

Modeling Repetitive Motions using Structured Light

Yi Xu and Daniel G. Aliaga

Abstract— Obtaining models of dynamic 3D objects is an important part of content generation for computer graphics. Numerous methods have been extended from static scenarios to model dynamic scenes. If the states or poses of the dynamic object repeat often during a sequence (but not necessarily periodically), we call such a repetitive motion. There are many objects, such as toys, machines, and humans, undergoing repetitive motions. Our key observation is that when a motion state repeats, we can sample the scene under the same motion state again but using a different set of parameters; thus providing more information of each motion state. This enables robustly acquiring dense 3D information difficult for objects with repetitive motions using only simple hardware. After the motion sequence, we group temporally disjoint observations of the same motion state together and produce a smooth space-time reconstruction of the scene. Effectively, the dynamic scene modeling problem is converted to a series of static scene reconstructions, which are easier to tackle. The varying sampling parameters can be, for example, structured-light patterns, illumination directions, and viewpoints resulting in different modeling techniques. Based on this observation, we present an image-based motion state framework and demonstrate our paradigm using either a synchronized or an unsynchronized structured-light acquisition method.

Index Terms— Three Dimensional Graphics and Realism, Digitization and Image Capture, Geometric Modeling.



1 INTRODUCTION

Modeling dynamic 3D objects is an important part of content generation for virtual reality, movies, gaming, and other applications. Motion in a scene can vary from being completely random, where no motions repeat, to exactly periodic, where all motions repeat in a fixed pattern. In this paper, we investigate a 3D modeling paradigm that exploits the middle ground of modeling motions that repeat in some way over time. Such repetitive (or quasi-periodic) motions occur with mechanical motions (e.g., toys, pendulum, etc.) and with human activities (e.g., exercises, walking, etc.). Our work investigates efficiently discovering, exploiting, and modeling objects undergoing such repetitive motions, rendering them from novel viewpoints, and generating new motion sequences.

Acquiring and modeling dynamic objects is a very challenging task. The process involves both creating a model of the object and capturing the motion sequence. Many general approaches have been proposed for capturing the geometry and appearance of dynamic scenes. One such group of approaches uses cameras to passively observe a scene and to reconstruct the objects and their motion (e.g. [3][4][8][23][32][41]). However, these methods either do not actually acquire any 3D geometry ([8] uses an external laser scanner to obtain a full body scan) or depend on less robust correspondence computations, color consistency over a large baseline, and/or background subtraction. Another group of approaches actively adds energy (i.e., light) into the scene in order to significantly improve the robustness of 3D reconstruction. Laser scan-

ners obtain detailed models but would require a very fast update rate to complete the scans many times a second as would be needed to support moving objects. Motion capture devices sample the motion accurately but do not obtain a model of the moving and deforming object. A single-shot structured-light method is able to reconstruct the geometry of dynamic objects using the information encoded in one frame ([17][18][38]). But, the resolution and level-of-detail of the reconstructed objects is low. Thus, while an enhanced structured-light method might be able to project a few patterns in rapid succession, they would have to assume the scene is temporarily stationary and/or consisting of slowly moving rigid objects [25]. Moreover, active methods in general interfere with the appearance of the objects which makes simultaneously obtaining dense depth and color hard for dynamic scenes.

In addition, none of these general methods exploit the redundancy introduced by repetitive motions to improve the space-time modeling of the moving objects. Some research efforts have focused on detecting and measuring repetitive motions from images captured by static and/or moving cameras (e.g., [6][21][30]), or on generating new video sequences from images captured over time from a common viewpoint (e.g., [1][28]). However, these methods do not build a 3D model of the observed object.

Our key observation is that repetitive motions can be discretized into a set of motion states that repeat one or more times and the repetitions enable changing the sampling parameters used for each occurrence. Thus, after several repeated observations of an object's motion states, we can accumulate a more complete representation of the object's geometry and of the object's motion. Effectively, we are correlating temporally disjoint samples observing the same motion state. These samples reconstruct a virtually static object thereby converting the dynamic 3D scene acquisition problem into a series of more robust static scene reconstructions. There are a variety of sam-

- Yi Xu is a Ph.D. candidate in the Department of Computer Science, Purdue University, 305 N. University St., West Lafayette, IN, 47907. E-mail: xu43@cs.purdue.edu.
- Daniel G. Aliaga is an Assistant Professor of Computer Science at Purdue University, 305 N. University St., West Lafayette, IN, 47907. E-mail: aliaga@cs.purdue.edu.

Manuscript received (October 31, 2008).

