Hamming Color Code for Dense and Robust One-shot 3D Scanning

Shuntaro Yamazaki[1]
http://www.dh.aist.go.jp/~shun/
Akira Nukada[2]
nukada@matsulab.is.titech.ac.jp
Masaaki Mochimaru[1]
m-mochimaru@aist.go.jp

[1] National Institute of Advanced Industrial Science and Technology
Tokyo, Japan

[2] Tokyo Institute of Technology
Tokyo, Japan

Abstract

We propose a novel color code, Hamming color code, designed for rapid 3D shape acquisition using structured light projection. The Hamming color code has several properties which are desirable for practical 3D acquisition as follows. First, the Hamming distance of adjacent colors is always 1, which makes the color detection robust to color blending due to defocusing, subsurface scattering, or chromatic aberration. Second, the substrings of a certain length is guaranteed to be unique. In other words, the Hamming code can be viewed as a subset of de Bruijn sequence. Third, a one-dimensional coordinate can be encoded for each pixel, which enables dense 3D reconstruction from a single pattern projection. Thanks to the uniqueness and robustness of the substrings, the structured light can be decoded stably by dynamic programming. We have implemented parallel dynamic programming on GPU and achieved the speed-up by a factor of 630 compared to the CPU-based implementation, and accomplished video-rate 3D acquisition using commodity hardware. Several experiments have been conducted to demonstrate the stability and performance of our algorithm. Finally we discuss the limitation and future direction of this work.

1 Introduction

Real-time 3D shape acquisition is the fundamental problem in many applications of interactive techniques and real-world modeling. When an object of interest moves and deforms during the acquisition, the reconstruction has to be fast enough. The acquisition speed is particularly crucial for the active vision system that uses multiple image acquisition for each depth recovery. For fast imaging systems, temporally-coded active illumination has been used to recover the shape from acquired images. Many existing systems [4, 6] use a set of binary patterns and/or phase-shifted sinusoidal patterns and achieve accurate 3D shape acquisition.

When the camera speed is not fast enough, spatially varying illumination pattern is used to estimate the correspondence from a single image [2, 5, 8, 11]. This one-shot techniques, however, is unstable due to the inherent ambiguity in pattern correspondence. The reconstruction becomes erroneous when the a camera and a projector are not perfectly focused, or

BMVC 2011 http://dx.doi.org/10.5244/C.25.96

subsurface scattering occurs on the object. The computational cost of estimating correspondence is also higher than temporal coding approaches, which make it hard to accomplish a real-time scanning system using the one-shot algorithm.

In this paper, we propose a system of real-time 3D shape acquisition using a one-shot color structured light projection. Our contribution is twofold. First, we propose a novel color code which is robust to local color blending and depth discontinuity. Our color code is based on a pseudo-random number sequence similar in spirit to the de Bruijn sequence. In addition to the uniqueness of substrings, it is designed to have robustness to the blending of adjacent colors which often occurs in real image acquisition due to defocus and subsurface scattering. Second, we present a parallel implementation of dynamic programming on GPU, by which interactive 3D shape reconstruction is achieved using commodity optical components and computers.

2 Related Work

The strategies of structured light patterns are comprehensively surveyed and systematized by Salvi et al. [7] Thanks to epipolar constraint, the coordinates of 3D points can be estimated by searching for one-dimensional correspondence between projector and camera. Using color is a common approach to encoding the one-dimensional coordinates into a single projection pattern.

One of the simple patterns is a color gradient (or rainbow pattern) [9]. Robustness to the inhomogeneous surface color can be improved by using pseudo-random number sequences. Hügli and Gilbert Maître use a one-dimensional color stripe based on de Bruijn sequence composed of unique substrings [3]. Zhang et al. proposed a different color stripe where the de Bruijn sequence is embedded into color transition. The one-dimensional correspondence problem can be recovered efficiently using dynamic programming [11]. In order to consider the smoothness between epipolar lines, a two-dimensional grid pattern rather than one-dimensional is projected, and global optimization methods, such as belief propagation [10], graph cut [8], and linear system [2], are used.

