Motion-Aware Structured Light Using Spatio-Temporal Decodable Patterns

Yuichi Taguchi, Amit Agrawal, and Oncel Tuzel

Mitsubishi Electric Research Labs (MERL), Cambridge, MA, USA

Abstract. Single-shot structured light methods allow 3D reconstruction of dynamic scenes. However, such methods lose spatial resolution and perform poorly around depth discontinuities. Previous single-shot methods project the same pattern repeatedly; thereby spatial resolution is reduced even if the scene is static or has slowly moving parts. We present a structured light system using a sequence of shifted stripe patterns that is decodable both spatially and temporally. By default, our method allows single-shot 3D reconstruction with any of our projected patterns by using spatial windows. Moreover, the sequence is designed so as to progressively improve the reconstruction quality around depth discontinuities by using temporal windows.

Our method enables *motion-aware* reconstruction for each pixel: The best spatio-temporal window is automatically selected depending on the scene structure, motion, and the number of available images. This significantly reduces the number of pixels around discontinuities where depth cannot be recovered in traditional approaches. Our decoding scheme extends the adaptive window matching commonly used in stereo by incorporating temporal windows with 1D spatial windows. We demonstrate the advantages of our approach for a variety of scenarios including thin structures, dynamic scenes, and scenes containing both static and dynamic regions.

Keywords: Structured light, motion-aware 3D reconstruction, spatio-temporal decoding, adaptive window matching.

1 Introduction

Structured light (SL) based triangulation is one of the most reliable active techniques for shape measurement in computer vision. The correspondence problem in stereo vision is simplified by projecting known patterns from a projector, which are imaged using a camera. For each camera pixel, the corresponding projector row or column is obtained by decoding the captured patterns, followed by a ray-plane intersection to compute the 3D shape. Temporally coded patterns including Gray codes [1–3] are widely used to provide high quality reconstructions for static scenes.

Single-shot SL methods [4–15] project a single pattern that allows per-frame reconstruction, and thus can be used for dynamic scenes and deforming objects. However, such methods lose spatial resolution and perform poorly around depth

A. Fitzgibbon et al. (Eds.): ECCV 2012, Part V, LNCS 7576, pp. 832–845, 2012.

discontinuities (e.g., thin structures), since a spatial neighborhood is required to perform the decoding for each pixel. More importantly, previous single-shot methods project the *same* pattern repeatedly for each time frame. Even if the scene was static or if parts of the scene were slowly moving, previous methods will still lose spatial resolution as if the entire scene was dynamic. Thus, traditional single-shot approaches are *not* motion-aware.

In this paper, we present how to design a sequence of structured light patterns that is decodable both spatially and temporally. These patterns allow *motion-aware* 3D reconstruction in an automatic way (without any user interaction). For example, if the scene consisted of both dynamic and static/slowly moving parts, our design would automatically obtain better depth reconstruction on the static/slowly moving parts by using temporal neighborhood information.

Previous single-shot SL approaches that employ a 1D stripe pattern can be augmented to become motion-aware using our technique. In particular, we build upon the 1D color De Bruijn stripe pattern (P_1) proposed by Pagès et al. [4] for single-shot reconstruction. We show how to design a sequence of N patterns, $P_1, \ldots, P_N$, by appropriately shifting the symbols of P_1. The sequence of patterns is projected repeatedly on the scene: $P_1, \ldots, P_N, P_1, \ldots, P_N, \ldots$. By default, each of the projected patterns P_i allows single-shot reconstruction using spatial neighborhoods in a traditional manner, since it is a shifted version of P_1. In addition, the patterns are designed such that the size of spatial neighborhood decreases as the number of patterns increases, leading to per-pixel reconstruction using all N patterns. Since the spatio-temporal neighborhood is chosen automatically, smaller spatial windows are used for slowly moving objects (1 pixel spatial neighborhood for static scenes), leading to motion-aware reconstruction.

Previous single-shot SL approaches reconstructed depths at *sparse* feature points such as edges [4, 7], intensity peaks [4] of color stripes, and 2D grid points [9–11], by using complex mechanisms and heuristics for decoding. In contrast, we demonstrate how to apply the plane sweeping algorithm [16] to structured light decoding. Such techniques have been well-studied for computing depth maps using stereo/multi-view matching and allow us to compute *dense* depth maps. Our reconstruction algorithm generates multiple layered images by projecting each pattern onto several depth layers, computes matching scores with the captured images for several spatio-temporal neighborhoods at each layer, and selects the best depth layer that has the maximum score.

1.1 Contributions

Our paper has the following main contributions:

- We propose the concept of motion-aware structured light reconstruction, which allows tradeoff between the amount of motion and reconstruction quality for dynamic scenes.
- We show how to design a sequence of stripe structured light patterns that is decodable both spatially and temporally.
- We extend the plane sweeping algorithm for structured light decoding using adaptive spatio-temporal windows.

1.2 Related Work

For extensive review and classification of existing SL methods, we refer the reader to a recent survey paper by Salvi et al. [17]. Below we discuss related work in the area of single-shot SL.