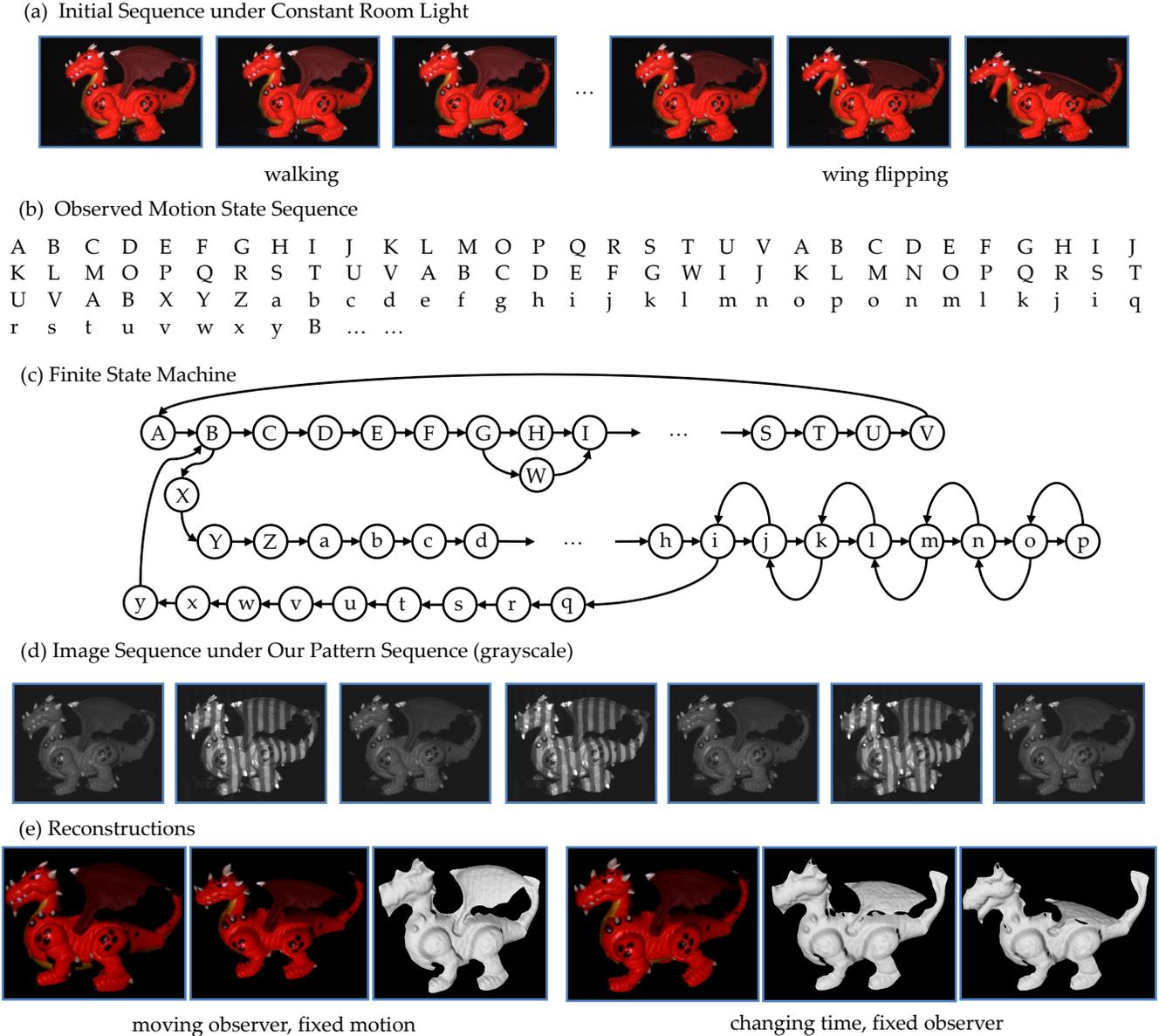


Fig 1. Overview. *a*) A static camera captures the repetitive motion under constant light. *b*) Image-based motion states are identified from the initial sequence. *c*) Using the observed states and transitions, we construct a finite state machine of the motion. *d*) Structured-light acquisition captures video frames under alternating constant room light and projected patterns. *e*) Space-time polygonal models (either texture mapped or gray shaded).

pling parameters that can be changed between each repeated observation of a static scene (e.g. the illumination parameters of an active acquisition method).

The correlation between identical motion states but at disjoint time instances is made feasible by further observing that many 3D motion states can be uniquely identified by their projection onto an image plane. Hence, by assuming a static camera is observing the scene; we can discretize the repetitive motion into image-based motion states (i.e. specific object poses or shape configurations). Although not all motions are distinguishable by their image projection (e.g., a spinning sphere with uniform color), in practice most of them are differentiable using a static camera due to texture, non-uniform color and geometry. We can also use contextual information (e.g., the frames before and after) to further distinguish motion states.

We present an image-based framework for modeling

and acquiring dynamic 3D objects undergoing repetitive motions. Using the assumption that the states of a 3D repetitive motion are distinguishable under image projections, our method first finds the repeating patterns in an initial image sequence captured by a static camera under constant lighting (Fig. 1a). Then, the original repetitive motion is discretized into M motion states, each of which is represented using an image (Fig. 1b). We arrange them into a finite state machine (Fig. 1c). The repetitive motion may include rigid transformations and arbitrary object deformations, but the object must stay within a working volume in order for the static camera to analyze the motion. We demonstrate the framework using a time-multiplexed structured-light method using a pair of camera and projector (Fig. 1d). The observed object and its motion can be played back in the original order or rearranged to produce new motion and in both cases observ-

able from novel viewpoints (Fig. 1e).

Our method obtains dense and robust depth and color samples of the dynamic object. We use time-multiplexed codes to densely sample the object during each motion state without assuming a static scene during several contiguous frames (or tracking patterns between adjacent frames) and without being limited to rigid scene motion. In addition, the repetitive motion also enables obtaining images of the dynamic object free of interferences generated by the projected illumination patterns. We demonstrate our method using several real-world objects and scenes undergoing a variety of repetitive motions. Our main contributions are:

- an image-based methodology for acquiring and modeling repetitive motions in real-world scenes,
- an active dense depth and color acquisition system for scenes containing rigid and non-rigid objects undergoing repetitive motions using synchronized hardware, and
- an unsynchronized version of the system which does not require projector and camera customization.

The work presented in this article extends that of our previous conference publication [37]. A previous passive method of ours [36] uses a static camera to observe repeating motion states, and uses a moving camera to sample each motion state from a set of viewpoints in order to perform a per-state visual hull reconstruction. In this article, we improve the framework introduced in [36] and introduce a new active method (Section 4.1), different than the one of [37] (Section 4.2), for robustly and unobtrusively acquiring dense samples of dynamic scenes undergoing repetitive motions. We also perform qualitative evaluation of our method and compare our method with a standard structured-light implementation for static scenes. For clarity and conciseness in this article, our passive method is not included; we refer the reader to [36] for a detailed discussion of that system.

2. RELATED WORK

Our research builds upon work in active acquisition and repetitive motion analysis and acquisition. While significant work exists in object acquisition, acquiring dynamic objects is still a major challenge for 3D geometric modeling and acquisition.

2.1 Active Acquisition of Dynamic Objects

Active acquisition techniques project patterns onto the scene and capture their projections to obtain depth information. 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 (or camera-projector) correspondences. Zhang et al. [38] use de Bruijn illumination patterns and multi-pass dynamic programming. Fong and Buron [10] use a fixed pattern that combines color stripes and sinusoidal intensity changes. Koninckx et al. [17][18] improve performance by adjusting geometric and color-coded patterns to the scene. Kang et al. [16] use active illumination to assist multi-baseline stereo computation. One-shot methods are

more flexible and easier to use than our method because they do not involve multiple frames. However, they obtain reconstructions of relatively low density or depend on carefully identifying many colors and shapes which is difficult and troublesome in an arbitrary scene.

Some one-frame methods have partially remedied simultaneously projecting structured-light patterns and acquiring scene color. Frueh and Zakhor [11] use two cameras to capture color and low-resolution infra-red structured-light patterns. Iddan and Yahav [13] capture both color and low-resolution depth using a single infra-red “light pulse”. In contrast, in our setting we acquire color and structured-light patterns separately without relying on infra red. Similarly, Waschbüsch et al. [33] use at least three cameras and a rapidly-alternating structured-light pattern and its inverse to enhance the scene with features for stereo matching and to acquire color textures that are illuminated with an all-lit projector. However, in contrast to our method, they cannot acquire the scene under normal room lighting, do not use time-multiplexed structured-light codes but rather stereo, and need a highly customized hardware infrastructure. Another system also uses very specialized hardware to capture models of human facial expressions [15][35]. The system operates at 1500Hz and projects a set of 24 binary structured-light patterns and 29 basis lighting directions at 24Hz. Each set of images are warped to the same time instance using optical flow. Similar to our method, they also insert frames captured without patterns. However, they use these pattern-free frames to track the optical flow; while we use these images to assist motion state matching and do not require a 1500 Hz camera.

