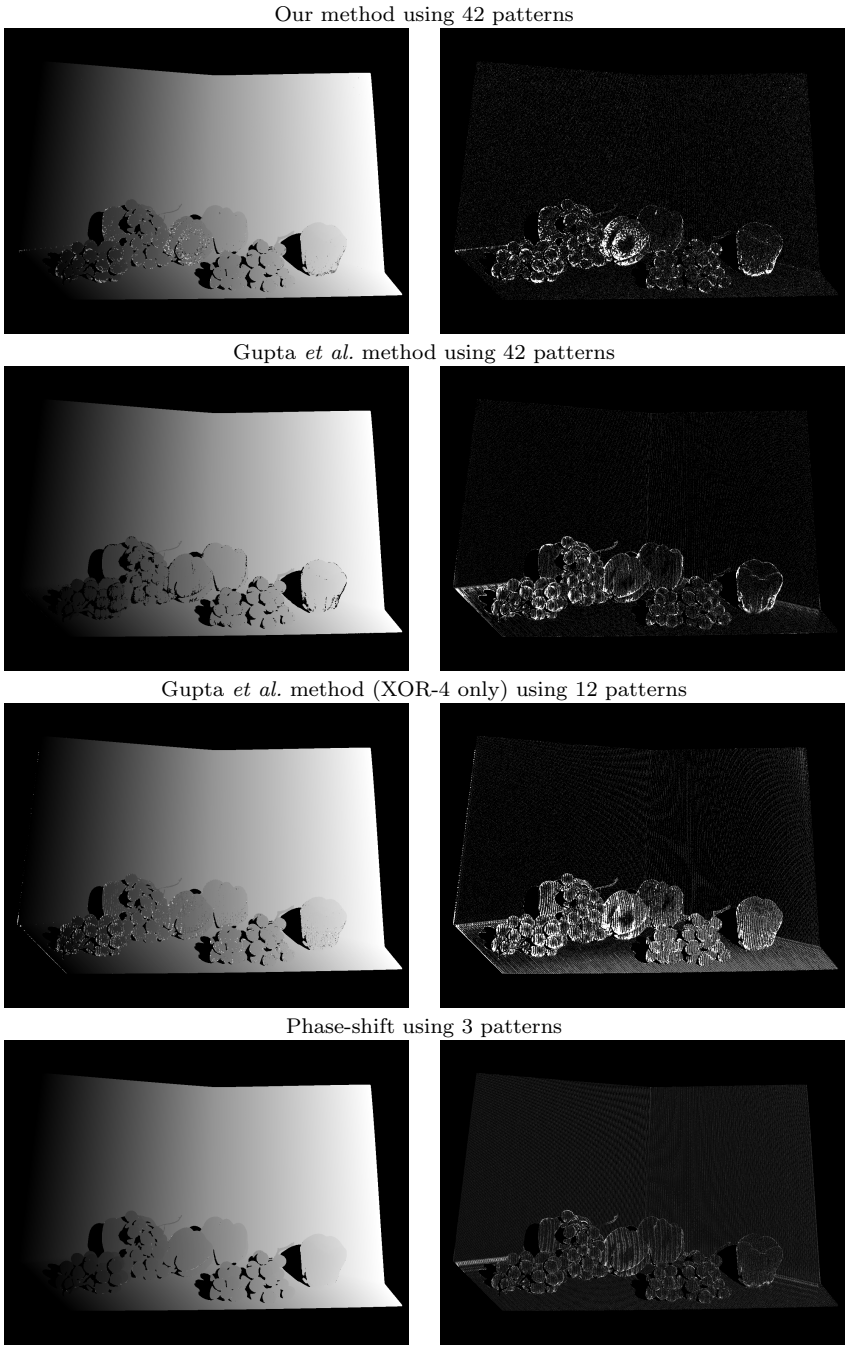


Fig. 18 Results for the `Grapes & Peppers` scene. The left column show the x correspondence map given by the tested methods. The right column shows the pixel difference w.r.t. the correspondences given by our method using 200 patterns.