Single-Shot Structured Light employs spatial multiplexing using both 1D and 2D patterns to allow decoding at each frame. Techniques such as [4, 7] project a 1D De Bruijn sequence having a *window uniqueness* property. This allows unique decoding if a small spatial window of symbols is detected around a pixel. Color stripe patterns are used to realize De Bruijn sequences, since more than two symbols are required. Examples of 2D patterns include grid patterns [9–11] and M-arrays and perfect sub-maps using various geometric shapes and colors [12–15]. Recently, Kinect has emerged as a low-cost 3D sensor for computer vision and human-computer interaction applications [18, 19]. Kinect projects an infrared random dot 2D pattern as the single-shot pattern, which is captured using an infrared camera. The matching is done per frame, and depth maps for a dynamic scene can be obtained in real time. However, the depth maps are noisy especially around depth discontinuities. All of the above methods project the *same* pattern for every frame, process each frame independently, and are not motion-aware as discussed earlier.

Izadi et al. [19] registered the depth maps obtained from Kinect and reconstructed a static 3D scene with higher quality compared to raw depth maps obtained from Kinect. Such a depth map fusion algorithm can be applied as post-processing on the output from our approach if the scene is static. However, our main focus in this paper is to improve the quality of individual depth maps for *dynamic* scenes.

Spatio-Temporal Decoding: Ishii et al. [20] presented SL patterns that are spatio-temporally decodable. However, their scheme requires *disconnected* windows of pixels for decoding, and thus does not reduce the effective size of spatial neighborhood as more patterns are used. Zhang et al. [7] also improved spatial resolution by shifting a single-shot color stripe pattern one pixel at a time and analyzing the temporal profile for each pixel [21, 22] using all the shifted patterns. The key difference with ours is that their approach is not hierarchical. It requires the entire scene to be static during the projection of all shifted patterns to reduce the spatial neighborhood to a single pixel. In contrast, the design of our patterns allows us to *selectively* use different number of frames (e.g., 1, 2, 4 or 8) at every pixel for decoding, even for dynamic scenes.

Flexible voxels [23] enable post-capture spatio-temporal resolution tradeoff for reconstructing a video depending on the motion of each pixel. Our approach is similar in spirit by allowing motion-aware depth reconstruction using SL systems.

Adaptive Window Matching: Spatio-temporal windows [24, 25] have been used for stereo processing to improve the matching quality. Our decoding scheme is motivated by previous work in this area. However, for stereo processing, the size of the window is typically fixed for every pixel, or regular box-shaped windows are used. In contrast, our goal is to select the optimal spatio-temporal window for each pixel that allows decoding. Note that based on our patterns, we

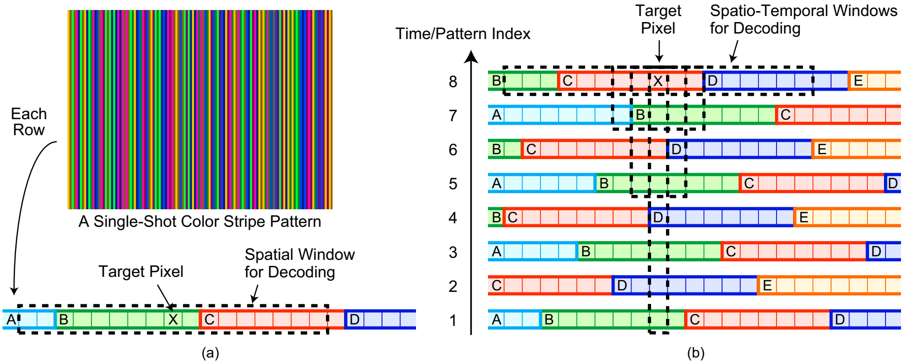

Fig. 1. Comparison between conventional 1D color stripe pattern and our approach. (a) Conventional color stripe pattern based on a De Bruijn sequence with window property $k = 3$ and stripe width $l = 8$. To uniquely determine the position at any pixel in each row, $k = 3$ symbols are required, resulting in a minimum spatial neighborhood of 17 pixels (8 pixels on left and right). (b) Our spatio-temporally decodable patterns generated by shifting the base color stripe pattern. The size of the spatial neighborhood window is reduced using temporal neighborhood. Note that each spatio-temporal window includes a unique set of symbols, ensuring unique decoding.

only evaluate those spatio-temporal windows with minimal window sizes that are sufficient for decoding, instead of regular box-like spatio-temporal windows.

2 Spatio-Temporal Decodable Patterns

We first describe the design of spatio-temporal decodable patterns for SL systems and then outline a reconstruction algorithm to choose the optimal spatio-temporal neighborhood for each pixel.

2.1 Background on Single-Shot Decoding

Similar to previous single-shot methods, we use a color stripe as our base pattern. De Bruijn sequences are typically used for designing the sequence of symbols (color of each stripe) to ensure the uniqueness of local spatial neighborhood windows. To uniquely determine the position in a De Bruijn sequence having the window property of k, we need to observe at least k consecutive symbols, encoded as colors in the pattern. Each stripe (symbol) is projected using a stripe width of l projector pixels to avoid color bleeding in projected images and to robustly detect the colors/edges of the stripes [4, 7].

For simplicity, let us assume a De Bruijn sequence with the window property $k = 3$ and the stripe width $l = 8$. As shown in [4], using 4 different colors and 2 different intensity levels, a De Bruijn sequence of length 128 can be obtained, which can be projected using $128 \times l = 1024$ pixels (sufficient for standard

Table 1. Spatial window size s required at each frame for decoding using t frames

Current Pattern	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8
$t = 1$	17	17	17	17	17	17	17	17
$t = 2$	$11^{\dagger}$	5	7	5	7	5	7	5
$t = 4$	5	3	5	3	5	3	5	3
$t = 8$	1	1	1	1	1	1	1	1

† It can be 7 if we continuously shift the pattern, instead of using the 8 patterns periodically.

projectors). Figure 1(a) shows such a sequence, where the triplets of symbols (A, B, C) and (B, C, D) are unique in the sequence.

