Unstructured light scanning to overcome interreflections

Conference Paper *in* Proceedings / IEEE International Conference on Computer Vision. IEEE International Conference on Computer Vision · November 2011

DOI: 10.1109/ICCV.2011.6126458 · Source: DBLP

3 authors, including:

Vincent Chapdelaine-Couture
Université de Montréal
14 PUBLICATIONS **113** CITATIONS

SEE PROFILE

Unstructured Light Scanning to Overcome Interreflections

Vincent Couture
Université de Montréal
chapdelv@iro.umontreal.ca

Nicolas Martin
Université de Montréal
martinic@iro.umontreal.ca

Sébastien Roy
Université de Montréal
roys@iro.umontreal.ca

Abstract

Reconstruction from structured light can be greatly affected by interreflections between surfaces in the scene. This paper introduces band-pass white noise patterns designed specifically to reduce interreflections, 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 scene discontinuities and interreflections. Furthermore, the method does not require calibration (geometric nor photometric) or post-processing such as dynamic programming or phase unwrapping. We show results for a challenging scene and compare them to correspondences obtained with the well-known Gray code and Phase-shift methods.*

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 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 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 [14] 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.

In this paper, we propose the use of band-pass white noise patterns that are specifically designed to reduce indirect lighting while still being able to handle depth discontinuities. These patterns follow the basic idea of *unstructured* light patterns [12, 7, 22] that are not based on 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 directly given by the observed pattern sequence, and has to be found using a high-dimensional search algorithm. The search method we present here is not limited to epipolar lines to avoid the need to geometrically calibrate any of the device to recover correspondence.

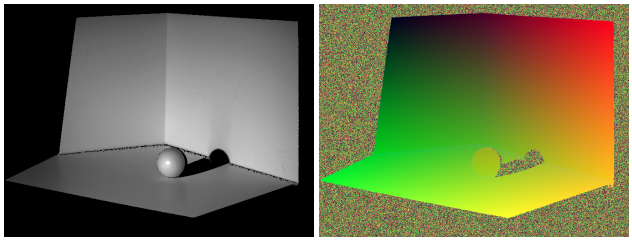


Figure 1. Example of a scene (left) and its correspondence map (right) with red and green are used for x and y coordinates respectively.

The spatial frequency of these patterns can be adjusted, making them robust to defocus (due to small depth of field, for instance) or small camera-projector pixel ratio[1]. Also, the method is designed to be independent of photometric properties of both the projector and the camera.

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 lighting, 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 search algorithm, namely locally sensitive hashing, which we describe in Sec. 5. Finally, we compute in Sec. 6 a camera-projector correspondence map using our unstructured light patterns and compare results produced by the standard Gray code and Phase-shift methods. We conclude in Sec. 7.

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

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 [19]. Here, we also emphasize the amount of supplemental information needed by the method to effectively work. For instance, prior photometric or geometric calibration is often required.

Temporal methods multiplex codes into pattern sequence [16, 20, 11, 10]. For instance, binary coded patterns introduced in [16] represent, after concatenation, a unique bit code for each pixel. One variation of this is Gray code patterns [11] 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 whole 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, highest bits often cannot be recovered. Solutions have been proposed, like in [10] 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 generated 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, 21, 18] in order to decrease the number of required patterns. For example, the patterns can be stripes [4], grids [17] or a more complicated encoding such as the popular De Bruijn patterns [21]. 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 [18] (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 very sparse results, as the correspondence can be recovered only at stripe transitions of the pattern. In [24], 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 measured intensity by the camera to directly estimate the 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 [23] introduced the so-called "three phase-shift" method which relies on the projection of three dephased sinusoidal patterns. This method was modified in [25] 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. 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 [12] where scanning is performed using a sequence of photographs or a sequence of random noise patterns for flexibility purposes. Contrary to [12] however, we designed unstructured patterns specifically to minimize indirect illumination. The method should also be robust to typical challenges that may arise in structured light setups. We review these in the following section.