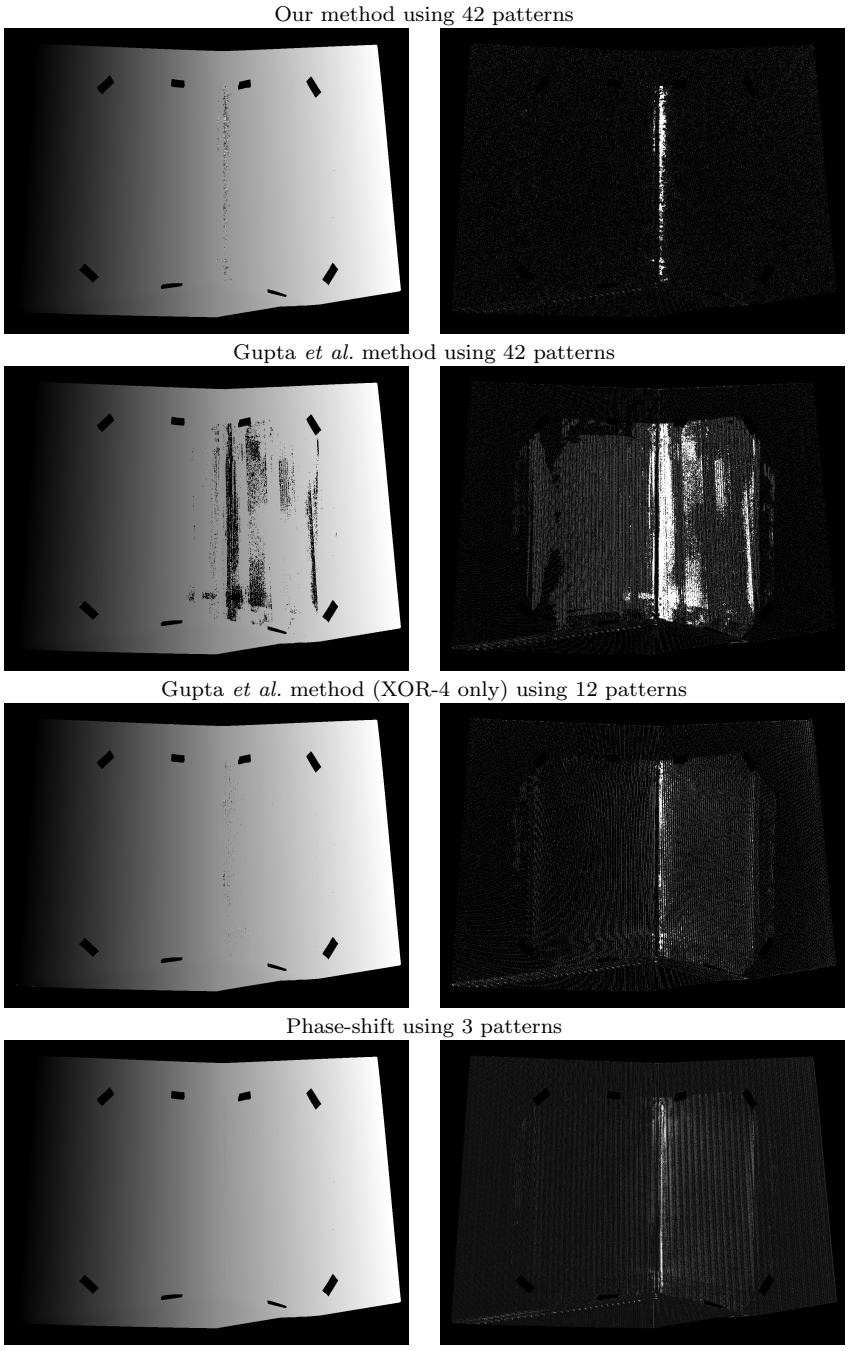


Fig. 19 Results for the `Corner` scene. The left column show the x correspondence map given by the tested methods. The right column shows the pixel difference w.r.t. the correspondences given by our method using 200 patterns.