Notice that for any target pixel, three unique symbols can be observed within $l = 8$ pixel left and right neighborhoods, i.e., in a spatial window of $2 \times 8 + 1 = 17$ pixels (in the projector image). In general, for the window property of length k and the stripe width l, a minimum spatial window of $2 \times l \times \lfloor k/2 \rfloor + 1$ pixels is required to ensure unique decoding at each pixel; if the sequence is broken due to depth discontinuities in this spatial window, the decoding cannot be performed. Thus, the reconstruction using a single color stripe pattern performs poorly around depth discontinuities and loses spatial resolution.

Our goal is to design a sequence of patterns such that the pixels from temporal neighborhood can be used for decoding to reduce the size of the spatial neighborhood.

2.2 Design of Pattern Sequence

The key idea is to arrange the transitions of symbols *hierarchically* in time by shifting a base pattern. In general, the number of patterns in the sequence required to enable pixel-wise decoding is equal to $\max(k, l)$. Figure 1(b) shows our spatio-temporally decodable patterns for $k = 3$ and $l = 8$. The eight patterns $P_i (i = 1, \ldots, 8)$ are generated by hierarchically shifting the base pattern P_1 with different shifts. Our pattern design is based on the following rules, where after each step we double the number of patterns.

1. Initialization: P_1 is set to the base pattern used.
2. Generate P_2 by shifting P_1 by $-\frac{3}{2}l = -12$ pixels.
3. Generate P_3 and P_4 by shifting P_1 and P_2 by $\frac{1}{4}l = 2$ pixels respectively.
4. Generate P_5 to P_8 by shifting P_1 to P_4 with $\frac{1}{4}l + 1 = 3$ pixels respectively.

We project P_1 to P_8 periodically. To decode a frame with a projected pattern P_i, we use previous frames with projected patterns $P_{i-1}, P_{i-2}, \ldots$ for spatio-temporal decoding.

Table 1 summarizes the spatial window size s required to observe unique triplets of symbols (since $k = 3$) at any target pixel at different times using t frames. Notice that the size of spatial neighborhood decreases as more temporal patterns are used. The spatio-temporal windows are not disjoint compared to [20]. In addition, decoding can be done for any pixel in *all* projected patterns.

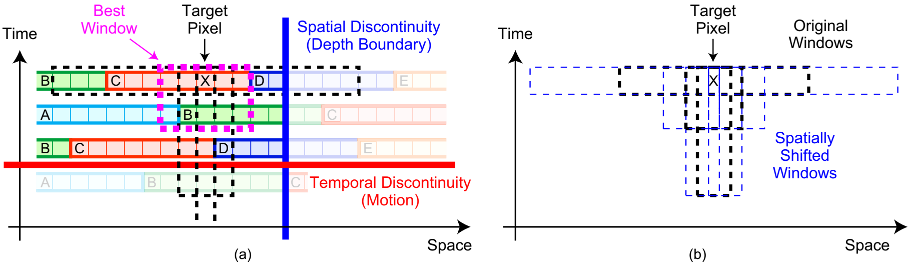
Time
Best Window
Target Pixel
Spatial Discontinuity (Depth Boundary)
B
C
X
D
E
A
B
C
B
C
D
E
A
B
C
Temporal Discontinuity (Motion)
Space
(a)
Time
Target Pixel
Original Windows
X
Spatially Shifted Windows
Space
(b)

Fig. 2. (a) Motion-aware spatio-temporal window selection. A spatially large window causes decoding error if the target pixel is located near a depth discontinuity, while a temporally large window causes error if the target pixel is not static. Our method selects the best window depending on the motion of the target pixel. (b) Spatially shiftable windows can be used to recover depth around depth discontinuities, if one of the shifted windows can be placed on a planar region. The figure shows windows shifted with $\pm\lfloor s/2 \rfloor$ pixels only for simplicity. In practice, we use a total of $2\lfloor s/2 \rfloor + 1$ windows shifted pixel-by-pixel within $\pm\lfloor s/2 \rfloor$ pixels.

If all eight patterns are used, then the spatial window size is only one pixel, allowing per-pixel reconstruction similar to temporally coded SL [1–3].

2.3 Motion-Aware Decoding

In the previous section, we described how different spatio-temporal windows can be used for decoding at any target pixel. However, what is the optimal spatio-temporal window to use for a given pixel?

Figure 2(a) depicts the principle of our motion-aware spatio-temporal window selection. A spatially large window causes decoding error if the target pixel is located near a depth discontinuity, while a temporally large window causes error if the target pixel is not static. An ideal window would have a small spatial support for enabling decoding near depth discontinuities, while aggregating temporal information if multiple frames are available and the target pixel is static or slowly moving.

Shiftable Window is a well-known technique for stereo matching [26, 27]. Suppose a target pixel is located on a planar region, but near a depth discontinuity. A spatial window centered at the target pixel would produce a decoding error, while spatial windows shifted left and right within $\pm\lfloor s/2 \rfloor$ pixels may avoid the discontinuity and can provide the correct disparity value. Note that we only need to use 1D spatial shifts, as shown in Figure 2(b), instead of 2D shifts typical in stereo matching.