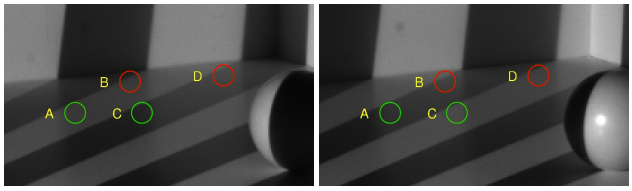


Figure 2. Interreflection problem. A stripe pattern (left) and its inverse (right) are displayed. Measured intensities at points (A, B, C, D) are $(56, 56, 35, 71)$ and $(46, 66, 72, 65)$ in the left and right images respectively. Points B and D are incorrectly classified because of interreflection.

3. 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.

3.1. Interreflection

It is generally assumed that when projecting a stripe pattern followed by its inverse, a camera pixel is lighter when observing a white stripe [19]. This is not always the case however, as illustrated in Figure 2 by points B and D. This situation severely deteriorates the quality of the recovered codes. As noted in [14], interreflections are more troublesome for low frequency projected light patterns.

When a scene is lit, the radiance L measured by the

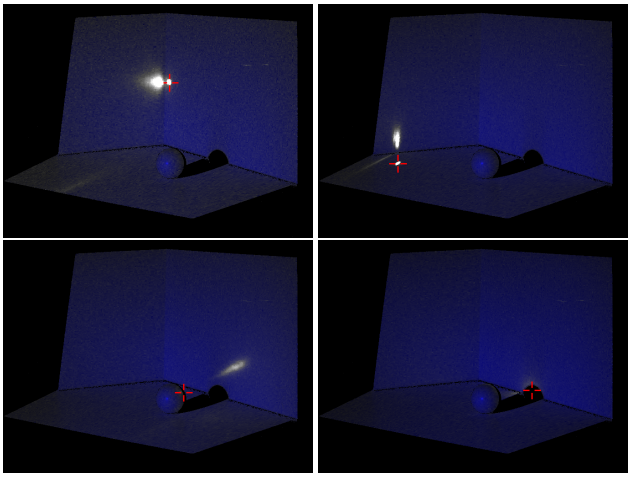


Figure 3. Illumination contribution for selected pixels, indicated by red crosshairs. The blue color is added artificially to provide a scene reference. Top has direct lighting with interreflections. Bottom left feature indirect lighting. Bottom right is a pure shadow.

camera has two components, namely direct lighting L_d due to direct illumination from the projector and L_g caused by light reflected from other points in the scene [14]:

$$L = L_d + L_g.$$

Indirect illumination L_g is often modeled by an integral of the light contribution of all scene points. In [14], a method was presented to separate direct and indirect components of illumination using a few high frequency patterns and their complement. This separation was possible for high frequency patterns only because geometry, reflectance map and direct illumination are assumed smooth with respect to the frequency of the illumination pattern.

Thus, structured light methods that use only high frequency patterns could potentially remove the effects of indirect lighting to improve performance. This would not be possible for low significant bits patterns of the Gray code method. For low frequency patterns, indirect lighting must be estimated using a light transport matrix [15, 13, 9] that relates every pixel of the projector to every pixel of the camera. This matrix however is very time consuming to capture 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 shows how indirect light contributes to the intensity perceived at selected camera pixels.

As noted by [14], the Phase-shift method can be used to estimate indirect light if the patterns have high enough frequency. In practice however, the increased periodicity makes the subsequent phase unwrapping step hard if not impossible to accomplish. Therefore, lower frequency patterns tend to be used [19].

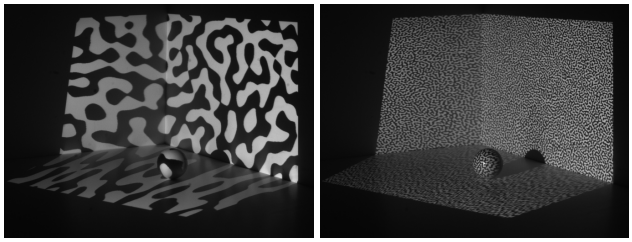


Figure 4. Synthetic noise patterns are generated in the Fourier domain by randomizing phase within an octave. Here, two patterns are shown projected on a scene. Vertical frequencies used are (left) 8 to 16 cycles per frame and (right) 64 to 128 cycles per frame.

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 at depth discontinuities where the camera observes a mixture of at least two projector pattern regions. This makes decoding ambiguous. 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 [24].