For slowly moving objects, multi-frame methods can be used. For example, Rusinkiewicz et al. [25] 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. Their method targets rigid objects. The space-time stereo methods [7][39] project changing patterns for a few consecutive frames and use a slanted window during stereo correspondence. However, since the number of patterns is limited, they use stereo matching to enhance resolution; thus requiring at least one more camera than our method. Weise et al. [34] present a fast 3D scanning system using a modified projector and three cameras. Their system uses three phase-shifting patterns and stereo-based phase unwrapping to establish correspondence. This requires a more complicated hardware setup than ours. In our method, temporally disjoint frames can be corresponded and no limit is placed on the code length; thus a single projector-camera pair can be used to acquire high-resolution samples. Moreover, we use robust Gray code patterns for correspondence; as compared to multi-view stereo methods, our method is more robust to non-Lambertian surfaces, depth discontinuity and supports faster moving scenes (see Section 6.4).

For acquiring non-rigid deforming objects, photometric stereo can also be used. Color photometric stereo method [12] uses three lights (e.g., red, green, and blue lights) and captures the scene under these lights simulta-

State Sequence	Motion Type
(A)*	static
ABCDEFGH...	random
ABCDGHA JKB ...	low multiplicity random
ABCAD E ABCABE...	high multiplicity random
(ABCDEFGH)*	periodic
(ABACDEFGH)*	low multiplicity periodic
(ABCAD E AB)*	high multiplicity periodic

Table 1. Motion Categories. We categorize motions into several types; the bottom four are considered as repetitive motions in this article. Each letter represents a motion state and is color coded to highlight repetitions.

neously using three color channels. This method assumes the surface of the object to capture has uniform material properties. In addition, the approach needs a calibration object of the same material as the target object. Furthermore, no absolute geometry can be acquired since the surface is integrated from a normal field.

Ultimately, impressive results can be obtained by fitting the acquired data to a prior synthetic model (e.g. the face model used in the space-time face [40]). The focus of our method is acquisition; nevertheless, the data acquired using our method can also be the input to other modeling and animation approaches that use predefined models.

2.2 Repetitive Motions

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 different motions exhibit such repetition. Thus, for clarity we show in Table 1 a categorization of object motions based on the sequencing pattern and frequency of the motion states. In particular, a static scene corresponds to a single state that repeats indefinitely. A completely random motion means every state is different and occurs only once. If some states appear more than once, then, based on the frequency of the appearance of the states, the motion is defined as low multiplicity or high multiplicity random motion. If a whole subsequence repeats itself infinitely, the motion is periodic. By using the repetition frequency of the motion states within a single period, periodic motion can also be sub-classified as low-multiplicity or high-multiplicity.

Several research efforts have focused on detecting and measuring periodic motions usually from image sequences captured by static and/or moving cameras (e.g., [6][21][30]). Video textures and its extension to panoramas generate an infinitely varying image sequence that enhances images from a stationary viewpoint with projections of repetitive motions [1][28]. State automata can also be used for hand gesture recognition [22]. However, these works focus only on periodic motion analysis and/or do not build a dynamic 3D model of the object over time.

Some passive 3D reconstruction methods have been proposed for periodic or quasi-periodic motions. For example, Starck et al. [31] capture human motion from a multiple (10) viewpoint video sequence. They use a motion graph [20] to generate transitions of different motions

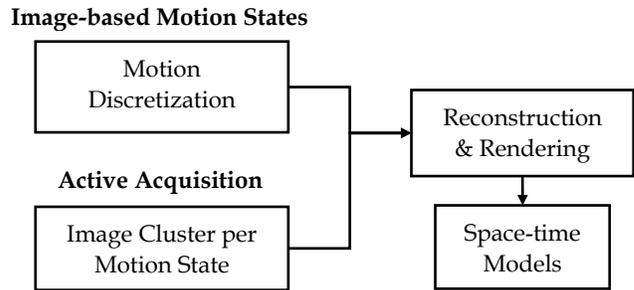


Fig 2. System Pipeline. Our system consists of three major components. Image-based motion states provide information for active acquisition in order to create space-time models for the repetitive motions.

in order to synthesize novel animation sequences. Sand et al. [27] use a complicated hardware infrastructure to acquire motion data by using a commercial motion capture system and silhouette data by using a set of surrounding synchronized cameras. They use the repetitiveness of local configurations of an articulated object (e.g., human body) to construct a complete and high-quality model. Both of the aforementioned methods require multiple camera setups and rely on foreground extraction. Einarsson et al. [9] follow a different approach that acquires a 7D time-varying reflectance field of constant-speed human locomotion on a treadmill, but no explicit geometry is reconstructed. Similar constant speed assumption is used to create motion lumigraph for toy helicopter in [2].

In contrast, we define repetitive motion to consist of the bottom four types of Table 1, namely high-multiplicity random motion and periodic motions. Moreover, our 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 3D scene reconstruction. Hence, the resulting captured motion and geometry is similar to, but not necessarily identical to, the true object motion.

3. IMAGE-BASED MOTION STATES

A fundamental part of our approach is to discretize the repetitive motion into a set of motion states. Fig. 2 shows the pipeline of our system. Our approach requires both determining a set of image-based states that represent the observed motion and choosing states that have been sampled sufficiently for the reconstruction algorithm. Towards this goal, we have developed two algorithms for separating the motion into states: global motion discretization and real-time motion discretization. We explain these methods in the following two subsections.

3.1 Global Motion Discretization

Global motion discretization requires all images to be already captured and obtains a set of motion states with maximized sample coverage per state but at the expense of additional capture time and storage space requirements. The image acquisition strategy is straightforward. We use a static camera to continuously record images of

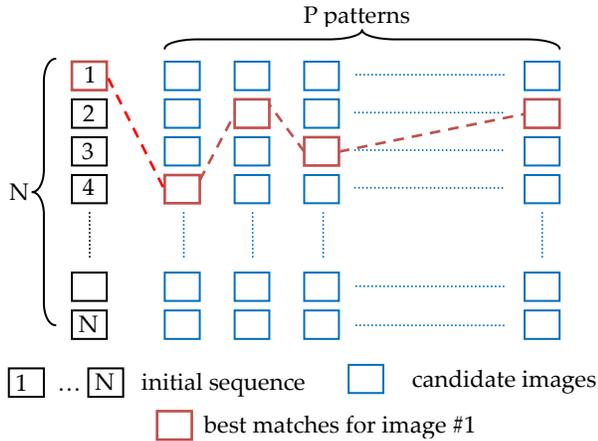


Fig 3. Global Motion Discretization. For each image in the initial sequence, we find an array of the best-matched candidate images. The initial N images are ranked based on the average similarity to their corresponding candidate images.

the repetitive motion and a projector to project structured-light patterns oblivious to the repetitive motions of the object. The static camera first captures N initial frames. The number N is chosen to be long enough so that general repetitive motion can be captured. Then it captures an additional N frames for each of the P structured-light patterns. The acquisition is not synchronized with the repetitive motion in any way. After acquisition, the images are grouped into clusters where each cluster corresponds to a sufficiently sampled motion state and the clusters are organized into a finite state machine that describes the overall motion sequence.