There is a fundamental tradeoff between the discriminability of code words and the robustness to noise. With the patterns which use many colors and all pixels in the image [9], the correspondence can be uniquely determined, in the case of zero noise. On the other hand, sparse grid pattens [2, 8, 10] are inherently ambiguous and require elaborate optimization for decoding, but the localization of each pattern is robust to small perturbation of observed intensity.

We attempt to maximize both discriminability and robustness. Our work is inspired by the method proposed by Zhang et al. [11] As we will discuss in the following sections, their method has several drawbacks that are often presented in practical situations. We propose a novel color code that can be decoded as efficiently as their code by one-dimensional dynamic programming, while the robustness to depth discontinuity and local color blending is considerably improved.

3 Designing Color Code

We consider a one-dimensional color code used for structured light projection, assuming the projector and cameras are calibrated and the epipolar constraint can be incorporated.

For practical application of one-shot 3D scanning, the following requirements should be considered.

- **Discriminability:** The code words are discriminative, which guarantees the unique correspondence without such postprocessing as phase unwrapping or global optimization.
- **Robustness to local color blending:** The code words maintain the discriminability even when it undergoes local color blending caused by chromatic aberration, defocusing, or subsurface scattering.
- **Robustness to discontinuity:** The code words maintain the discriminability even when the pattern undergoes discontinuities caused by occlusion or insufficient reflectance.
- **Density:** The code word is embedded into a single pixel in a color pattern image to make the best of projector resolution.

The discriminability can be maximized by encoding the one-dimensional coordinate in 24-bit RGB color for each pixel. This approach, however, is prone to erroneous decoding, because the image intensities observed by a camera differ from those projected from a projector, due to shading, texture, subsurface scattering and other distortion in optical components. To deal with the issue of intensity modulation, we design our color code based on a pseudo-random number sequence, assuming that the amount of intensity modulation is relatively small and the adjacency of color stripe is preserved locally in acquired images.

3.1 Pseudo-random number sequence

De Bruijn sequence has been widely used for one-shot 3D scanning using structured light projection. The sequence guarantees the discriminability of the subsequence of a certain length, which enables unique identification from a partial observation. Zhang et al. [11] propose a color stripe code based on the de Bruijn sequence embedded into color transition. Their code has two interesting properties. First, the code has theoretically optimal discriminability for the de Bruijn sequence. Second, sub-pixel accuracy can be achieved by localizing the one-dimensional decoding can be performed at sub-pixel accuracy by localizing the color edges.

In practice, however, their method is prone to erroneous correspondence and unbearably poor 3D reconstruction, for the following two reasons. First, an acquired image often undergoes imaging noise such as chromatic aberration and defocusing, resulting in a color sequence which is different from the one projected. Second, the projected color code can be partially shifted, occluded or duplicated due to the depth discontinuity in the scene. The color edges generated by the depth discontinuity result in the fake color transitions which do not exist in the projected pattern. Even worse, the magnitude of the color transition at the depth boundary is usually more than those observed between the adjacent colors on a smooth surface. Although the color edge can be localized in subpixel accuracy, the disadvantage of fake correspondence at every object boundary is crucial in many practical applications, as shown in the left column of Figure 1.

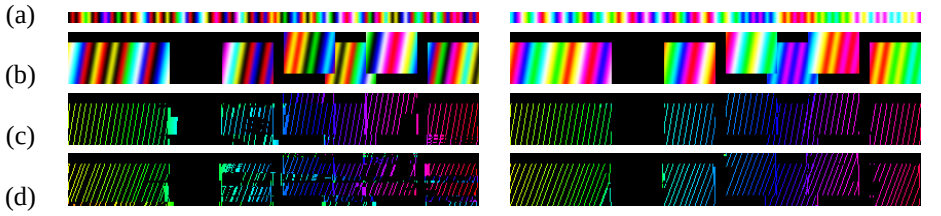


Figure 1: Comparison of color code based on pseudo-random number sequence. (a) One-dimensional color patterns. (b) Synthetic images of light projection. The color pattern is slightly rotated and partly displaced into horizontal direction to simulate depth discontinuities. The images are also blurred to simulate the defocusing effect of a camera and a projector, and sub-surface scattering on object surfaces. (c) Correspondences recovered from the images in (b). Correspondence is represented in color. Hamming color code is robust to image blur and depth discontinuities, while de Bruijn code embedded in color transition [11] causes considerable errors. (d) Correspondences recovered from the images that undergo one pixel of chromatic aberration.

