

Dense Depth and Color Acquisition of Repetitive Motions

Yi Xu

Daniel G. Aliaga

Department of Computer Science at Purdue University

{xu43|aliaga}@cs.purdue.edu

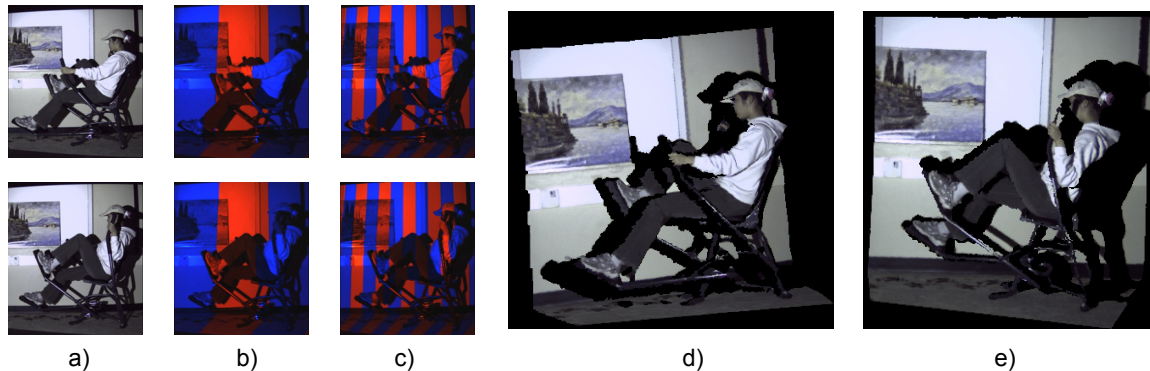


Figure 1. Acquiring repetitive motions. a) Example input video frames observing repetitive motions. b-c) Images showing two motion states under two projected patterns. d-e) After reconstruction, renderings for novel viewpoints can be generated at any sampled time instance of the motion.

Abstract

Modeling dynamic scenes is a challenging problem faced by applications such as digital content generation and motion analysis. Fast single-frame methods obtain sparse depth samples while multiple-frame methods often rely on the rigidity of the object to correspond a small number of consecutive shots for decoding the pattern by feature tracking. We present a novel structured-light acquisition method which can obtain dense depth and color samples for moving and deformable surfaces undergoing repetitive motions. Our key observation is that for repetitive motion, different views of the same motion state under different structured-light patterns can be corresponded together by image matching. These images densely encode an effectively “static” scene with time-multiplexed patterns that we can use for reconstruction of the time-varying scene. At the same time, color samples are reconstructed by matching images illuminated using white light to those using structured-light patterns. We demonstrate our approach using several real-world scenes.

1. Introduction

Acquiring dense depth and color samples of dynamic scenes is a challenging problem faced by applications such as digital content generation and motion analysis. In some dynamic environments, a subset of motion states repeat over time and form

repetitive (or quasi-periodic) motions, such as with some mechanical motions (e.g., toys, pendulum, etc.) and human activities (e.g., exercises, walking, etc.). In this paper, we present a robust method to densely capture both depth and color samples of such repetitively moving and deforming surfaces.

Many approaches have been proposed for capturing the geometry and appearance of dynamic scenes. On the one hand, passive methods are unobtrusive and simultaneously obtain color and depth information per pixel (e.g. [29]). However, they depend on fragile correspondence computations, color consistency over a large baseline (e.g. [24]), and/or background subtraction (e.g. [2][16]). On the other hand, active methods add energy into the scene in order to significantly improve robustness. But, laser scanners would require an excessively fast update rate for dynamic scenes. A traditional structured-light approach might be able to project a few patterns in rapid succession and assume the scene is temporarily stationary, but obtaining dense depth, as well as color information, is hard. Incrementally building a model might be possible but it usually assumes rigid motions and cooperation between scene and capture operator.

The main challenges in robustly acquiring dynamic scenes are three-fold. First, the method should obtain dense information, as expected with active methods, despite constant scene motion. Second, the approach should obtain both depth and color for each

sample. Third, capture should be able to sample moving and perhaps deforming objects.

Our key insight is that for scenes with repetitive motions temporally disjoint images capturing the same motion state but under different illumination patterns can be corresponded together. This allows using time-multiplexed codes to capture each motion state *without* the need to track pattern boundaries, *without* assuming a static scene during several contiguous frames, and *without* being limited to rigid scene motion. In this way, the dynamic scene acquisition problem is converted to a series of more robust static scene reconstructions. Furthermore, since the state of the moving scene can also be matched against fully-illuminated images, the color and texture of the moving scene can be recovered as well.