3.1.1 Image Clusters

First, the captured data is grouped into a list of N image clusters where each cluster contains one initial frame and the best-matched candidate image from each of the remaining P patterns. The images are clustered by comparing an initial frame against all the candidate frames and finding the one that minimizes the average per-pixel difference for each pattern. As shown in Fig. 3, the comparison for each initial frame produces an array of length P where each entry contains the index of the most similar image from each of the P pattern image sequences. Images in the same cluster observe and sample the same motion state. The per-pixel image difference between a candidate image (with pattern) and an initial frame

(without pattern) is large, thus it is difficult to compare them directly. To enable image clustering, we use a dynamics preserving image matching or a color-calibrated image matching. These two methods will be introduced in Section 4 with more details. The discretization process also assigns to each of the initial N frames a rank, which is computed by comparing the average per-pixel difference of each initial frame to all the candidate images within the cluster. The ranking will give preference to motions that tend to repeat and will downplay spurious motions.

3.1.2 Finite State Machine

Using the ranked initial images, a compact finite state machine of the motion states is extracted through an iterative process. First, each initial frame (and its associated image cluster) is labeled as a unique motion state. Two states are merged into one if and only if their per-pixel image difference is smaller than a predefined clustering threshold. The merging process repeats iteratively until no states are similar enough. After convergence, each motion state has one or more initial frames. The initial frame with the highest rank is used as the representative frame for that motion state. From the resulting motion states, we construct a finite state machine that is representative of the observed repetitive motion. The overall process ensures that the best set of images per motion state is used during reconstruction; thus, the reconstruction quality is maximized. The best matching quality comes at a price of many image comparisons: each of the N initial frames is compared against NP candidate images and against each of the other N initial frames during finite state machine creation.

Although one would expect a perfectly repeating set of motion states for periodic motion, both periodic and quasi-periodic motion may include spurious states together with the repeating motion sequence. In the case of periodic motion, the discrete time sampling of the camera may cause a new and different state to be occasionally captured. For example, a pendulum can be observed from the static camera as predominantly having the state sequence $(ABCD CBA)^*$ but sometimes the sequence $(ABCDCEA)$ may appear. A quasi-periodic motion is one that does not exactly repeat but exhibits significant similarities over time. We place the states and transitions that appear more than a predefined number of times into a finite state machine and we disregard spurious states. To ignore dead-ends, we extract the largest strongly-connected component of the state machine. A similar method is used in [20]

State Sequence and Ranks (the frames with underscores are the best from each motion state):

A	B	C	<u>D</u>	<u>B</u>	C	D	B	<u>C</u>	<u>E</u>	<u>F</u>	G	A	B	C	H	F	G	<u>A</u>	B	C	E	F	<u>G</u>	A
24	22	14	6	1	21	10	11	4	13	2	7	26	16	9	23	25	12	5	15	19	17	18	3	8

Finite State Machine:

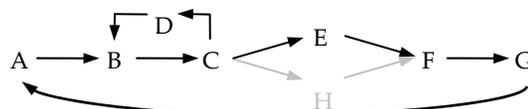


Fig 4. Motion State Machine. a) Top: using the static camera, the observed initial frames are partitioned into motion states (letters) and ranked based on sampling quality (numbers). b) Bottom: both periodic and quasi-periodic motions are approximated by a finite state machine.

to prune motion graphs.

Fig. 4 shows a didactic example of the ranks of the initial frames, the corresponding motion state sequence, and the resulting finite state machine. Given a hypothetical initial sequence of 26 frames, the system groups the motion into a high-multiplicity periodic-motion sequence of only 7 states, and one spurious state that is discarded (state H in this case). A more complicated finite state machine for a real dataset can be found in Fig. 1.

3.2 Real-Time Motion Discretization

Real-time motion discretization first uses a short initial frame sequence to calculate a set of motion states. Then it classifies all subsequent images in real time to one of the motion states. It provides the benefit of reduced capture time and storage space but a slightly more complicated image acquisition strategy. From the initial sequence, a finite state machine is constructed as previously described (without the ability to rank images). Then, our method matches a current image to a cached representative image of one of the motion states. If an input image could not be sufficiently matched to any state, it is discarded. This implies that spurious motion states of the repetitive motion are appropriately ignored since they do not help to sample the desired motion. The threshold for the matching controls the overall quality of the data fed to the later reconstruction phase. A large threshold might lead to a fast acquisition, but images of the same state can vary significantly. On the other hand, a small threshold leads to slow completion, but images within each state are very similar. During real-time motion discretization, a currently captured frame is compared against all the frames in the cached finite state machine; therefore, only simple motions can be discretized in real time using lower resolution. For example, the *flag* example shown in Fig. 7 can be acquired using real-time motion discretization at quarter resolution and 30 frames per second (~ 30 states).

3.3 Motion Rendering

Our method allows us to rearrange the order of the motion states producing new motion sequences and repetitive motions. The finite state machine contains all the potential states and transitions. Similar to video textures [28], we produce repetitive motions similar to or different from the original motion by walking through the finite state machine in different ways. However, since for each state we already have sampled images and a reconstructed 3D model, the observer can freely move the virtual viewpoint during the new motion sequence. For example, if a motion is $(ABABABABCDCCDCD)^*$, then we can play back as is or change it to $(ABCD)^*$, $(BCDCDA)^*$, or to $(AB)^*$ and in all cases allow the observer to continuously change the viewpoint.

4. ACTIVE ACQUISITION

Our image-based motion state framework enables an active and robust acquisition of repetitive motions. We capture a single video sequence using a static camera and use only one additional digital projector to project illumination patterns onto the scene. Further, when observing

repetitive motions, our framework enables using either a synchronized or unsynchronized camera-projector system. The synchronized version provides the ability to obtain images of the scene under constant room lighting without requiring a controlled environment while the second approach omits the need for any synchronization efforts but only acquires images under controlled projector illumination. In the following two sections, we describe both active methods.

4.1 Synchronized Acquisition

Our synchronized camera-projector system samples the motion using a structured-light approach which captures repeating three-frame sequences. Our method assumes the scene observed during a three-frame sequence is moving and will repeat.