3.2 Hamming Color Code

To remedy the first drawback mentioned in the previous section, we propose a novel pseudo-number sequence which is similar in spirit to de Bruijn sequence. Our sequence is specially designed for the use of color pattern for structured light projection. In addition to the uniqueness of substrings in de Bruijn sequence, our color code has the following property.

- **Robustness to blending:** Linear blending of two adjacent code words does not yeild other code words.
- **Truly Colored:** The stripe is composed of non-black colors.

The first property is accomplished by representing an RGB color by a 3-bit number and generating a color sequence so that the hamming distance between adjacent colors is always one. Thus, we call the code as **Hamming color code**. The first property guarantees that, even when two stripes are blended at their boundary, the resultant color matches either of the two colors. The second property is introduced from a practical viewpoint and can be regarded as optional. The color pattern proposed in [11] has black stripes. It is impossible to distinguish unilluminated holes and surface illuminated in black, which prohibits some postprocessing such as hole filling.

We encode one-dimensional coordinate into a color, instead of color transition, to alleviate the second drawback mentioned in the previous section. In decoding, correspondence is determined at the center of each color band detected by a gradient operator. We are currently investigating the combination of the Hamming color code with other coding scheme such as a luminance-modulated pattern propose by Pagès et al. [5] The Hamming color code is constructed in exhaustive counting, in contrast to the mathematical algorithm of constructing de Bruijn sequence. The bottom-up algorithm of generating the Hamming color code remains future work.

We generate the Hamming color code using the RGB colors except black, whose channel values are either 0 (minimum) or 1 (maximum). The elements in the code are represented by 3-bit RGB colors, or equivalently by 1-digit number as $\{1,2,3,4,5,6,7\}$. The sequence

$H(4)$ =

6457326751376457646231546267323137515457546754513264515731576237626464545464673131515131323262623237573767575767673737 5

Figure 2: Hamming color code. (top) Code sequence $H(4)$ used in our experiments. The sequence starts with 6, contains 1 to 7, and consists of distinctive substrings of size 4. (bottom) Color structured light pattern.

is characterized by a single parameter, the order of the sequence k which is the length of unique subsequence in the sequence. The sequence starts with any of the elements, and the following digits are generated by flipping one of the color channel in the previous color. If the generated number is zero or the last k digit has already included in the sequence, the number is discarded and other bits are flipped. The order-k Hamming color code $H(k)$ is the longest sequence generated by repeating the above algorithm. Apparently $H(k)$ is not unique, but any of the sequence can be used for our purpose. We use $H(4)$ of length 119 shown in Figure 2.

3.3 Solving Correspondence

The correspondence between projector pixel to camera pixels are determined by matching colors. The matching is solved by dynamic programming as proposed in the prior work. [11] The brief overview of the algorithm is reviewed as follows.

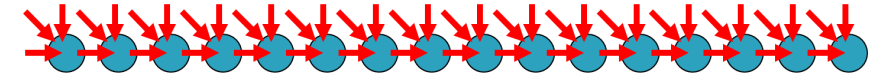
1. Normalize pixel colors in an acquired image.
2. Classify the colors into seven code words using normalized correlation.
3. Detect the peaks in each color band.
4. Solve correspondences by Dynamic Programming.
5. Discard correspondences included in less than k consecutive correspondence.
6. Update the location of the peaks with sub-pixel accuracy.