In the next section, we present a simple algorithm that automatically selects the best window from the set of spatio-temporal windows depending on the context of a target pixel.

3 Reconstruction Algorithm

Single-shot SL systems using color stripe and 2D patterns typically reconstruct depths at sparse points such as edges between strips [4, 7], intensity peaks [4], or grid points [9–11]. Here we present a reconstruction algorithm that enables pixel-wise dense depth recovery based on the plane sweeping algorithm [16].

We define D depth layers with depth values $[d_i]_{i=1}^{D}$ in the camera coordinate system. Our goal is to find the best depth for each pixel in the camera. For each depth layer, we compute a matching score between the captured images and the patterns projected onto the depth layer using several spatio-temporal windows.

Let Q be the set of all spatio-temporal windows used for matching. Q depends on the number of available frames. Let $t_{\max}$ be the maximum temporal window size among all windows in Q. We refer to such decoding as $t_{\max}$-frame decoding. The matching score for a pixel $\mathbf{x}$ in the camera at depth d_i is defined as

$$S(\mathbf{x}, d_i) = \max_{q \in Q} \; w(q)\, S(q, \mathbf{x}, d_i), \tag{1}$$

where $S(q, \mathbf{x}, d_i)$ is the matching score using the spatio-temporal window q at the pixel $\mathbf{x}$ and the depth d_i, and $w(q)$ is the weight for the window. For scenes including large motions, we decrease the weight $w(q)$ according to the temporal window size t. To use shiftable windows, we simply augment Q by adding shifted versions of the spatio-temporal windows. We set the weight $w(q)$ for a shiftable window such that it decays linearly with the amount of the spatial shift to avoid strong staircase-like artifacts for slanted planes. Note that we need to determine the size of the spatial window depending on the size of the stripes in the captured images (not in the projector image as in Figure 1(b)). In experiments, we assume that the size of the stripes does not change considerably over the image.

Matching Score: We use normalized cross correlation (NCC) as the matching score. To efficiently compute the NCC score for several spatio-temporal windows with different sizes, we use 2D integral images [28]. The integral images for patterns projected onto depth layers can be computed offline; we only need to compute those for captured images online. In the results shown in the paper, we reconstruct the depth value for each pixel $\mathbf{x}$ by finding the the maximum score, $S_{\max} = \max_i S(\mathbf{x}, d_i)$, after smoothing the scores with a small local window (3×3). If the maximum score $S_{\max}$ is smaller than a threshold (0.8 in experiments), we mark the pixel as *unknown*, depicted with the black color in depth maps. Unknown pixels are caused by decoding errors and also include pixels in the occlusion (shadow) regions. Note that one can use global optimization techniques such as graph cuts or belief propagation [27] by using the scores $S(\mathbf{x}, d_i)$ as the data term. However, all our results show raw estimated depth values without any global optimization for fair comparisons.

4 Experiments and Results

We show extensive evaluation of our method on various scenes captured with a projector-camera pair. We projected 1024×768 pixel patterns using an NEC

LT170 projector and captured 800 × 600 pixel images using a Point Grey Flea2 camera. We performed geometric calibration between the camera and projector using checkerboards [29] and performed color calibration by computing the projector-camera coupling matrix [30] and an RGB offset due to ambient light.

As our base pattern, we used a De Bruijn sequence with color stripes as shown in Figure 1(a). Similar to Pagès et al. [4], we alternated the intensities of neighboring stripes with high (1) and low (0.5) values, while using 4 hues $(45°, 135°, 225°, 315°)$ to encode 8 symbols with the window property of 3.

We compare three different cases: (a) 1-frame decoding without shiftable windows (similar to traditional single-shot methods), (b) 1-frame decoding with shiftable windows, and (c) multi-frame decoding with or without shiftable windows. The goal of (b) is to demonstrate that our reconstruction algorithm can improve traditional single-shot methods around depth discontinuities. The goal of (c) is to demonstrate that our designed patterns achieve motion-aware reconstruction. Note that in all our experiments, t-frame decoding means that *up to* t frames are used for the spatio-temporal window selection, as defined in Section 3.

4.1 Static Scenes

We first show that our sequence of patterns allows similar per-pixel dense reconstruction as temporally coded SL systems on static scenes. As a reference, we use temporally coded Gray codes that maximize the minimum stripe width as proposed in [31]. These Gray codes perform better than traditional Gray codes in the presence of diffuse inter-reflections and other global illumination effects. We captured 20 images corresponding to the 10 Gray code patterns and their inverse patterns, and decoded the symbols as described in [31]. For our method, we captured only eight images using our eight patterns $P_1, \ldots, P_8$.

Reconstruction around Depth Discontinuities: Figure 3 shows reconstruction results for a piece-wise planar static scene. Firstly, notice that 1-frame decoding without using shiftable windows fails to recover pixels around depth discontinuities. This demonstrates the inherent limitations of previous single-shot methods at depth discontinuities. Secondly, by employing shiftable windows in the decoding process, pixels around depth discontinuities can be recovered using 1-frame decoding. Thus, our reconstruction algorithm can be directly used to improve previous single-shot methods. However, using shiftable windows with 1-frame decoding leads to staircase-like artifacts on the slanted planes, since this method tends to perform piece-wise smooth reconstruction. Finally, notice how our method improves the reconstruction accuracy around depth discontinuities as the number of frames used for decoding increases. Our 8-frame decoding provides accurate results similar to using temporally coded SL (using Gray codes).