4.1.1 Dynamics-Preserving Image Matching

Our method discretizes the motion but uses “states” effectively defined by a consecutive triple of frames. First, a sequence of initial frames is captured under constant room lighting (i.e., projector turned off) as in Section 3. Each of these initial frames corresponds to a different motion state frame (Fig. 5 middle row). Then, the dynamic scene is captured with the camera synchronized to the projector and using an alternating sequence of structured-light pattern (projector turned on) and constant room light (projector turned off). Fig. 5 top and bottom rows

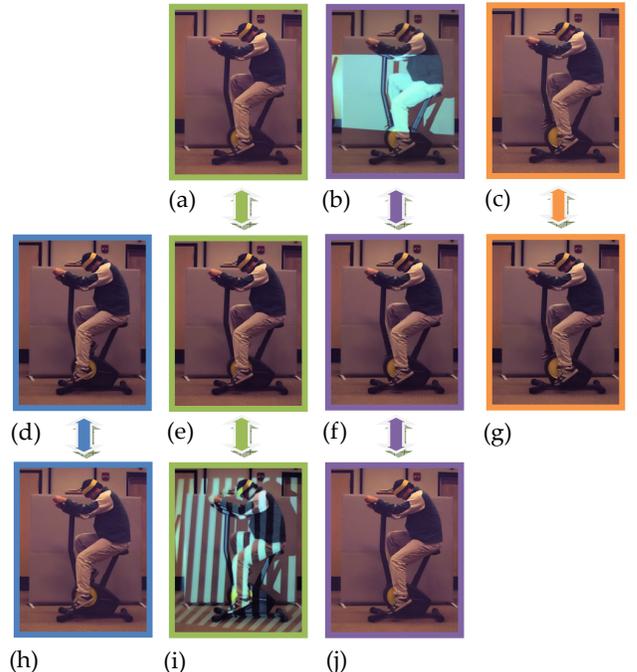


Fig 5. Dynamics-Preserving Image Matching. Middle row ($d-g$) shows four consecutive motion state images. First and third rows show three consecutive frames using two different alternating patterns respectively. Images from the same column are observing the same motion state. Frame b (captured under a binary pattern) observes the same motion state as frame f because frames a and c match frame e and g respectively. Frame i also observes the same motion state as frame e .

show such image sequences for two different patterns. The images before and after each structured-light pattern frame are captured under the constant room light.

During motion discretization, we best match a captured pattern frame to a motion state image by comparing the temporally adjacent frames. For example, the difference between Fig. 5b (candidate pattern frame) and 5f (pattern-free motion state frame) is calculated as the sum of the difference between Fig. 5a and Fig. 5e and the difference between Fig. 5c and Fig. 5g. Let R represent a frame captured under constant room light and S represent a frame captured under structured light; thus, $R_1R_2R_3$ represents a three-frame sequence of motion state images. The best-matched three-frame candidate sequence should have the configuration $R_1S_2R_3$, and not $S_1R_2S_3$ in order to minimize the image differences for the first and third frames. This is because the image difference between an R frame and an S frame is large. By assuming the motion dynamics is preserved during a three-frame sequence, we can conclude that frame S_2 and frame R_2 sample the same motion state. In this way, we avoid comparing an S frame directly to an R frame. The S frames are then grouped together eventually yielding a full time-multiplexed sampling of each motion state.

We highlight that this comparison method supports a wide range of motion during the three-frame sequence (i.e., we do not suppose only slow motions occur) and does use both neighbors for image matching. Consider the following pendulum-like repetitive motion state sequence: $(ABCDCA)^*$. Motion state B can be followed by both state C and A . This leads to ambiguity if only the leading frame is used for image matching. Although two motion states might sandwich more than one possible state; the probability is low due to the fact that motion dynamics is preserved in three consecutive frames captured at a relatively high frame rate. The use of neighboring and contiguous images to preserve motion dynamics is also exploited by Schödl et al. [28]. However, in our work the window of images is used to match motion states under very different illumination scenarios.

It is worth noting that our method for image matching does not require an explicit classification of the candidate frame as under room or structured-light illumination. The best matching mechanism implicitly enforces the three-frame sequence to be under room light, structured light, and room light. It is also interesting that the frame rate of the reconstructed motion is not reduced in half because

all the motion-state images (Fig. 5d-g) can be sufficiently sampled and they are captured at full camera frame rate.

4.1.2 Reconstruction

We use a time-multiplexed Gray code [14] to reconstruct the scene but perform additional computations to account for minor scene fluctuations among the captured patterns of the same motion state. Since the motion will not be perfectly repeating, there is a small variation over the multi-pattern images per motion state. This leads to bad samples along the boundary of the foreground object. To alleviate this problem, we represent the point samples as a depth map and perform morphological operations. We first apply erosion to remove outliers and smooth the boundary of the reconstructed geometry. Then, we perform dilation to compensate for missing samples due to motion variation. The dilation is guided by the segmented motion state image, i.e. we only extrapolate within each color segment so as not to pass through the depth discontinuity similar to the method in [33]. After world-space and image-space outlier rejection, the final point samples are then triangulated from the 2D viewpoint of the static camera. Triangles either too big or too thin are rejected. Based on the triangulation information, a Laplacian operator is used to smooth the raw reconstruction.

4.2 Unsynchronized Acquisition

An alternative method that omits the need for any synchronization and pattern alternating is using two-color patterns, rather than standard black-and-white patterns. This requires capturing fewer images than the synchronized method because all candidate frames are captured with a colored binary pattern, while in the synchronized method only half of them have a projected structured-light pattern.

4.2.1 Two-Color Patterns

Projecting constant structured light prevents having to rapidly change illumination patterns. However, it necessitates robustly matching the candidate images illuminated by structured-light patterns to initial frames captured under constant illumination. This is difficult because compared images would be under very different illumination conditions. It is even worse when using traditional black-and-white patterns (either stripes or sinusoidal patterns) because any motion occurring in the non-illuminated (i.e., black) areas will be invisible (Fig. 6c). The problem is severe during the initial stripe patterns of

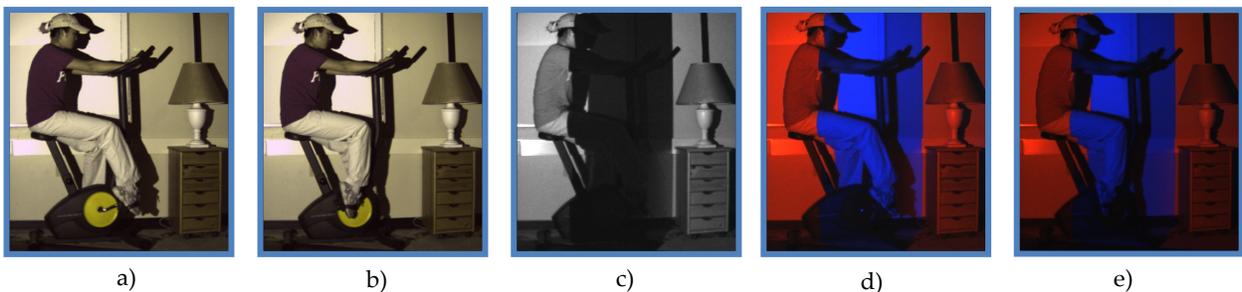


Fig 6. Two-Color Structured Light. a-b) Two example motion states under all white light. c) Using conventional white-black stripes, both states seem identical. d-e) Our calibrated two-color structured light discerns between the two states.