4 GPU-accelerated Dynamic Programming

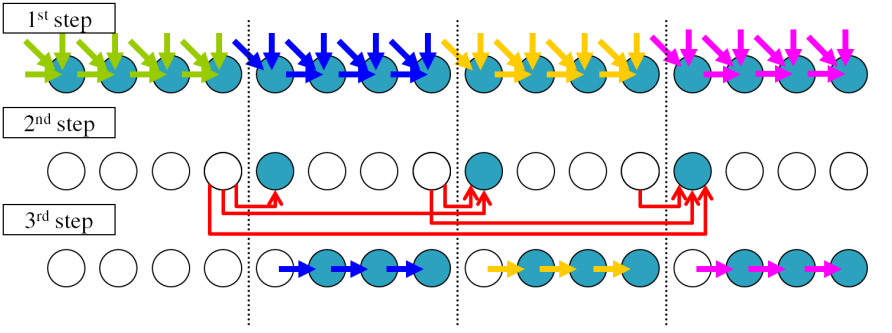
We solve the correspondence for each scanline in the acquired image by dynamic programming. While the global optimal solution can be found by the dynamic programming, it is not fast enough to achieve video-rate reconstruction on a standard PC platform. This issue is resolved by using GPU to parallelize the computation.

We implemented a standard DP matching algorithm using CUDA. Computations on CUDA-capable GPU require two-level parallelism. CUDA kernel launches several thread blocks which solve a dynamic programming corresponding to a horizontal line. The problem size is $w \times l$ where w is the width of acquired images, and l is the length of color code. Each thread block consists of multiple threads (32 threads in our case). Those threads run on eight Streaming Processors (SPs) and have access to the fast on-chip shared memory. So as to utilize all SPs, we have to parallelize each dynamic programming as described in Figure 3.

Since the correlation matrix is pre-generated and stored in the device memory, this computation is extremely memory bound enough to allows redundant computations to exploit



(a) Original Dynamic Programming



(b) Parallel Dynamic Programming

Figure 3: Parallel Dynamic Programming on GPU: (a) In conventional dynamic programming, the result of each subproblem is sequentially propagated to the following subproblems to find the global optimal solution. (b) Our dynamic programming partitions the subproblems into multiple blocks, each of which is solved parallelly in three steps: (1) Each block is optimized by solving conventional dynamic programming. (2) The solution of the block is sequentially propagated to the following blocks. (3) The solution of each block is updated taking account of the propagation in the step 2.

higher parallelism. The original dynamic programming has strong dependency in horizontal direction which prevents parallelization. To do this using multiple threads, we first divide them into multiple sections. In the 1st step, each thread computes the local optimum score while the dependencies between the sections are ignored. The dependencies between the sections are updated in the 2nd step using shared memory, and each thread updates its own section again in the 3rd step.

In addition to the parallelism within scanline, each scanline can be processed independently. Due to the nature of algorithm, the performance scales nearly linearly to the number of GPU cores.

5 Experiments Result

We have implemented the abovementioned algorithm and constructed a projector-camera system for 3D shape acquisition. The acquisition system is composed of a single data projector (EPSON EMP-765, 1024×768 resolution), a single USB camera (Artray ARTCAM 300MI, 1024×768 resolution), and a PC server of Intel Core2 Duo 2GHz and NVidia GeForce 480 GTX graphics hardware. The projector and camera are geometrically calibrated using OpenCV calibration toolkit. The radiometric response of the projector and camera are also calibrated in advance and the color crosstalk is minimized by linear optimization [1]. The reconstruction algorithm is implemented by C++ for CPU and by NVdia CUDA for GPU.