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 that directly rely on the measured intensity to estimate correspondence.

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 larger 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 detected 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 interreflections 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 [14].

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, amplitude is set to 1 and 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 $\hat{I}(f_x, f_y)$, giving a periodic pattern image $I(x, y)$. To avoid periodicity, we generate pattern larger than the desired width (say 110% larger) and cut the extra borders. The pattern intensities are then rescaled to have values ranging in [0:255]. Each pattern is finally binarized by the use of 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). This affects the number of required patterns, which is discussed next.

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 maximize the amount of information.

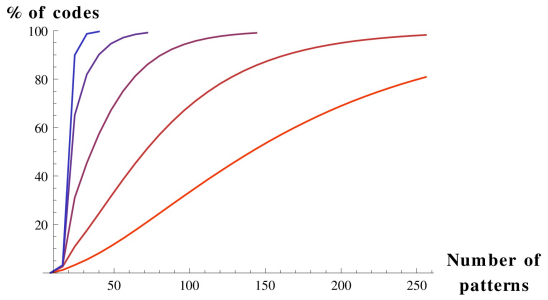
In Fig. 5(a), we measure the number of patterns required to disambiguate at least 99% of all pixels as frequency f is varied. We consider HD projectors having 1920×1080 pixels. One can see that low frequency noise require more patterns, Moreover, low frequency patterns often cause interreflections when large white pattern regions are projected in surface concavities and/or highly reflective materials.

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

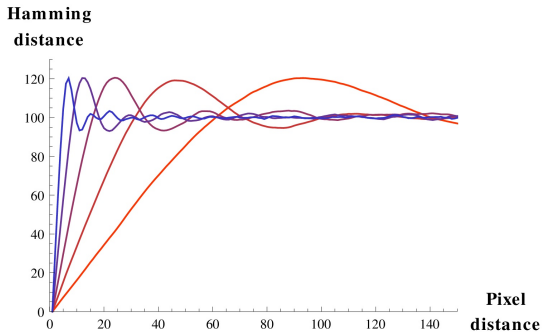
4.2. Neighborhood code similarity

One important property of our patterns is the similarity between neighboring codes. Fig. 5(b) presents the average 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.

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



(a)



(b)

Figure 5. (a) For HD images, the percentage of pixels having unique codes while increasing the number of patterns. The curves correspond to f ranging from 8 to 128 with curves on the right corresponding to patterns of lower frequencies. Curves stop being drawn if they reached 99%. (b) Average hamming distance between a pixel and its neighbors with increasing distance, for a code length of 200 patterns. 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).

provides robustness to various local imaging problems like defocussing or low pixel contrast.

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 half the bits are the same in both codes and will be accurately detected, making these two codes (and their neighbors) much more likely to be matched than any other code. In contrast, if the detected bits of two blended Gray code patterns are not all from the same code, then the resulting code may be completely unrelated.

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. This makes the matching 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.

A number of random unstructured light patterns are generated at a preselected band-pass frequency interval. Those patterns are projected, one at a time while a camera observe the scene. A number of patterns N are projected, captured by the camera, and then matched. Note that the number of patterns can also be increased simultaneously to the matching process until a desired quality has been obtained.

We first convert the gray images captured by the camera to 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[2], 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 ambient lighting due to interreflections.

Thus, as codes from *unstructured* light patterns have no correlation to projector pixel position, the correspondence problem is reduced to 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 $1920 \times 1080 = 2$ 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).

Matching is achieved efficiently using a high-dimensional search method based on hashing of binary vectors as described in [2, 8, 3]. 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 in practice some level of noise. 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 key has no bit errors can be modeled as:

$$P = 1 - (1 - (1 - \rho)^b)^k \quad (1)$$

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

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

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, and pixel ratio may also change, in the presence of slanted surfaces for instance. In practice, ρ is unknown and would need to be estimated to get a good indicator of how many iterations are required. The estimation of ρ is not addressed in this paper however. Matching is simply performed until no significant changes are detected.

Fig. 7(a) shows how adding code errors affects the convergence. We used $N = 200$ noise patterns and flipped bits according to different values of ρ. For instance, the best match should on average have an 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.