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. This ambiguity affects both the motion-state matching and the decoding process for scene reconstruction. It 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). In general, it leads to pixels being inconsistently labeled amongst the pattern images and thus producing fewer correctly decoded pixels. However, the total amount of ambiguity is scene and motion dependent.

Our approach uses two-color (red and blue) patterns (Figure 6d-e) to enable accurately matching images with projected patterns to initial frames captured under a floodlit projector pattern (a pattern with all pixels set to white). This is a different use of color than that for reducing the total number of patterns [5] or for improving robustness [24]. Further, it is worth noting that in many structured-light methods for dynamic scenes, the images illuminated by the patterns are the only source of color information. In our case, we can extract color from the floodlit motion-state images and thus the two-coloring of the patterns does not affect the final color of the reconstructed object.

4.2.2 Color-Calibrated Image Matching

To compare two-color structured light candidate images to the initial floodlit motion-state images, we use a color-calibrated image matching method. The key idea is to compare a pixel illuminated by a red (blue) stripe to the red (blue) channel of the corresponding pixel in the motion-state image. Based on the typical color spectrum of digital cameras, we expect the red and blue color channels to offer little or no overlap and thus we choose red and blue color stripes. We adjust camera parameters to prevent saturation when capturing floodlit motion-state images. These 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 due to the different spectral compositions of projectors and cameras and the spectral overlap between green channel and red/blue channels.

The actual radiometric chain from projector pixel value to camera pixel intensity is complicated. The projector pixel value is first mapped to radiant flux. Then, the surfaces modulate the incoming light and reflect it towards

the camera. The camera maps irradiance from the scene to camera pixel values. Both the camera and projector response curves are nonlinear; thus making the calibration difficult. However, a complete calibration is not necessary for our purpose. Unlike other structured-light methods which rely on accurate recognition of projected colors, the red-blue stripes in our method can be easily distinguished from each other. Furthermore, the image matching method finds a best matched object pose from a set of candidates using image difference. An exact zero difference is not crucial because a mismatch in object pose leads to a larger difference than that caused by imperfect color calibration. Please refer to [19] for a detailed discussion on projector-camera chain calibration for structured light.

In our system, we assume a simple linear color model. 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 and blue channels of the all-white-light floodlit image that minimize the difference between the single-channel images and the floodlit image. The scale values depend on the scene and on the actual hardware used. In practice, we found this simple linear color response model works quite well.

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

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

After motion-state matching, the captured images are rearranged to yield a time-multiplexed sampling of the scene. A 3D reconstruction is then performed and the all-white-light floodlit motion-state images are used as textures for the resulting triangulation.

Dataset		Capture Time	Capture Frame Rate	No. of States	Images Stored	No. of Points	Discretization Time	Reconstruction Time
Synchronized	Dragon	13.3 min	60Hz	52	13600	~14000	4.3 hr	0.12 hr
	Flag	14 min	30Hz	36	6120	~14700	0.6 hr	0.1 hr
	Bicycle A	5.1 min	20Hz	20	6120	~35000	1.2 hr	0.1 hr
	Cruiser A	5.7 min	20Hz	22	6800	~34500	2.2 hr	0.1 hr
Unsynchronized	Bicycle B	5.5 min	30Hz	34	9900	~33000	3.5 hr	0.2 hr
	Cruiser B	4.6 min	30Hz	53	8250	~33200	3.2 hr	0.5 hr
	Exerciser	2.7 min	30Hz	20	4900	~34000	0.6 hr	0.15 hr

Table 2. Datasets. We list the characteristics of the datasets used for testing our methods. The first four are captured using our synchronized active acquisition method and the later three are captured using our unsynchronized method.

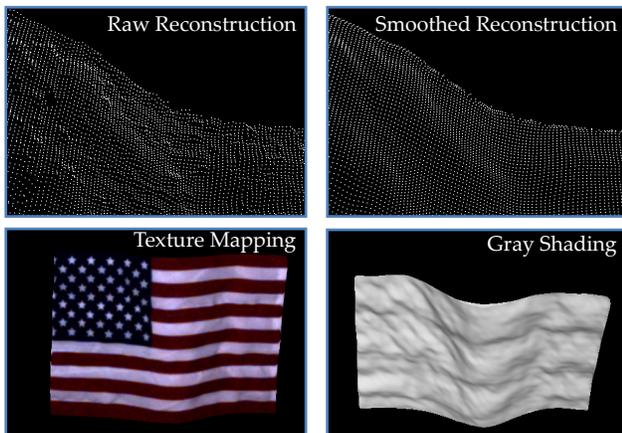
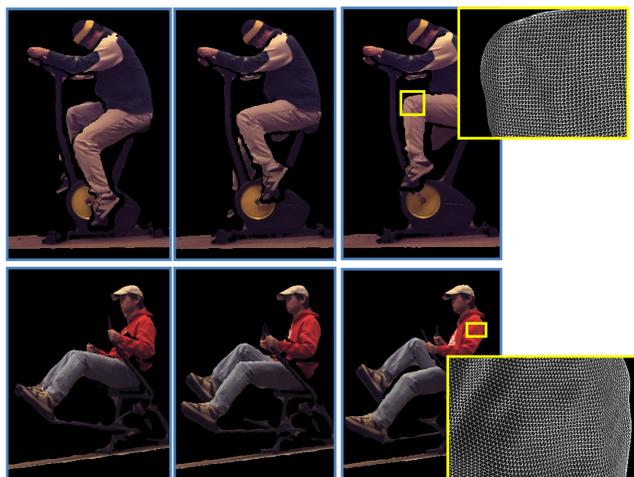


Fig 7. **Flag.** We show the reconstruction of a flag toy, which uses a motor to generate periodic waving effects.

5. IMPLEMENTATION DETAILS

We implemented our system in C/C++ using OpenGL and OpenCV libraries on a PC with a 3.6GHz CPU and 2GB memory. The prototype implementation of our system uses a 640x480 200Hz Point Grey Research (PGR) digital camera connected to the PC for determining image-based motion states and a Canon Realis SX6 LCOS projector (1280x1024 pixels and three explicit LCD panels so as minimize spectral overlap). The projector and camera are geometrically calibrated using the method in [26]. In the synchronized method, the camera is externally triggered using the v-sync signal of the projector at 60Hz. Due to disk write performance, images can only be stored in memory at this frame rate. In order to output the frames to disk in real time, a lower frame rate (e.g. 30Hz) can be used. In the unsynchronized version, the exposure is set to be long enough to ensure the camera grabs a stable pattern image. Thus, the system runs at 30Hz. Color calibration used for two-color structured-light patterns



a) changing time, fixed novel viewpoint

computed that the camera’s average scale factors to make floodlit images match separately-illuminated images are 1.398 and 1.043 for red and blue, respectively.