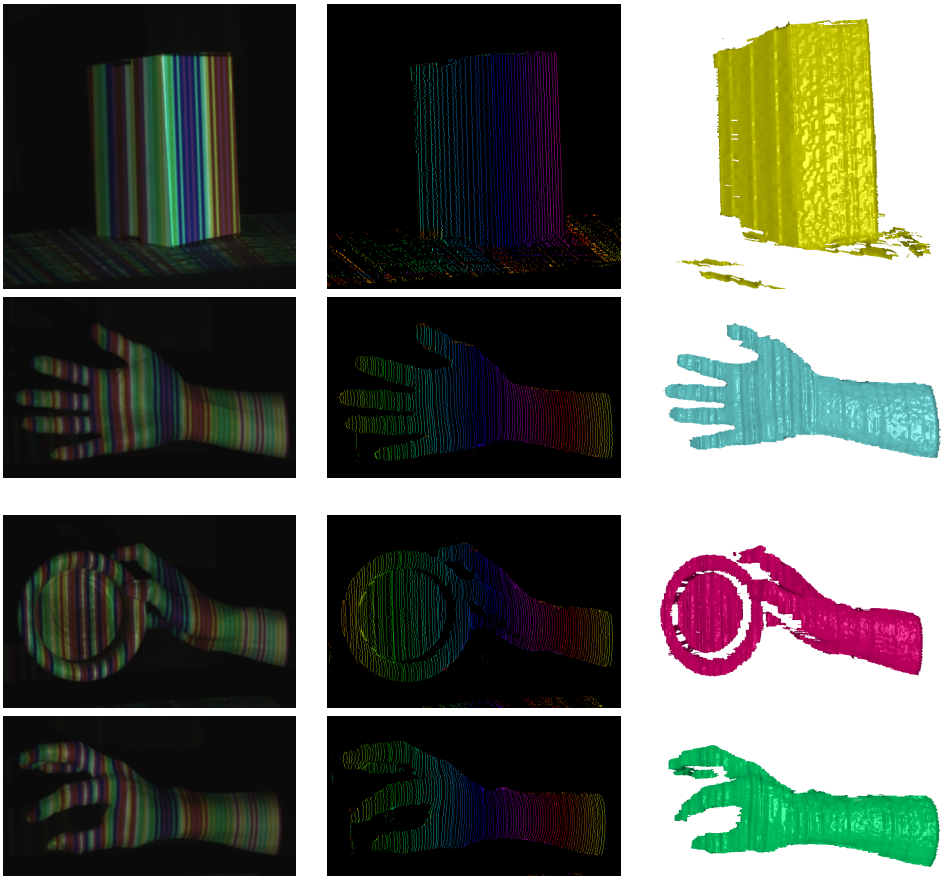


Figure 4: Result of 3D shape reconstruction: (left) Acquired images. (middle) Estimated correspondences shown in color. (right) Surface rendering of reconstructed 3D shapes.

Figure 4 summarizes the result of 3D shape reconstruction. The 3D shapes are reconstructed by our GPU software running at a video-frame rate. We do not remove the background and use all pixels for computing correspondences. Image rows are processed independently, and no inter-scanline smoothness is considered. No postprocessing such as noise reduction has been applied. We can see two kinds of artifacts in the reconstruction results. First is the jittering effects across the scanlines. Second is systematic vertical patterns on the 3D surface. The first problem can be drastically improved by considering inter-scanline smoothness by using additional optimization such as dynamic programming in vertical direction or belief propagation. The second problem is caused by the quantization in our correspondence. In dynamic programming, the correspondence is limited to pixel coordinate. We can improve the correspondence using the gradient of pixel intensity, which we are now working on.

Table 1 shows the computation speed of our 3D reconstruction. We show the timing of dynamic programming since our focus is on estimating correspondence from an acquired image to a projected color pattern. The overall speed-up of our GPU implementation is

Table 1: Computation time of Dynamic programming: This table shows the average time of 100 dynamic programming on CPU and GPU in milliseconds. Images are captured into the main memory on PC, then uploaded to the device memory on GPU. After the computation, correspondence is downloaded back from the device memory to the main memory.