In Sec. 5, we introduce matching heuristics that improve convergence considerably (see Fig. 7(b)). However, (near)-optimal matches do not guaranty quality matches. For instance, when $\rho = 0.3$ is used, good matches give errors that are not well separated from random codes ($\rho = 0.5$), distributed at about half the number of bits $\frac{N}{2}$.

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 chose first hashed code. Even if this strategy does not choose the closest code, the time gained can be used to perform another matching iteration. Typically, the execution time for one iteration is around one second when matching an HD camera to an HD projector, and the iteration time is doubled when applying the heuristics.

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 smoothing. Unfortunately, smoothing can introduce errors at discontinuities or wherever the assumption does not hold. In our case, we can use two simple heuristics to take advantage of scene smoothness to get a dramatic speedup in the matching convergence. One great advantage

[2]Photometric distortion is usually a linear transformation including but not limited to gamma factors, scene albedo, aperture [6].

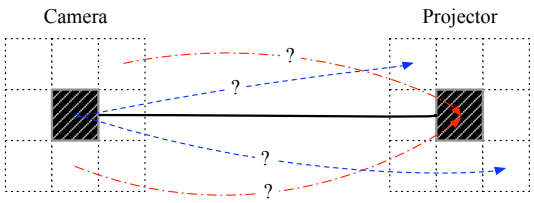


Figure 6. When a match is found (black solid line), two simple matching heuristics can be used : *forward matching* (blue dashed lines) attempts to improve an existing match and *backward matching* (red dot-dashed lines) attempts to improve neighborhood matches.

of our *smoothness* heuristics is that no degradation of the solution can ever occur, since they do not change the solution, only the time it takes to converge.

The heuristics are illustrated in Fig. 6. *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 current matches. *Backward matching* tests the neighborhood of a camera pixel to check if they could also match its corresponding projector pixel. This heuristic tends to create new matches. The speedup is shown in Fig. 7(a,b), where the convergence is plotted as a function of the number of iterations needed with and without the use of the heuristics.

6. Experiments

In this section, we present correspondence results on a real scene using unstructured light patterns, and compare to the results given by Gray codes as well as Phase-shift. We show results for a scene that contains significant interreflections, depth discontinuities and regions not in focus. Results on scenes having other types of global illumination effects such as sub-surface scattering and translucency are also available online [1].

The scene is composed of two walls, a floor and a ball (see Fig. 1). Results of all tested methods are shown in Fig. 9. On the left, our method is shown to give good results for high and low significant bits of the correspondence map. A frequency f of 128 cycles per image was used, as can be seen in the right image of Figure 4. In the middle, observe that Gray codes fail to recover highly significant bits on the floor near the walls because of indirect lighting. Phase-shift results are presented at the right of Fig. 9, without applying phase unwrapping. It also has difficulties near the walls and features a wavy matching typical of direct coding methods. This artifact gets worse when using a lower frequency.

Furthermore, we tested our method over a range of unstructured pattern frequencies. The results for selected regions are shown in Fig. 10. Notice that for regions not lighted directly, random codes should be expected. This

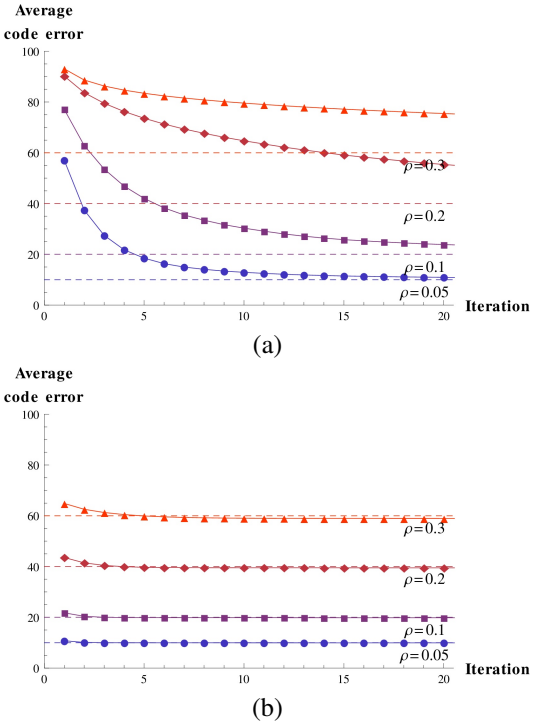


Figure 7. For noisy data, matching convergence may take several iterations. (a) Convergence of the hashing method with increasing noise. The dashed lines represent the lowest code errors the method should get on average (see text). (b) Convergence with increasing noise becomes much better when applying backward and forward matching heuristics at each iteration.