Our approach uses a geometrically and spectrally calibrated camera-projector pair to capture a scene containing repetitive motions (Figure 1). For acquisition, an all white image and a set of two-color Gray-code patterns are sequentially projected onto the scene. Each pattern lasts for a while. Image analysis is performed to find a set of motion-state images under white illumination that generate a smoothly changing and repeating image sequence and that can be well-matched against the images captured under the illumination of the two-color patterns. Matching is done using an image differencing operator that is calibrated to work with different colored light sources. Since the observed motion tends to repeat, eventually many motion states are sampled by all patterns, yielding the ability to reconstruct each motion state individually. The collection of reconstructed motion states can then be used to re-create a scene similar to the original or to produce new motion sequences. We demonstrate our approach using several real-world scenes rendered as colored points or as texture-mapped triangles.

Our main contributions include

- a dense depth and color acquisition system for scenes containing rigid and non-rigid objects undergoing repetitive motions,
- an algorithm for finding smoothly varying motion states spanning all illumination patterns, and
- a two-color structured-light method combined with a color-calibrated image matching operation to uniquely match images of a scene under different structured-light patterns.

2. Related work

Our approach relates to both active ranging of dynamic scenes and repetitive motion analysis. In the following, we summarize related work in these areas.

2.1 Dynamic scene acquisition

Active ranging techniques project patterns onto the scene and capture their projection to obtain range. Methods that only require “one-frame” are suitable for dynamic scenes. However, the pattern must encode as much information as possible to facilitate camera-camera (camera-projector) correspondences. Davies and Nixon [5] use windows of neighboring dots with unique color configurations. Chen et al. [4] use projected color stripes and stereo matching to correspond features. Zhang et al. [27] use de Bruijn illumination patterns and multi-pass dynamic programming. Fong and Buron [8] use a fixed pattern that combines color stripes and sinusoidal intensity changes. Koninckx et al. [13][14] improves performance by adjusting geometric and color-coded patterns to the scene. As opposed to our dense acquisitions, one-frame methods obtain reconstructions of limited density. Furthermore, while our method uses colored patterns, we only require two colors and two transitions; this makes color decoding much easier.

Some one-frame methods have partially remedied simultaneously projecting structured-light patterns and acquiring scene color. Frueh and Zakhor [9] use two cameras to capture color and low-resolution infra-red structured-light patterns. The Z-Cam [11] captures both color and low-resolution depth using a single infra-red “light pulse”. Waschbüsch et al. [25] use at least three cameras and a rapidly-alternating structured-light pattern and its inverse to acquire color and to enhance the scene with features for stereo matching. In contrast, in our setting we acquire color and structured-light patterns separately without having to quickly change patterns or use infra-red.

For slowly moving objects, multi-frame methods can be used. For example, Rusinkiewicz et al. [10][18] sample rigid and colorless objects by projecting and tracking at high frame-rate a small number of patterns where each stripe boundary has a unique code over four consecutive frames. The space-time stereo framework [28] can be extended to use projected patterns but dense correspondence is ultimately limited by feature-matching performance. In our method, temporally disjoint frames can be corresponded and no limit is placed on the code length. Moreover, we are not limited to rigid objects.

2.2 Repetitive motion analysis & acquisition

Analyzing and capturing repetitive motions in 2D and in 3D settings has gained significant interest from the computer graphics community and related fields in past years. Several research efforts have focused on detecting and measuring repetitive motions usually from image sequences (e.g., [6][15][22]). More

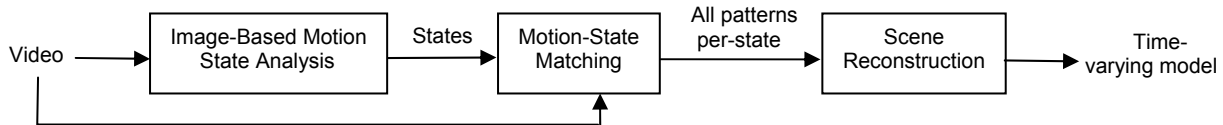


Figure 2. System pipeline.

recently, Schödl et al. introduced the concept of a 2D video texture [21], and Agarwala et al. presented its panoramic video extension [1]. Both produce an infinitely-long varying video sequence. However, these efforts focus on analyzing 2D images and/or do not obtain dense depth and color data.

Several passive 3D methods have been proposed as well for repetitive motions. For example, Starck et al. [23] capture human motion from a multiple viewpoint video sequence and use a motion graph to generate transitions of different motions in order to synthesize novel animation sequences. Einarsson et al. [7] acquire a 7D time-varying reflectance field of constant-speed human locomotion on a treadmill, but no explicit geometry is reconstructed. In addition, our captured motion does not need to be of constant speed. Instead, we find a set of best matches for each motion state by comparing images. Our early work [26] acquires low-resolution volumetric representations of table-top objects undergoing repetitive motions by using two cameras. In that work, we use a static camera to determine the motion state and a constantly moving second camera at known position to capture the motion from multiple viewpoints. By assuming constant illumination, motion state can be determined by simple image matching. In this paper, the scene is illuminated by varying structured-light patterns, producing robust correspondences, and requiring a robust color-calibrated image difference operation.