6. RESULTS AND DISCUSSION

We have captured several datasets. Table 2 shows the details. For each dataset, we list the time for acquiring the images, the capturing frame rate, the number of reconstructed motion states, the total number of captured images, the number of points, the motion discretization time, and the model reconstruction time.

6.1 Reconstruction Results

Fig. 1 depicts the system pipeline using the *dragon* dataset. The static camera captures a video sequence of the scene under constant room light (Fig. 1a) and observes 52 motion states (Fig. 1b). From the observed states and transitions, our method automatically builds a finite state machine of the motion sequence (Fig. 1c). During synchronized active acquisition, the projector illuminates the scene with different Gray code patterns (alternating projector on and off) and the camera captures the images (Fig. 1d). After global motion discretization, our method constructs a model for each of the 52 motion states. After reconstruction, a moving observer can freeze the motion (Fig. 1e left) and a static observer can see the motion over time (Fig. 1e right).

Fig. 7-8 show additional results using the synchronized acquisition system. Our method can faithfully reconstruct dense depth and color samples of repetitive motion, including certain types of human motion such as motion sequences on a *bicycle* and *cardio cruiser*.

Fig. 9 shows results using the unsynchronized version of our active method. Using two-color structured light, we are able to cluster images captured under different color patterns with the floodlit motion-state images (Fig. 9a). Using the images with Gray code patterns per motion



b) moving observer, fixed motion state

Fig 8. **Synchronized Active Acquisition:** We show two datasets (*bicycle A* and *cruiser A*) acquired using our synchronized active method. a) Changing motion states and a fixed novel viewpoint. b) Changing novel viewpoints and a fixed motion state. Examples are rendered using either texture mapping (using motion state images captured under constant room light as the textures) or gray shading. Wireframe zoom-in views are also provided (highlighted in yellow).

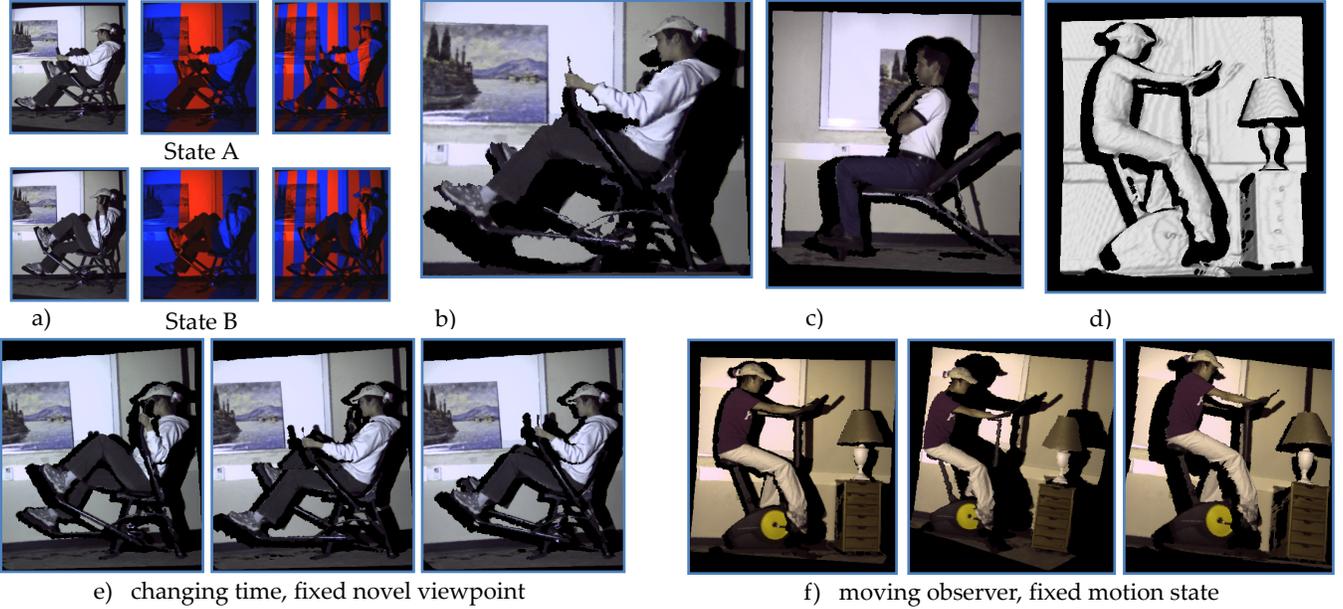


Fig 9. Unsynchronized Active Acquisition. a) Images captured under all white light and two-color structured light for two different motion states. b & c) Texture mapped polygonal models for the *cruiser B* and *exerciser* datasets rendered from novel viewpoints and d) polygonal model rendered using gray shading for the *bicycle B*. (e-f) Additional modeling results.

state, our method successfully reconstructs 3D models of several scenes (Fig. 9b-d) allowing novel view rendering. Fig. 9e-f shows additional space-time modeling results using our unsynchronized active acquisition method.

Finally, the repetitive motions of any of our datasets can be altered yielding new and different motions. The user specifies a new way to traverse the finite state machine and can still freely move the virtual viewpoint. More examples are in the accompanying video.

6.2 Clustering Threshold

Our global motion discretization method clusters the initial N frames to form motion states. Fig. 10 illustrates the tradeoff that results from different clustering thresholds using the unsynchronized acquisition. The horizontal axis represents the amount of clustering performed between initial frames. The vertical axis represents the

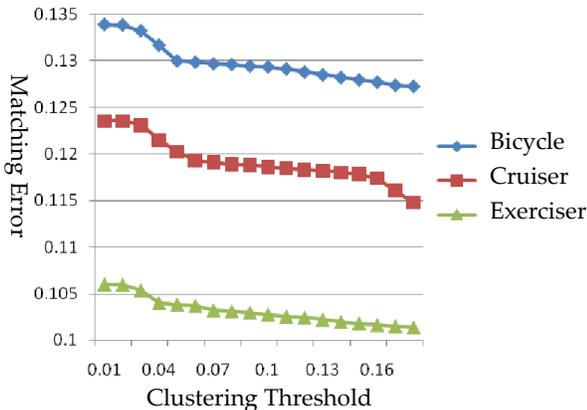


Fig 10. Clustering Trade-off. We show the tradeoff between clustering and matching error. Larger clustering threshold leads to smaller matching error to the pattern images. The sweet spot is in the range of $[0.04, 0.1]$.

average of the smallest matching error between candidate images and initial images of each motion state. The units of both axes are average per-pixel difference. Ideally, we desire a solution at $y=0$ (i.e., zero motion difference between the pattern images and the white-light motion-state images) and $x=0$ (i.e., no clustering, thus exactly each frame of the initial sequence is reconstructed). However, except for perfectly periodic motion and temporal sampling perfectly synchronized with the repetitions (which results in a horizontal line at $y=0$), this is impractical. Thus, we seek for a “sweet spot” that produces a balanced trade-off of the two errors. For example, the matching error decreases monotonically when the clustering threshold increases, but at a slower rate when the clustering threshold is larger than $x=0.04$; thus we can use this threshold to achieve a good balance. A different criterion is to minimize the absolute difference between clustering threshold and matching error. In this way, the difference between pattern images and motion-state images is no more than the difference amongst motion-state images -- and vice versa. This in general occurs at $x=0.1$. Therefore, for our datasets, the sweet spot is in the clustering range of $[0.04, 0.1]$, and this is the range we used.