Thin Structures: Figure 4 shows results on a scene including thin structures. As discussed earlier, pixels around thin structures and depth discontinuities cannot be recovered using 1-frame decoding due to the large size of spatial neighborhood required for decoding. Employing shiftable windows does not help in this case, since the depth discontinuities are not well-separated. In contrast,

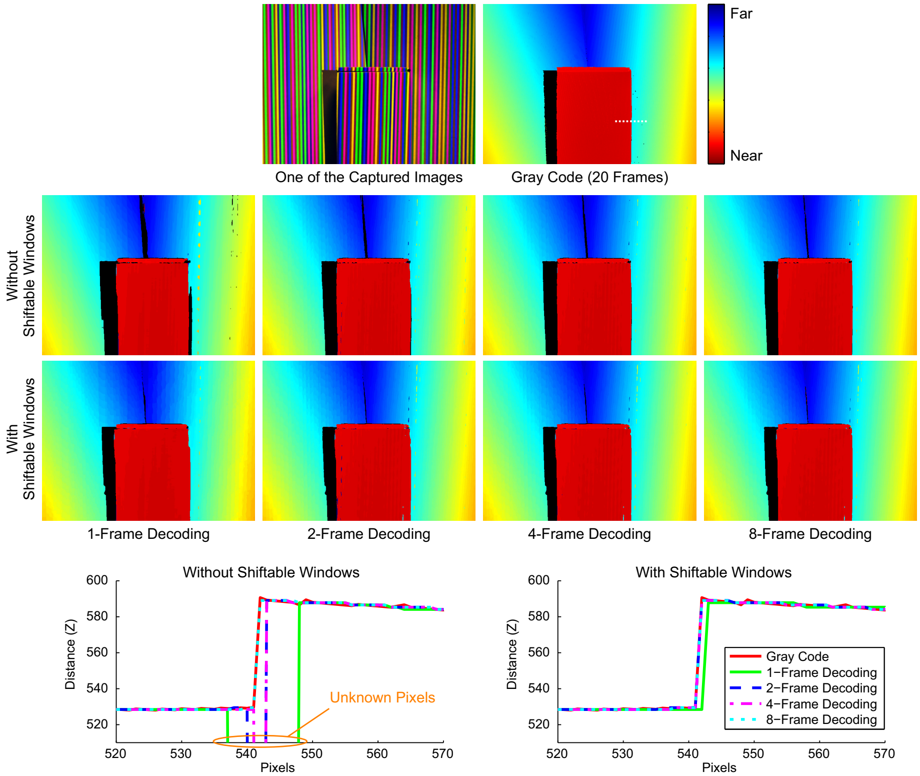

Fig. 3. Reconstruction results for a piece-wise planar static scene using different numbers of frames. Black pixels correspond to *unknown* pixels as described in Section 3. *First row*: One of the 8 captured images for our method and depth map reconstructed using Gray codes. *Second row*: Depth maps computed without shiftable windows. Pixels around depth discontinuities cannot be recovered using 1-frame decoding, while using multiple frames those pixels can be progressively recovered. *Third row*: Depth maps computed with shiftable windows. Shiftable windows allow better reconstruction using 1-frame decoding, but produce staircase-like artifacts on slanted planes. *Fourth row*: Distance profiles along the white dotted line for different methods.

our sequence of patterns improves the reconstruction accuracy as more frames become available using spatially small and temporally large windows.

4.2 Dynamic Scenes

Motion-Aware Reconstruction: Figure 5 illustrates motion-aware reconstruction for a scene including both static and dynamic regions. In this example, we pick objects one by one and capture a single image after removing each object. We reconstruct the depth map at each time instant by using the current and previous frames (up to 8 frames). For parts of the scene changed by the

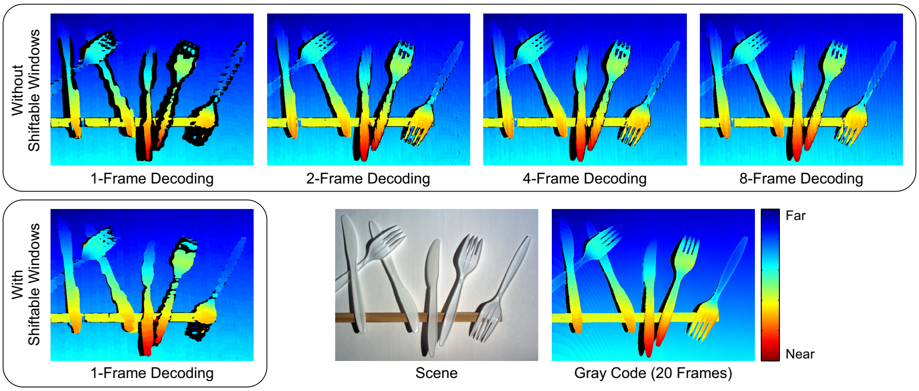

Fig. 4. Results for a static scene with thin structures using different numbers of frames. As more frames are used, the reconstruction quality around thin structures and depth discontinuities improves. Shiftable windows do not improve the reconstruction quality since the depth discontinuities are not well-separated spatially (e.g., rightmost fork).

picking, our method automatically selects a larger spatial window for decoding, recovering depth at a coarser resolution. However, for the other parts of the scene that remain static, the reconstruction accuracy improves as more frames become available (see the close ups in Figure 5). Thus, motion-aware reconstruction would be useful in applications such as robotic bin picking [32, 33].