3. Acquisition

Our method captures a single video sequence and sequentially projects white light, a set of colored Gray-code patterns [12], and their inverses. We choose to use time-multiplexed patterns due to high accuracy in the encoding/decoding process. A repeating motion sequence from the video captured under white-light is found via image analysis. Images capturing the same motion state under different structured-light patterns are grouped together enabling the reconstruction of the dynamic scene. Figure 2 contains our system pipeline.

3.1 Image-based motion states

Our system chooses from among the images under white-light illumination a compact subset of image-based motion states. We desire images that form a smoothly varying repetitive motion (in similar spirit to [21]), and represent motion states that repeat while the structured-light patterns are projected in the rest of the

video. Our method is based on the following two assumptions: (1) scene motion can be decomposed into states that repeat over time and (2) each motion state can be uniquely identified by its projection onto the camera’s image plane. The observed motion does *not* need to be periodic and does *not* need to be of constant speed. Rather, our approach automatically extracts the motion states that can be assembled to form a smoothly repeating sequence and can be used for scene reconstruction. The resulting motion sequence will be similar to the original but not necessarily the same.

Our method uses a two-step algorithm. The input video has a total of $(N+2PN)$ images where P equals the number of patterns and N is the number of images captured under the illumination of each pattern. More precisely, the video contains N white-light images and, for each of P patterns, the video has N images for the pattern and N images for its inverse. Although we capture equal-length pattern segments, they are not synchronized with the motion in anyway. In a first step, our method computes a ranking of the N white-light images based on their similarity to images in the subsequent pattern segments. The ranking will give preference to motion states that tend to repeat and will downplay spurious motions. A complete ranking of white-light images is computed by comparing each white-light image to each pattern image. As shown in Figure 3, the comparison for each white-light image produces an array of length P where each entry contains the index of the most similar image from each pattern segment. The list of arrays is sorted based on the average similarity between the grouped white-light images and pattern images.

In a second step, our method matches the highest-ranked white-light images to a chosen white-light video segment for reconstruction. Our method allows the user either to choose an automatically computed looping segment of the input video or to manually specify a (new) desired image/motion sequence. To automatically compute a best looping segment, our system computes a 2D image similarity matrix. The entries along the diagonal correspond to image self-similarity while the typically sparse off-diagonal entries correspond to strong similarities between distinct images at the beginning/end of a looping segment. The algorithm compares the matrix against a user-specified minimum similarity threshold which results in only a few possible repeating motion

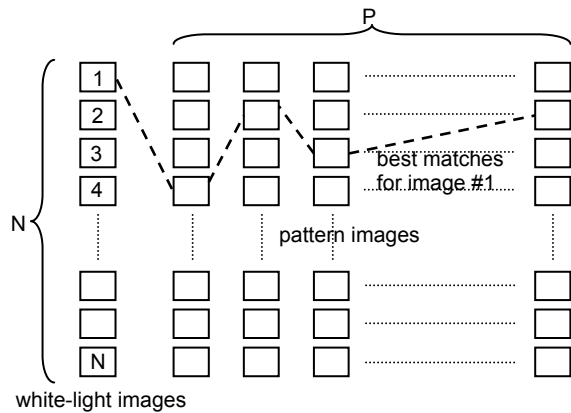


Figure 3. Motion states. For each white-light image, we find an array of the best-matched pattern images. White-light images are ranked based on the average similarity to their corresponding best pattern images. Then, the highest-ranked white-light images and their matched pattern images are used to reconstruct a chosen video segment.

sequences with small transitions errors. The system or the user chooses the best or any one for reconstruction.

Once the looping white-light segment is selected, our system maps all other white-light images, within a threshold of image similarity, to these selected images. This clustering process has the effect of grouping together white-light images observing the same motion state (as well as their associated pattern images) and thereby increasing the chance of finding a good set of pattern images to reconstruct the motion state. The improvement of reconstruction quality comes at the price of less faithfully reproducing the specified looping segment. From the multiple images per cluster, the one with the smallest difference between pattern images and corresponding white-light image is chosen; thus, the reconstruction quality is maximized. The resulting (image-based) motion states are the ones used for reconstruction and are stored as the time-varying model. In the next two sections, we describe how the pattern images are best matched to the white-light images.

3.2 Two-color structured light

Our system matches pattern images to the white-light images of the motion states. Our assumption that motion states eventually repeat enables projecting each structured-light pattern for a relatively long amount of time (namely, until all states have been observed at least once). This prevents having to rapidly change illumination patterns and omits needing camera-projector synchronization. However, it necessitates robustly matching the motion-state images (under white-light) to the pattern images that are illuminated by structured-light patterns. As opposed to methods which use multiple colors to reduce the total number of patterns [3] or to improve robustness [17], our approach uses two-color patterns to enable accurately matching pattern images to white-light images. It is worth noting that in many structured-light methods for dynamic scenes only the images illuminated by the patterns can be used to obtain color information. In our case, we extract color from the white-light motion-state images and thus the coloring of the patterns does not affect the color quality of the final reconstruction.