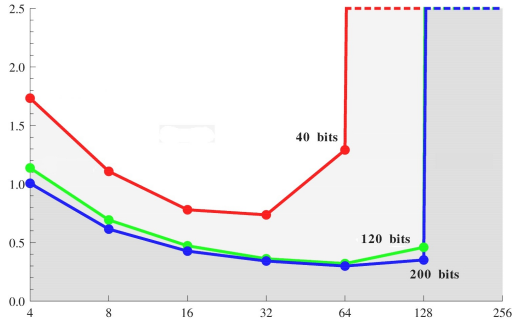


Figure 8. Average correspondence error as a function of pattern frequency (4,8,...,256), for various vector lengths (40,120 and 200 bits). Observe that more bits give lower errors. Low frequency patterns give slightly larger average errors because they required even more than 200 bits to disambiguate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

is indeed what is observed behind the ball when using high frequency patterns as in Fig. 10 (top right). High-frequency patterns also improve matching on the floor near the wall.

Finally, using the results of our method as ground truth, we measured errors by varying pattern frequencies and the number of patterns used. Fig. 8 shows that errors are smaller

with more patterns and middle frequencies. Low frequencies are not only unsuitable to reduce indirect lighting but they also require more patterns to disambiguate codes locally. Very high frequencies (here 256) would be ideal to reduce indirect lighting, but they are severely affected by aliasing due to camera-projector resolution. The problem of interreflection has been reduced to a problem of aliasing.

7. Conclusion

We presented a new approach to active reconstruction that uses patterns not based on 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 presented band-pass white noise patterns are designed to reduce scene interreflections and be robust to 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.

References

[1] http://vision3d.iro.umontreal.ca/en/projects/unstructured-light-scanning/. 6

[2] A. Andoni. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *IEEE Symposium on Foundations of Computer Science*, pages 459–468. IEEE Computer Society, 2006. 5

[3] A. Andoni and P. Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Communications of the ACM*, 51(1):117–122, 2008. 5

[4] K. Boyer and A. Kak. Color-encoded structured light for rapid active ranging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(1):14–28, January 1987. 2

[5] R. N. Bracewell. The fourier transform and its applications. 1965. 4

[6] D. Caspi, N. Kiryati, and J. Shamir. Range imaging with adaptive color structured light. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Jan 1998. 5

[7] J. Davis, D. Nehab, R. Ramamoorthi, and S. Rusinkiewicz. Spacetime stereo: A unifying framework for depth from triangulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 27(2):296–302, Feb. 2005. 1

[8] A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. In *VLDB '99: Proceedings of the 25th International Conference on Very Large Data Bases*, pages 518–529, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc. 5

[9] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. The lumigraph. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '96, pages 43–54, New York, NY, USA, 1996. ACM. 3

[10] J. Gühring. Dense 3-d surface acquisition by structured light using off-the-shelf components. *Videometrics and Optical Methods for 3D Shape Measurement*, January 2001. 2

[11] S. Inokuchi, K. Sato, and F. Matsuda. Range imaging system for 3-d object recognition. In *ICPR84*, pages 806–808, 1984. 2

[12] A. Kushnir and N. Kiryati. Shape from unstructured light. In *3DTV07*, pages 1–4, 2007. 1, 2

[13] M. Levoy and P. Hanrahan. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '96, pages 31–42, New York, NY, USA, 1996. ACM. 3

[14] S. K. Nayar, A. Krishnan, M. D. Grossberg, and R. Raskar. Fast separation of direct and global components of a scene using high frequency illumination. *ACM Transactions on Graphics*, 25:935–944, 2006. 1, 2, 3, 4

[15] P. Peers, D. K. Mahajan, B. Lamond, A. Ghosh, W. Matusik, R. Ramamoorthi, and P. Debevec. Compressive light transport sensing. volume 28, pages 3:1–3:18, New York, NY, USA, February 2009. ACM. 3