Dynamic and Deforming Objects: Even for dynamic scenes, if a region remains static or is slowly moving for several consecutive frames, our method can improve the reconstruction accuracy. Figure 6 demonstrates this effect on a moving and deforming hand. Notice that even using 1-frame decoding, our reconstruction algorithm using shiftable windows produces better results than that without using shiftable windows. Using multiple frames, our method further improves the accuracy by using temporal information for regions that remained static or were slowing moving in previous frames. Notice the reduction in the number of *unknown* pixels around depth discontinuities that cannot be decoded as more frames are utilized. Compared to recent single-shot results using 2D patterns [15] (resolution of 150×100 on hand sequence), our results are at much higher resolution. Please see supplementary materials for video results.

5 Discussion and Conclusions

Limitations: Our algorithm assumes that in each spatio-temporal window, the local image patch is front-parallel and has similar reflectance property. Slanted planes, curved surfaces, and highly textured regions violate these assumptions. As we decrease the size of spatial window using more frames, our multi-frame approach can improve the reconstruction quality for these regions. Our technique also shares the limitations of color-based SL systems in handling colorful scenes: If a region does not reflect all the pattern colors, it cannot be decoded.

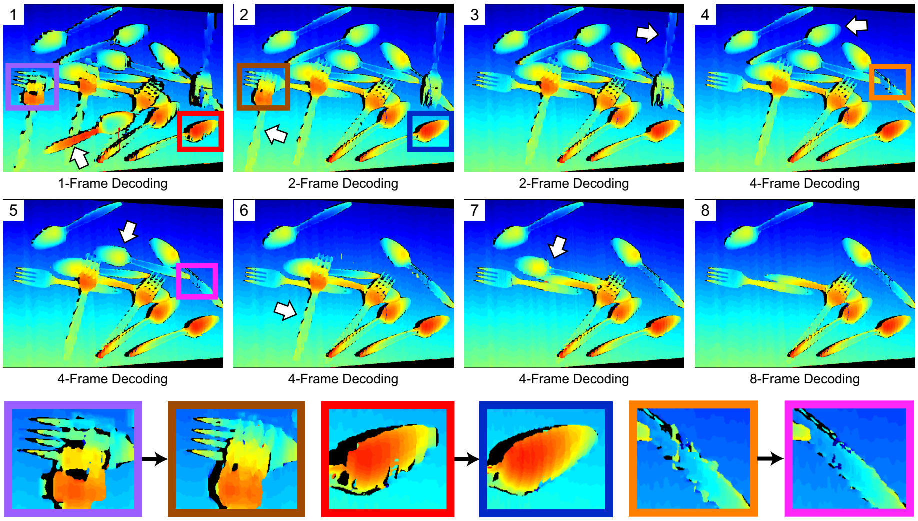

Fig. 5. Motion-aware reconstruction for a dynamic scene with static and moving parts. We captured a single projected pattern at each time, while removing a single object (shown by the arrow) after the capture. Thus, some parts of the scene remain static while others change dynamically. At each time, the depth map was reconstructed using the current and previous frames (up to 8 frames). In static regions, the reconstruction accuracy improves as more frames are used (see the close ups), while we can recover coarser depth for the dynamic regions using a larger spatial window.

Figure 7 shows results for a static colorful scene. Notice that the number of unknown pixels where depth cannot be estimated is substantially reduced using our multi-frame algorithm. However, regions with vibrant colors and dark regions do not reflect all the pattern colors. In those regions, decoding can fail even for multi-frame decoding algorithm as evident from Figure 7. Our approach can also fail in the presence of global illumination effects such as strong inter-reflections, and on objects with non-Lambertian BRDF (highlights, specularities, etc.).

Accuracy vs. Run Time: The accuracy of our reconstruction depends on the number of depth layers D, which causes the tradeoff between the accuracy and run time; if the depth variation of the scene is large, more depth layers are required to achieve similar depth resolution, increasing the run time. Typical single-shot reconstruction algorithms using color stripes [4, 7] do not have this tradeoff, because they find symbols at feature points (edges or intensity peaks) in each row of an image and use them for triangulation. Such algorithms compute depth values only for the detected feature points, while our algorithm computes pixel-wise dense depth maps. We set D to 60 for Figure 5 (a depth range of 525mm to 560mm), 80 for Figure 4 (520mm to 560mm), and 100 for Figures 3 (520mm to 640mm) and 6 (470mm to 565mm). On a standard PC with an Intel Core i7-950 processor, the NCC score computation for each depth layer took about 10 msec for 1-frame decoding and 70 msec for 8-frame decoding.