Conventional Gray-code binary stripe patterns rely on large illumination variance between stripes. However, this is precisely the opposite of what is needed for matching to homogeneously-illuminated white-light images. In particular, binary stripe patterns typically consist of white and black stripes. Any motion in the non-illuminated (black) areas would be invisible to the camera or appear too dark for robust processing (Figure 4a-c). The problem is worse during the initial stripe patterns of Gray codes because of the large contiguous regions and, unfortunately, these are exactly the patterns corresponding to the most-significant bits of the Gray code sequence. The total amount of ambiguity is, however, scene and motion dependent.

For our objectives, this ambiguity affects both motion-state matching and decoding process for scene reconstruction (i.e., determining if a pixel is red or blue). Having less illuminated pixels clearly places more burden on the image comparison metric to

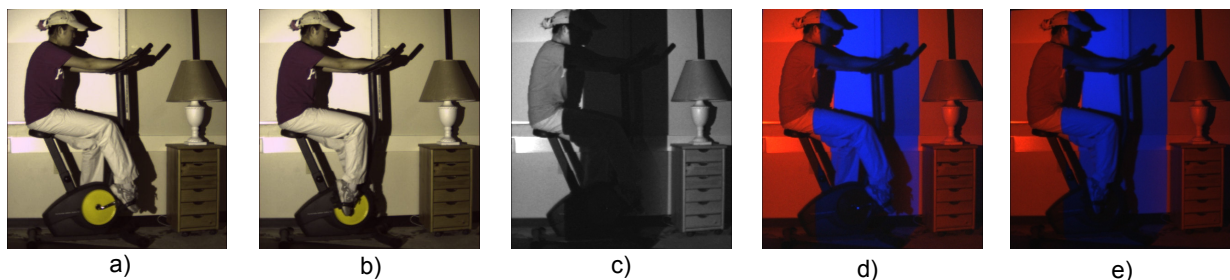


Figure 4. Two-color structured light. a-b) Two example motion states. c) Using conventional white-black stripes, both states seem identical. d-e) Our calibrated two-color structured light discerns between the two states.

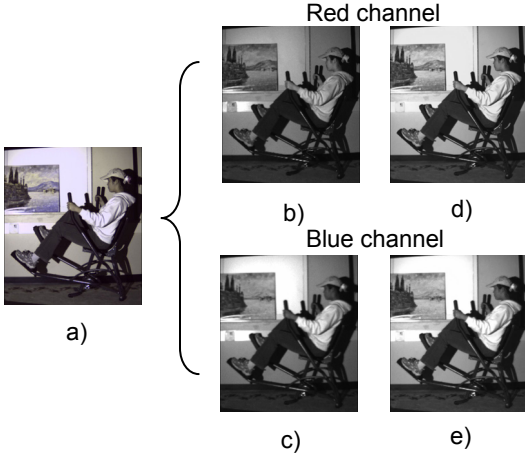


Figure 5. Color calibration for image matching. a) Image of scene under white light after white balancing. b-c) Channels of the original white-light image. d-e) Observed channels under red/blue light only. After calibration, images in (b-c) become virtually identical to (d-e).

correctly identify the motion state. Moreover, even if a best match is found it does not guarantee the correct motion state has been identified. Thus, when the decoding process is performed to reconstruct a particular motion state, the assumption of all patterns observing the same motion state (i.e., a static scene) would be violated. This might even occur between the projection of a pattern and its inverse (a commonly used method to improve robustness against unknown surface albedo and global illumination effects [20]). In general, this leads to pixels being inconsistently labeled amongst the pattern images and thus producing fewer correctly decoded pixels.

In our system, we use two visible colors (i.e., not black) to represent the equivalent of a binary stripe pattern (Figure 4d-e). The colored pattern pixels are compared to the pixels in the corresponding channel of the white-light motion-state image. However, the spectral performance of the camera/projector is not the same under color-stripe illumination and white-light illumination which leads us to the following section.

3.3 Color-calibrated image matching

To compare color-pattern images to white-light images, we use a color-calibrated image matching procedure (Figure 5). Based on the standard color spectrum, we expect the red and blue color channels to offer little or no overlap and thus we use red and blue color stripes. We fix the shutter speed, white balance, and adjust gain to prevent overflow. A smaller gain is used for capturing white-light images. The white-light illuminated motion-state images encode the response of the scene to red and blue light. However, in general

the response is not the same as red-only or blue-only light. This is due to the various projector and camera technologies which use different strategies to project or capture color.

We estimate the spectral response of the camera to projector light using a triple of calibration images. We capture a scene (either the actual scene with no motion or a similar one-time calibration scene) with red-only, blue-only, and white projector light. Then, we compute scale factors for the red channel and for the blue channel of the white-light image that minimize the difference between the single-channel images and the multi-channel image. The scale values depend on the scene and on the actual hardware used.