[16] J. Posdamer and M. Altschuler. Surface measurement by space-encoded projected beam systems. *Computer Graphics and Image Processing*, Jan 1982. 2

[17] M. Proesmans, L. Van Gool, and A. Oosterlinck. One-shot active 3d shape acquisition. *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, 3:336 – 340 vol.3, 1996. 2

[18] J. Salvi, J. Batlle, and E. Mouaddib. A robust-coded pattern projection for dynamic 3d scene measurement. *Pattern Recognition Letters*, January 1998. 2

[19] J. Salvi, J. Pagès, and J. Batlle. Pattern codification strategies in structured light systems. *Pattern Recognition*, 37:827–849, 2004. 2, 3

[20] D. Scharstein and R. Szeliski. High-accuracy stereo depth maps using structured light. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)2003*, volume 1, pages 195–202, June 2003. 2

[21] P. Vuylsteke and A. Oosterlinck. Range image acquisition with a single binary-encoded light pattern. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(2):148–164, February 1990. 2

[22] Y. Wexler, A. W. Fitzgibbon, and A. Zisserman. Learning epipolar geometry from image sequences. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2:209, 2003. 1

[23] C. Wust and D. Capson. Surface profile measurement using color fringe projection. *Machine Vision and Applications*, Jan 1991. 2

[24] L. Zhang, B. Curless, and S. Seitz. Rapid shape acquisition using color structured light and multi-pass dynamic programming. In *International Symposium on 3D Data Processing Visualization and Transmission. 3DPVT 2002*, pages 24 – 36, 2002. 2, 3

[25] S. Zhang and S. Yau. High-speed three-dimensional shape measurement system using a modified two-plus-one phase-shifting algorithm. *Optical Engineering*, Jan 2007. 2