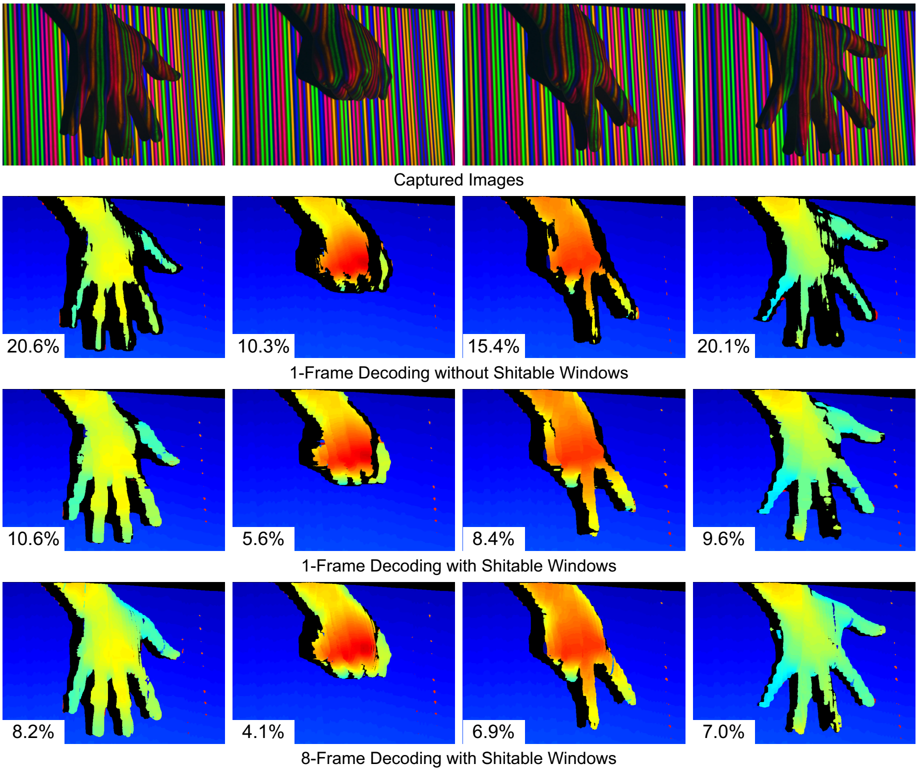
Captured Images
20.6%
10.3%
15.4%
20.1%
1-Frame Decoding without Shitable Windows
10.6%
5.6%
8.4%
9.6%
1-Frame Decoding with Shitable Windows
8.2%
4.1%
6.9%
7.0%
8-Frame Decoding with Shitable Windows

Fig. 6. Captured images and depth maps for several frames of a video of a deforming hand. The inset numbers show the percentages of unknown pixels computed over the entire image (excluding the top-right region, which is out of the field-of-view of the projector). For 1-frame decoding, shiftable windows improve reconstruction accuracy around depth discontinuities. Our motion-aware reconstruction algorithm using 8-frame decoding produces even better results by fusing temporal information for pixels that are static/slowly moving for several consecutive frames. Supplementary materials contain a video of the reconstructed depth maps.

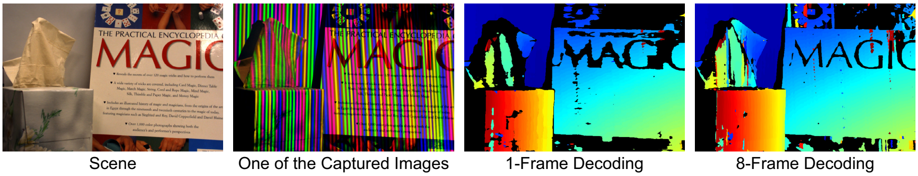

Fig. 7. Results for a colorful scene. Since color boundaries act as spatial discontinuities, decoding using a large spatial window fails to recover depth around color boundaries. By using a smaller spatial window with multiple frames, the reconstruction improves. However, even 8-frame decoding fails to recover depth for regions that do not reflect all projected colors. This limitation applies to all existing color-based SL systems.

Conclusions: We described a motion-aware structured light system that can handle dynamic scenes with different motions in different parts of the scene. Compared to traditional approaches, our system improves depth reconstruction for static/slowly moving parts of the scene and results in better estimation of depth discontinuities. We showed how structured light patterns that are decodable both spatially and temporally can be designed for motion-aware reconstruction. Our reconstruction algorithm extends stereo matching techniques using adaptive windows and can be easily implemented on GPU for faster processing.

Acknowledgments. We thank Jay Thornton, Gowri Somanath, Makito Seki, and Yukiyasu Domae for their helpful comments and support.