For motion state matching, our method computes a normalized image difference. The stronger color channel (r or b) of each pixel p of a pattern image is subtracted from the corresponded and scaled pixel q of the white-light image. Thus, given a predefined threshold t , per-pixel image difference is defined to be

$$d(p, q) = \begin{cases} |p_r - q_r| & p_r > p_b + t \\ |p_b - q_b| & p_b > p_r + t \\ \text{undefined} & \text{otherwise} \end{cases}$$

4. Reconstruction

Scene reconstruction is performed only on selected states, each of which has a complete set of pattern images. The two-color patterns are used in a similar fashion to standard binary structured-light methods. To better account for global illumination effects and unknown surface albedo, we project Gray code patterns and their inverses. Same-code pixel clusters that span too much image area are culled. Corresponded points whose camera and projector rays do not pass close enough are culled. Points floating in space without nearby points are removed as well.

5. Implementation details

We implemented our system in C/C++ using a PC with a 3.6GHz CPU and 2GB memory. Our camera is a Point Grey Dragonfly Express camera capturing 640x480 resolution color images at up to 200Hz. The fast shutter speed of the camera helps reduce motion blur. However, our system only stores images to disk at a rate of 30Hz. To project patterns, we use a Canon Realis SX6 LCOS projector with a resolution of 1280x1024 and three explicit LCD panels so as minimize spectral overlap. Color calibration computed that the camera's average scale values to make white-light images match separately-illuminated images are 1.398 and 1.043 for red and blue, respectively. The projector and camera are geometrically calibrated by the method of Sadlo et al. [19]. Our system projects a total of 32 stripe patterns (8 horizontal, 8 vertical, and

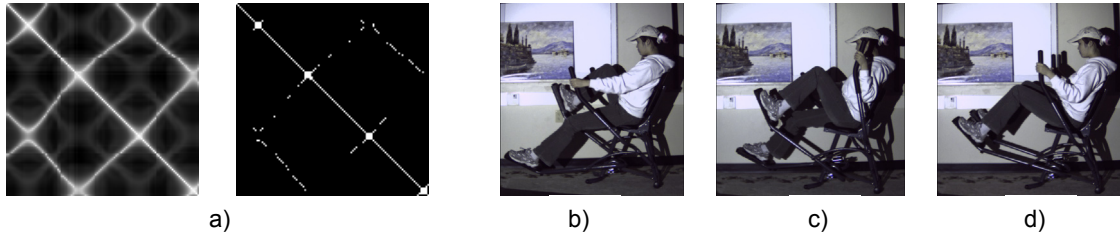


Figure 6. Image-based motion states. a) Partial motion-state inter-similarity matrix (left) and the thresholded one (right). b-d) Example motion states of the Cruiser dataset.

their inverses). All image comparisons are done on the CPU at full-resolution.

6. Experimental results

We have captured three different scenes to test our method: *Bicycle*, *Cruiser*, and *Exerciser*. Details and statistics are in Table 1. For all datasets, we reconstruct the best repeating sequence obtained via the thresholded similarity matrix of the white-light images. As an example, Figure 6 shows part of the original similarity matrix, thresholded matrix, and several motion states of the Cruiser dataset.

Given the desired sequence for reconstruction, our method clusters the white-light images as described in Section 3.1. Figure 7 illustrates the tradeoff that results from different clustering thresholds. The horizontal axis represents the amount of clustering performed between white-light images. The vertical axis represents the matching error between pattern images and white-light images. The units of both axes are normalized per-pixel error. Ideally, we desire a solution at $y=0$ (i.e., zero motion difference between the pattern images and the white-light images) and $x=0$ (i.e., no clustering, thus exactly each frame of the desired looping segment is reconstructed). However, except for perfectly periodic motion (which results in a horizontal line at $y=0$), this is impossible. Thus, we seek for a “sweet spot” that produces a balanced trade-off of the two errors. In general, this occurs at the point on the graph closest to $(x=0, y=0)$. For our datasets, this occurs in the clustering range $[0.03, 0.06]$ and is the range we use.

Figure 8 compares the results of using our red-blue patterns for the Bicycle dataset to only using a

Dataset	No. Points	Time/No. Images	No. States	Compute Time
Bicycle	33000	5.5 mins/9900	34	3.5 hrs
Cruiser	33200	4.6 mins/8250	53	3.2 hrs
Exerciser	34000	2.7 mins/4900	20	0.6 hrs

Table 1. Dataset statistics. Time refers to length of recorded video, number of states refers to number of unique images in the best repeating sequence.

single color for the subset of the image surrounding the motion (e.g., using “red” and assuming “blue” corresponds to black). The graph shows an advantage in our favor of about 10% – the benefit is of course motion and scene dependent. Pictorial examples of failures were previously shown in Figure 4. Once the motion states and patterns have been matched, scene reconstruction can proceed. Figure 9a shows a virtual viewer observing the Bicycle dataset using colored points. Figures 9b-c contain renderings from novel viewpoints of our other datasets using texture-mapped triangles with “skins” removed automatically. Figures 9d-f contain the corresponding depth maps. Our acquisition provides freedom to change the viewpoint, to change the time instance, and to generate reordered motion sequences. Thus, Figures 10a-c show a temporal sequence from a novel but fixed viewpoint for the Cruiser dataset. Figures 10d-f show a moving viewpoint at a fixed time using our Bicycle dataset.