	CPU	1 GPU	2 GPUs
Data upload	—	0.286	0.285
Color normalization	5.1	0.097	0.053
Consistency computation	809.6	8.533	5.130
Table update	12948.4	11.756	5.892
Backtrack	13.5	0.817	0.440
Data download	—	0.376	0.340
Total	13776.6	21.865	12.140

about 630 times against CPU implementation, which CPU software is not fully optimized. We can also see that the GPU implementation scales nearly linearly without any changes to the implementation. The frame rate of our 3D reconstruction system, including image acquisition, triangulation, and rendering, is about 60 Hz which we believe is sufficient for many graphics and vision applications.

6 Summary

We have proposed a novel color code, the Hamming color code for robust and efficient structured light projection. We have also presented the GPU implementation of dynamic programming which is about 630 times faster than CPU implementation and sufficient for interactive 3D rendering. Several experiments using both synthetic and real images demonstrate that our method is of practical use. Since our current implementation is just a proof of concept, there are many extension to improve the accuracy. Among them are inter-scanline smoothness, subpixel matching, adaptive thresholding for background removal. The color structured light becomes unstable when the object has color and textures on the surface. We are investigating the technique of adaptive color normalization which is necessary in practical applications.

References

[1] Dalit Caspi, Nahum Kiryati, and Joseph Shamir. Range imaging with adaptive color structured light. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20 (5):470–480, 1998.

[2] Ryo Furukawa and Hiroshi Kawasaki. Geometrical constraint based 3D reconstruction using implicit coplanarities. In *Proc. the 18th British Machine Vision Conference*, pages 780–789, 2007.

[3] Heinz Hügli and Gilbert Maître. Generation and use of color pseudo random sequences for coding structured light in active ranging. In *Proc. Industrial Inspection*, volume 1010, pages 75–82, 1989.

[4] Srinivasa G Narasimhan, Sanjeev Jagannatha Koppal, and Shuntaro Yamazaki. Temporal dithering of illumination for fast active vision. In *Lecture Notes in Computer Science (European Conference on Computer Vision)*, volume 5305, pages 830–844, 2008.

[5] Jordi Pagès, Joaquim Salvi, and Josep Forest. A new optimised de bruijn coding strategy for structured light patterns. In *Proc. International Conference on Pattern Recognition*, volume 4, pages 284–287, 2004.

[6] Szymon Rusinkiewicz, Olaf Hall-Holt, and Marc Levoy. Real-time 3D model acquisition. In *Proc. SIGGRAPH*, pages 438–446, 2002.

[7] Joaquim Salvi, Jordi Pages, and Joan Batlle. Pattern codification strategies in structured light systems. *Pattern Recognition*, 37(4):827–849, 2004.

[8] Christoph Schmalz and Elli Angelopoulou. Robust single-shot structured light. In *Proc. Projector-Camera Systems*, pages 13–18, 2010.

[9] Johji Tajima and Masato Iwakawa. 3D data acquisition by rainbow range finder. In *Proc. International Conference on Pattern Recognition*, pages 309–313, 1990.

[10] Ali Osman Ulusoy, Fatih Calakli, and Gabriel Taubin. Robust one-shot 3d scanning using loopy belief propagation. In *Proc. CVPR Workshop on Applications of Computer Vision in Archaeology*, pages 15–22, 2010.

[11] Li Zhang, Brian Curless, and Steven M. Seitz. Rapid shape acquisition using color structured light and multi-pass dynamic programming. In *Proc. the 1st International Symposium on 3D Data Processing, Visualization, and Transmission*, pages 24–36, 2002.

Appendix

A List of Hamming Color Code

The table below shows example Hamming color codes for different *k*s. The Hamming color code is not unique. This table presents the longest code found by exhaustive numeration.

order	code length	code
1	7	5732645
2	19	7513764573267546231
3	45	645732675137645154623157626467673132373757545
4	119	64573267513764576462315462673231375154575467545132645157315762376264645454646731315151313232626232375737675757676737375
5	269	64573267513764573154623154573764623762313762675462673264576451375132673157323157646751546775457623751573137546732376757375451323132646264515451576264673133757673767645467646457513157513732623732326762326267676262623232373151313737331315151575464545757545454646467675