References

1. Posdamer, J., Altschuler, M.: Surface measurement by space-encoded projected beam systems. Computer Graphics and Image Processing 18, 1–17 (1982)
2. Inokuchi, S., Sato, K., Matsuda, F.: Range imaging system for 3-D object recognition. In: Proc. Int'l Conf. Pattern Recognition (ICPR), pp. 806–808 (1984)
3. Altschuler, M., Altschuler, B., Dijaki, J., Tamburino, L., Woolford, B.: Robot vision by encoded light beams. Three-Dimensional Machine Vision 87, 97–149 (1987)
4. Pagès, J., Salvi, J., Collewet, C., Forest, J.: Optimised De Bruijn patterns for one-shot shape acquisition. Image and Vision Computing 23, 707–720 (2005)
5. Hügli, H., Maïtre, G.: Generation and use of color pseudo random sequences for coding structured light in active ranging. In: Proc. SPIE Industrial Inspection, vol. 1010, pp. 75–82 (1989)
6. Forster, F., Rummel, P., Lang, M., Radig, B.: The HISCORE camera: A real time three dimensional and color camera. In: Proc. IEEE Int'l Conf. Image Processing (ICIP), vol. 2, pp. 598–601 (2001)
7. Zhang, L., Curless, B., Seitz, S.M.: Rapid shape acquisition using color structured light and multi-pass dynamic programming. In: Proc. Int'l Symp. 3D Data Processing, Visualization, and Transmission (3DPVT), pp. 24–36 (2002)
8. Koninckx, T., Van Gool, L.: Real-time range acquisition by adaptive structured light. IEEE Trans. Pattern Anal. Mach. Intell. 28, 432–445 (2006)
9. Salvi, J., Batlle, J., Mouaddib, E.: A robust-coded pattern projection for dynamic 3d scene measurement. Pattern Recognition Letters 19, 1055–1065 (1998)
10. Sagawa, R., Ota, Y., Yagi, Y., Furukawa, R., Asada, N., Kawasaki, H.: Dense 3D reconstruction method using a single pattern for fast moving object. In: Proc. IEEE Int'l Conf. Computer Vision (ICCV), pp. 1779–1786 (2009)
11. Ulusoy, A.O., Calakli, F., Taubin, G.: One-shot scanning using De Bruijn spaced grids. In: Proc. Int'l Conf. 3-D Digital Imaging and Modeling (3DIM) (2009)
12. Griffin, P., Narasimhan, L., Yee, S.: Generation of uniquely encoded light patterns for range data acquisition. Pattern Recognition 25, 609–616 (1992)
13. Morano, R., Ozturk, C., Conn, R., Dubin, S., Zietz, S., Nissanov, J.: Structured light using pseudorandom codes. IEEE Trans. Pattern Anal. Mach. Intell. 20, 322–327 (1998)
14. Claes, K.: Structured light adapted to control a robot arm. PhD thesis, K.U.Leuven (2008)

15. Maurice, X., Graebling, P., Doignon, C.: A pattern framework driven by the hamming distance for structured light-based reconstruction with a single image. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), pp. 2497–2504 (2011)
16. Collins, R.T.: A space-sweep approach to true multi-image matching. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), pp. 358–363 (1996)
17. Salvi, J., Fernandez, S., Pribanic, T., Llado, X.: A state of the art in structured light patterns for surface profilometry. Pattern Recognition 43, 2666–2680 (2010)
18. Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., Blake, A.: Real-time human pose recognition in parts from single depth images. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), pp. 1297–1304 (2011)
19. Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A., Fitzgibbon, A.: KinectFusion: Real-time 3D reconstruction and interaction using a moving depth camera. In: ACM Symp. User Interface Software and Technology, UIST (2011)
20. Ishii, I., Yamamoto, K., Doi, K., Tsuji, T.: High-speed 3D image acquisition using coded structured light projection. In: Proc. IEEE/RSJ Int'l Conf. Intelligent Robots and Systems (IROS), pp. 925–930 (2007)
21. Kanade, T., Gruss, A., Carley, L.R.: A very fast VLSI rangefinder. In: Proc. IEEE Int'l Conf. Robotics Automation (ICRA), vol. 2, pp. 1322–1329 (1991)
22. Curless, B., Levoy, M.: Better optical triangulation through spacetime analysis. In: Proc. IEEE Int'l Conf. Computer Vision (ICCV), pp. 987–994 (1995)
23. Gupta, M., Agrawal, A., Veeraraghavan, A., Narasimhan, S.G.: Flexible Voxels for Motion-Aware Videography. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part I. LNCS, vol. 6311, pp. 100–114. Springer, Heidelberg (2010)
24. Zhang, L., Curless, B., Seitz, S.M.: Spacetime stereo: Shape recovery for dynamic scenes. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), vol. 2, pp. 367–374 (2003)
25. Davis, J., Ramamoorthi, R., Rusinkiewicz, S.: Spacetime stereo: A unifying framework for depth from triangulation. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), vol. 2, pp. 359–366 (2003)
26. Kanade, T., Okutomi, M.: A stereo matching algorithm with an adaptive window: Theory and experiment. IEEE Trans. Pattern Anal. Mach. Intell. 16, 920–932 (1994)
27. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. Int'l J. Computer Vision 47, 7–42 (2002)
28. Lewis, J.P.: Fast normalized cross-correlation. Vision Interface 10 (1995)
29. Lanman, D., Taubin, G.: Build your own 3D scanner: 3D photograhy for beginners. In: ACM SIGGRAPH 2009 Courses, pp. 1–87 (2009)
30. Caspi, D., Kiryati, N., Shamir, J.: Range imaging with adaptive color structured light. IEEE Trans. Pattern Anal. Mach. Intell. 20, 470–480 (1998)
31. Gupta, M., Agrawal, A., Veeraraghavan, A., Narasimhan, S.G.: Structured light 3D scanning in the presence of global illumination. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), pp. 713–720 (2011)
32. Ikeuchi, K.: Generating an interpretation tree from a CAD model for 3D-object recognition in bin-picking tasks. Int'l J. Computer Vision 1, 145–165 (1987)
33. Choi, C., Taguchi, Y., Tuzel, O., Liu, M.Y., Ramalingam, S.: Voting-based pose estimation for robotic assembly using a 3D sensor. In: Proc. IEEE Int'l Conf. Robotics Automation (ICRA), pp. 1724–1731 (2012)