Regarding limitations, our system cannot faithfully recover dark or green-only surfaces. However, in a typical scene most surfaces are a mixture of colors and are rarely pure green. Our approach is also limited to in-place repetitive motions. We look to feature tracking and image warping as a means to re-center translating objects. Finally, motion capture research focuses on capturing moving humans.

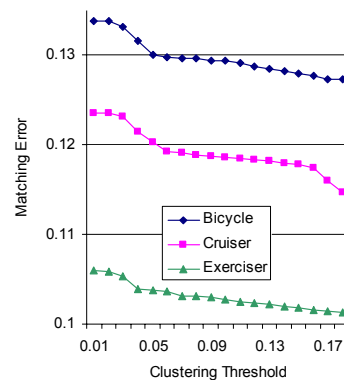


Figure 7. Clustering trade-off. We show the tradeoff between clustering and matching error.

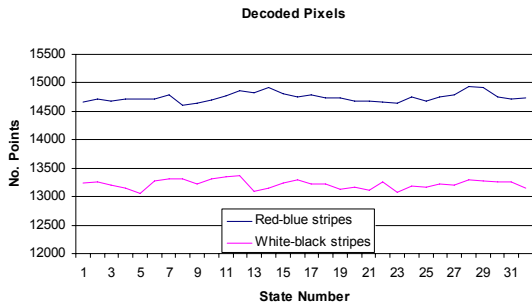


Figure 8. Comparison of decoded pixels. Because of better state matching, our two-color structured light patterns decode more pixels.

But, such approaches often require *a priori* models of the human body and do not actually recover a 3D model but rather parameters of the joints of an existing model. While we do demonstrate our method with humans, we make no special assumptions. In fact the deformations that occur (e.g., bending of clothing) actually make the problem more difficult because we cannot assume rigid motions.

7. Conclusions and future work

We have presented a robust method to densely capture both depth and color samples of dynamic scenes containing repetitively moving and deforming surfaces. Our demonstrations indicate that with video sequences of only a few minutes in length and by automated processing, high quality reconstructions of a time-varying model can be obtained. By converting a dynamic scene problem into a series of more robust static scene reconstructions, our method achieves high density by using the more accurate time-multiplexed

codes (as opposed to single-frame codes).

There are several avenues of future work. In particular, we would like to use graphics hardware to accelerate our image processing, thus reducing processing time, to extend our method to a multi-viewpoint capture, and to pursue automatic adaptation. Thus, as capture proceeds the system should advance to the next pattern as soon as possible.

Acknowledgement

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

References

- [1] A. Agarwala, K.C. Zheng, C. Pal, M. Agrawala, M. Cohen, B. Curless, D. Salesin, and R. Szeliski, "Panoramic Video Textures", *ACM Trans. on Graphics*, 24, 3, 2005, pp. 821-827.
- [2] J. Carranza, C. Theobalt, M.A. Magnor, and H.-P. Seidel, "Free-Viewpoint Video of Human Actors", *ACM Trans. on Graphics*, 22, 3, 2003, pp. 569 - 577.
- [3] D. Caspi, N. Kiryati, J. Shamir, "Range Imaging with Adaptive Color Structured Light", *IEEE PAMI*, 20, 5, 1998, pp. 470 - 480.
- [4] C.-S. Chen, Y.-P. Hung, C.-C. Chiang, and J.-L. Wu, "Range Data Acquisition using Color Structured Lighting and Stereo Vision", *Image and Vision Comp.*, 15, 6, 1997, pp. 445-456.
- [5] C.J. Davies, and M.S. Nixon, "A Hough Transform for Detecting the Location and Orientation of Three-Dimensional Surfaces via Color Encoded Spots", *IEEE Trans. on Systems, Man and Cybernetics*, 28, 1, 1998, pp. 90 - 95.
- [6] R. Cutler and L. Davis, "Robust Real-Time Periodic Motion Detection, Analysis, and Applications", *IEEE Trans. on PAM.*, 22, 8, 2000, pp. 781-796.
- [7] P. Einarsson, C.-F. Chabert, A. Jones, W.-C. Ma, B. Lamond, T. Hawkins, M. Bolas, S. Sylwan, and P. Debevec, "Relighting