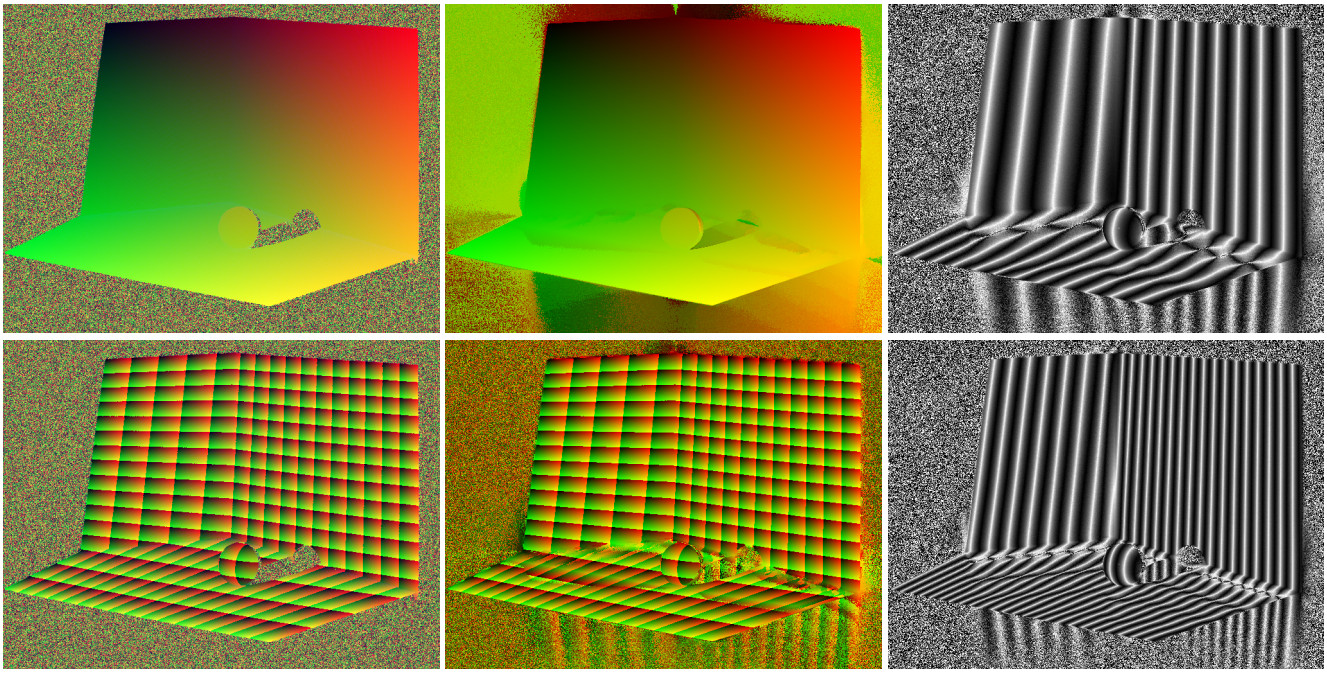


Figure 9. Results on our scene for all the methods tested : unstructured light patterns (left), Gray codes (middle) and Phase-shift (right). For the first two methods, results are shown with only the high significant bits of the correspondence map (top) and only the low significant bits (bottom). For Phase-shift, results are presented for different frequency on each row.

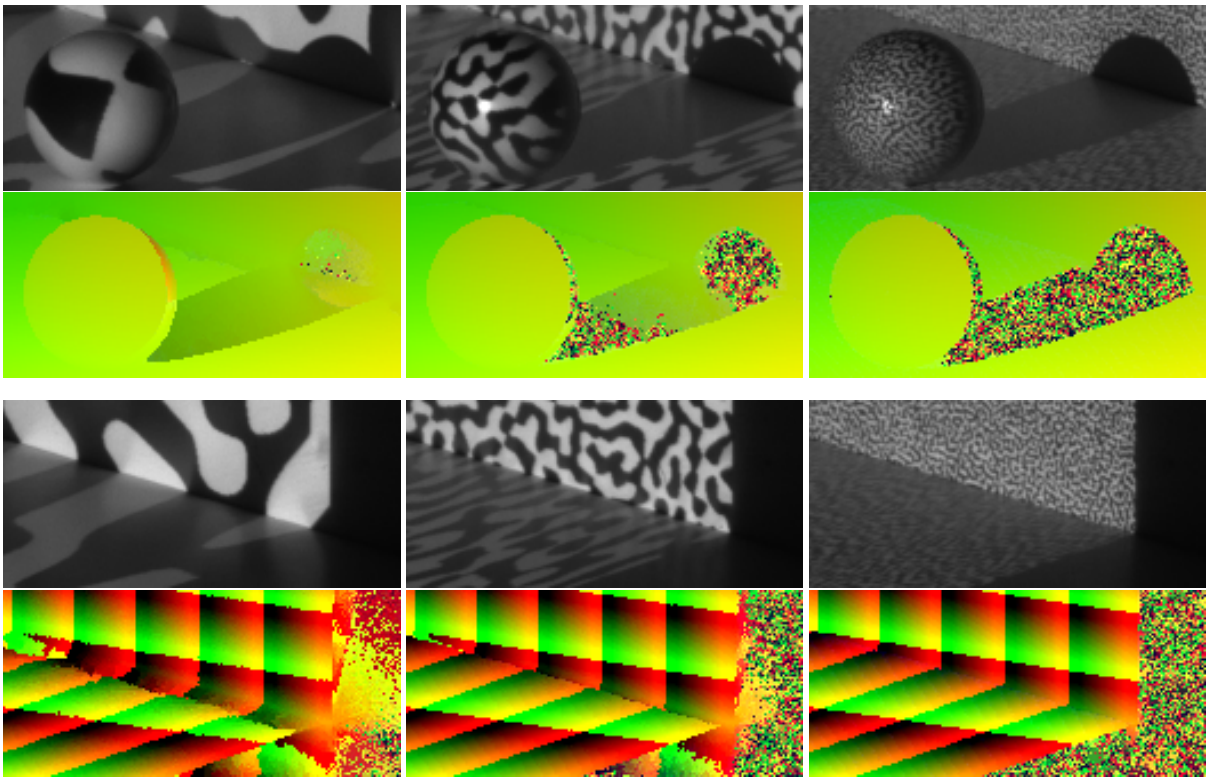


Figure 10. Correspondence from unstructured patterns at frequencies 8 (left), 32 (middle) and 128 (right). The effects of reducing indirect lighting using higher frequency patterns are exposed on the edge of the ball and its shadow (top), and the corner of the walls (bottom).