12. Gortler, S.J., Grzeszczuk, R., Szeliski, R., Cohen, M.F.: The lumigraph. In: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, SIGGRAPH '96, pp. 43–54. ACM, New York, NY, USA (1996)
13. Gu, J., Kobayashi, T., Gupta, M., Nayar, S.K.: Multiplexed illumination for scene recovery in the presence of global illumination. In: IEEE International Conference on Computer Vision (ICCV), pp. 1–8 (2011)
14. Gühring, J.: Dense 3-d surface acquisition by structured light using off-the-shelf components. Videometrics and Optical Methods for 3D Shape Measurement (2001)
15. Gupta, M., Agrawal, A., Veeraraghavan, A., Narasimhan, S.G.: Structured light 3d scanning in the presence of global illumination. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 713–720. IEEE Computer Society (2011)
16. Gupta, M., Agrawal, A., Veeraraghavan, A., Narasimhan, S.G.: A practical approach to 3d scanning in the presence of interreflections, subsurface scattering and defocus. In: International Journal of Computer Vision, pp. 1–23 (2012)
17. Hermans, C., Francken, Y., Cuypers, T., Bekaert, P.: Depth from sliding projections. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1865–1872 (2009)
18. Inokuchi, S., Sato, K., Matsuda, F.: Range imaging system for 3-d object recognition. In: ICPR84, pp. 806–808 (1984)
19. J.-P. Tardif, S.R., Trudeau, M.: Multi-projectors for arbitrary surfaces without explicit calibration nor reconstruction. International Conference on 3-D Digital Imaging and Modeling pp. 217–224 (2003)
20. Kushnir, A., Kiryati, N.: Shape from unstructured light. In: 3DTV07, pp. 1–4 (2007)
21. Levoy, M., Hanrahan, P.: Light field rendering. In: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, SIGGRAPH '96, pp. 31–42. ACM, New York, NY, USA (1996)
22. Minou, M., Kanade, T., Sakai, T.: A method of time-coded parallel planes of light for depth measurement. Transactions of the Institute of Electronics and Communication Engineers of Japan **E64**(8), 521–528 (1981)
23. Nayar, S.K., Krishnan, A., Grossberg, M.D., Raskar, R.: Fast separation of direct and global components of a scene using high frequency illumination. ACM Transactions on Graphics **25**, 935–944 (2006)
24. Peers, P., Mahajan, D.K., Lamond, B., Ghosh, A., Matusik, W., Ramamoorthi, R., Debevec, P.: Compressive light transport sensing. pp. 3:1–3:18. ACM, New York, NY, USA (2009)
25. Posdamer, J., Altschuler, M.: Surface measurement by space-encoded projected beam systems. Computer Graphics and Image Processing (1982)
26. Proesmans, M., Van Gool, L., Oosterlinck, A.: One-shot active 3d shape acquisition. Pattern Recognition, 1996., Proceedings of the 13th International Conference on **3**, 336 – 340 vol.3 (1996)
27. Salvi, J., Batlle, J., Mouaddib, E.: A robust-coded pattern projection for dynamic 3d scene measurement. Pattern Recognition Letters (1998)
28. Salvi, J., Pagès, J., Batlle, J.: Pattern codification strategies in structured light systems. Pattern Recognition **37**, 827–849 (2004)
29. Scharstein, D., Szeliski, R.: High-accuracy stereo depth maps using structured light. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)2003, vol. 1, pp. 195–202 (2003)
30. Vuylsteke, P., Oosterlinck, A.: Range image acquisition with a single binary-encoded light pattern. IEEE Transactions on Pattern Analysis and Machine Intelligence **12**(2), 148–164 (1990)
31. Wexler, Y., Fitzgibbon, A.W., Zisserman, A.: Learning epipolar geometry from image sequences. Computer Vision and Pattern Recognition, IEEE Computer Society Conference on **2**, 209 (2003)
32. Wust, C., Capson, D.: Surface profile measurement using color fringe projection. Machine Vision and Applications (1991)
33. Xu, Y., Aliaga, D.G.: An adaptive correspondence algorithm for modeling scenes with strong interreflections. IEEE Transactions on Visualization and Computer Graphics **15**, 465–480 (2009)

34. Zhang, L., Curless, B., Seitz, S.: Rapid shape acquisition using color structured light and multi-pass dynamic programming. In: International Symposium on 3D Data Processing Visualization and Transmission. 3DPVT 2002, pp. 24 – 36 (2002)
35. Zhang, S., Yau, S.: High-speed three-dimensional shape measurement system using a modified two-plus-one phase-shifting algorithm. Optical Engineering (2007)