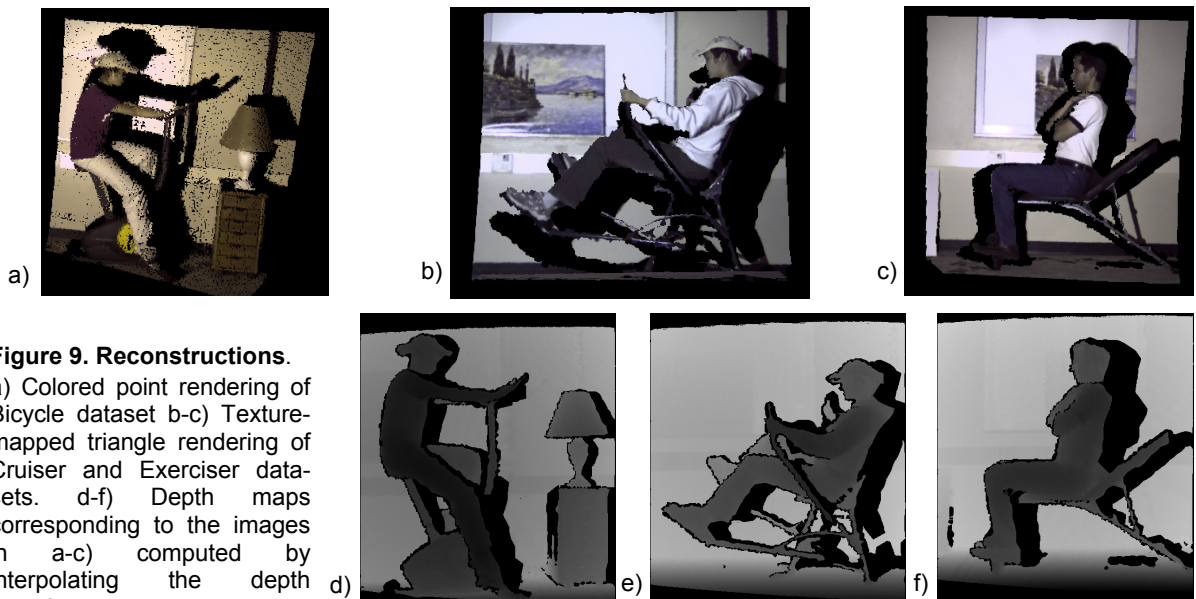


Figure 9. Reconstructions.

a) Colored point rendering of Bicycle dataset b-c) Texture-mapped triangle rendering of Cruiser and Exerciser datasets. d-f) Depth maps corresponding to the images in a-c) computed by interpolating the depth samples.

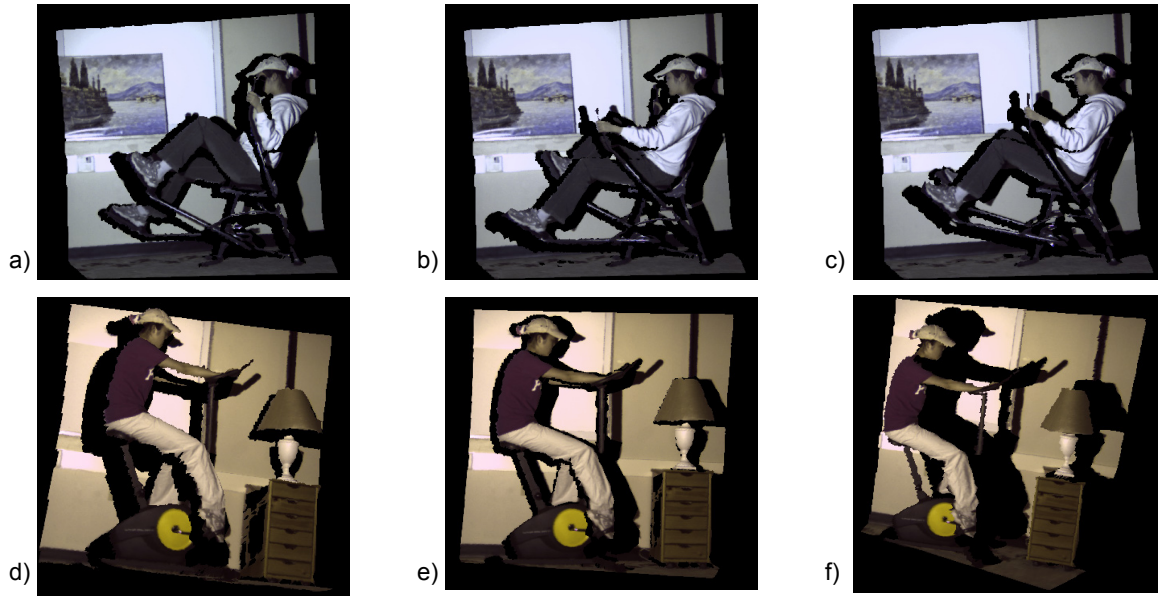


Figure 10. Synthetic sequences. a-c) A temporal sequence of Cruiser from a stationary but novel viewpoint. d-f) A frozen time sequence of Bicycle from novel and different viewpoints.

- Human Locomotion with Flowed Reflectance Fields”, *Proc. of EGSR*, 2006, pp. 183-194.
- [8] P. Fong, F. Buron, “High-resolution 3-dimensional Sensing of Fast Deforming Objects”, *Proc. IROS*, 2005, pp. 1606- 1611.
- [9] C. Frueh, A. Zakhor, “Capturing 2½D Depth and Texture of Time-varying Scenes using Structured Infrared Light”, *Proc. of 3DIM*, 2005, pp. 318 - 325.
- [10] O. Hall-Holt, S. Rusinkiewicz, “Stripe Boundary Codes for Real-time Structured-light Range Scanning of Moving Objects”, *Proc. of ICCV*, 2001, pp. 359-366.
- [11] G.J. Iddan, and G. Yahav, “Three-dimensional Imaging in the Studio and Elsewhere”, *Proc. SPIE Vol. 4298*, 2001, pp. 48-55.
- [12] S. Inokuchi, K. Sato, and F. Matsuda, “Range imaging system for 3-D object recognition”, *Proc. ICPR*, 1984, pp. 806-808.
- [13] T.P. Koninckx, I. Geys, T. Jaeggli, and L. Van Gool, “A Graph Cut based Adaptive Structured Light Approach for Real-time Range Acquisition”, *Proc. of 3DPVT*, 2004, pp.413 - 421.
- [14] T.P. Koninckx, A. Griesser, L. Van Gool, “Real-time Range Scanning of Deformable Surfaces by Adaptively Coded Structured Light”, *Proc. of 3DIM*, 2003, pp. 293 - 300.
- [15] I. Laptev, S. Belongie, P. Perez, and J. Wills, “Periodic Motion Detection and Segmentation via Approximate Sequence Alignment”, *Proc. ICCV*, 2005, pp. 816-823.
- [16] W. Matusik, C. Buehler, R. Raskar, S.J. Gortler, and L. McMillan, “Image-based Visual Hulls”, *Proc. of ACM SIGGRAPH*, 2000, pp. 369-374.
- [17] C. Rocchini, P. Cignoni, C. Montani, P. Pingi, and R. Scopigno, “A Low Cost 3D Scanner Based on Structured Light”, *Computer Graphics Forum*, 20, 3, 2001, pp. 299-308.
- [18] S. Rusinkiewicz, O. Hall-Holt, and M. Levoy, “Real-time 3D Model Acquisition”, *ACM Trans. Graphics*, 21, 3, 2002, pp. 438 - 446.
- [19] F. Sadlo, T. Weyrich, R. Peikert, and M. Gross, “A Practical Structured Light Acquisition System for Point-based Geometry and Texture”, *Proc. of PBG*, 2005, pp. 89-98.
- [20] D. Scharstein and R. Szeliski, “High-Accuracy Stereo Depth Maps Using Structured Light”, *Proc. of CVPR 2003*, pp. 195-202.
- [21] A. Schödl, R. Szeliski, D.H. Salesin, and I. ESSA, “Video Textures”, *Proc. of ACM SIGGRAPH*, 2000, pp. 489-498.
- [22] S. Seitz and C. Dyer, “View-Invariant Analysis of Cyclic Motion”, *Int’l Journal of Computer Vision*, 3, 1997, pp. 231-251.
- [23] J. Starck, G. Miller, and A. Hilton, “Video-based Character Animation”, *Proc. of SCA*, 2005, pp. 49-58.
- [24] S. Vedula, S. Baker, S. Seitz, and T. Kanade, “Shape and Motion Carving in 6D”, *Proc. of CVPR*, 2000, pp. 592 – 598.
- [25] M. Waschbüsch, S. Würmlin, D. Cotting, F. Sadlo and M. Gross, “Scalable 3D Video of Dynamic Scenes”, *The Visual Computer*, 21, 8-10, 2005, pp. 629-638.
- [26] Y. Xu and D.G. Aliaga, “Efficient Multi-viewpoint Acquisition of 3D Objects Undergoing Repetitive Motions”, *Proc. of I3D*, 2007, pp. 113-120.
- [27] L. Zhang, B. Curless, S.M. Seitz, “Rapid Shape Acquisition using Color Structured Light and Multi-pass Dynamic Programming”, *Proc. of 3DPVT*, 2002, pp. 24 - 36.
- [28] L. Zhang, B. Curless, and S.M. Seitz, “Spacetime Stereo: Shape Recovery for Dynamic Scenes”, *In Proc. of CVPR*, 2003, pp. 367-374.
- [29] C.L. Zitnick, S.B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, “High-Quality Video View Interpolation using a Layered Representation”, *ACM Trans. on Graphics*, 23, 3, 2004, pp. 600-608.