6.3 Accuracy and Completeness

To evaluate the reconstruction quality of our method, we compare the reconstruction of a static pose of the *flag* to the most similar motion state during a repetitive motion sequence. Since the two reconstructions are implicitly registered (i.e., they are observed by the same camera), we can easily compute the average distance between points that share the same code value and are in both reconstructions. We perform the analysis using the synchronized system. The limiting accuracy of our structured light system, at the resolution of our hardware and for a static scene, is 0.6mm. This is the average distance from

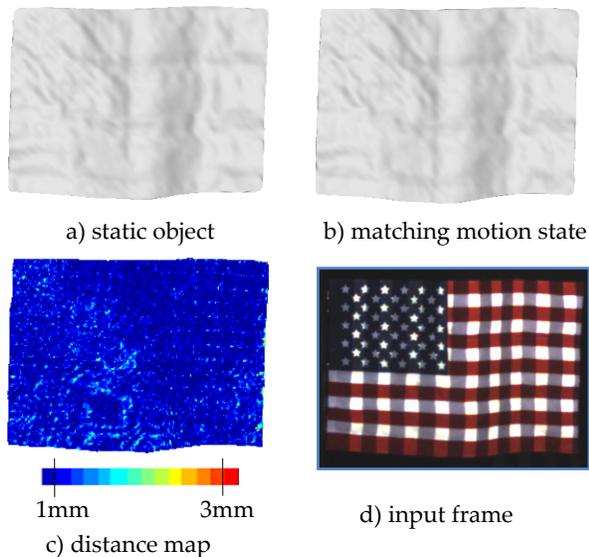


Fig 11. Accuracy Comparison of the Reconstructions. The top row shows the reconstructions of *a)* reference static object and *b)* its corresponding motion state captured by our system and rendered using gray shading. *c)* A color-coded distance map between the two reconstructions. *d)* One of the input frames to our system.

the point samples reconstructed for a diffuse planar board to a fitted hypothetical plane. The average point-to-point distance between the reconstructions of a motion state and the corresponding static pose is only 0.69mm. The standard deviation is 0.73mm. The diagonal of the flag is about 300mm. Therefore, our active acquisition is able to obtain dense samples for moving objects with good accuracy. Fig. 11 shows a color-coded distance map between the two reconstructions. Most of the points are close except for some boundary points.

To further evaluate the accuracy and completeness of our method, we use the same metric as in [29]. First, we place a *vase* object (with a radius of about 100mm) on an electric turntable rotating at 4 rpm (0.8 degrees of rotation per image when capturing at 30Hz). The object is placed approximately 60 mm off-center so that when it rotates, an apparent large rotational motion occurs. We capture six static poses S_i of the object by manually rotating the turntable about 20 degrees each time to six different orientations and using a structured-light acquisition system for static objects. Then, we turn on the electric turntable and capture a space-time model of the object rotating on the turntable by using both our methods. The reconstructions R_i of the six corresponding motion states to the six static poses are used in the evaluation. To measure accuracy, we compute the distance d_i for which 90% of the points of R_i are within distance d_i of S_i . Smaller d_i value means more accurate reconstructions. Fig. 12a shows the results for the six R_i . As can be seen, our method in almost all cases produces accurate reconstructions within 1mm of those of the corresponding static poses. To measure completeness, we compute the fraction of points of S_i that are within distance d_i of reconstruction R_i . Fig. 12b shows the results for $d_i=1$ mm. Our method always produces more than 85% of the maximum possible number of

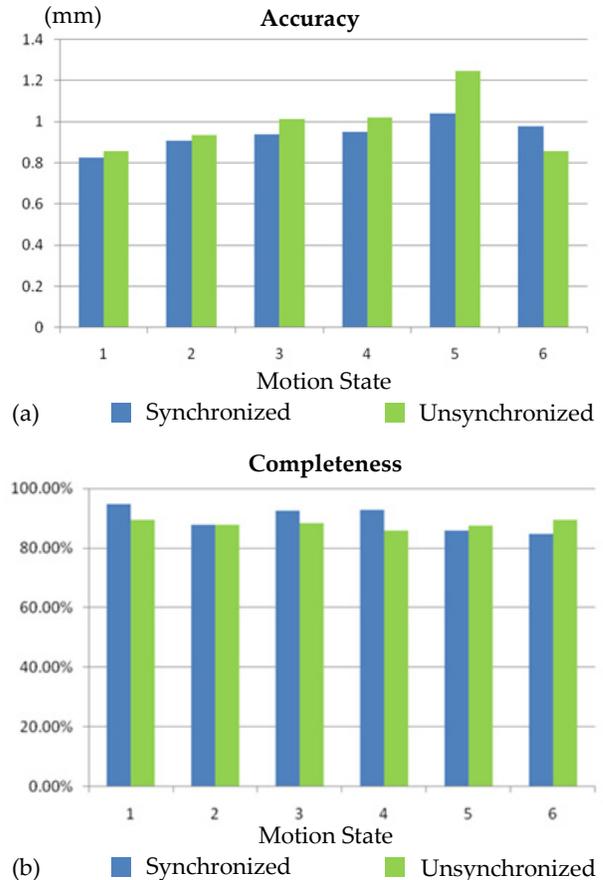


Fig 12. Accuracy and Completeness. We compute the a) accuracy score and b) completeness score for six different motion states of a vase rotating on a turntable. The two methods produce similar results.

points. In most cases, the synchronized method produces slightly better (lower accuracy score and higher completeness score) reconstructions than unsynchronized method, but not always.

To quantify the advantage of using our two-color patterns vs. standard black-white patterns during unsynchronized acquisition, Fig. 13 compares the results of using red-blue patterns for the bicycle dataset to only using a 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 Fig. 6.

6.4 Comparison with Multi-frame Methods

Multi-frame methods (e.g. space-time stereo [7][39]) utilize inter-frame consistency to enhance acquisition resolution of a single frame active stereo. A thorough study in [7] concludes that for a static scene, a long temporal window is optimal (i.e. using many patterns in a structured-light setting). However, for dynamic scenes, they indicate shorter temporal windows yield better results. In their experiments, a translational motion of 3-4 pixels between consecutive frames reduces the optimal temporal window size to about 3-4 frames. For rotational motion,

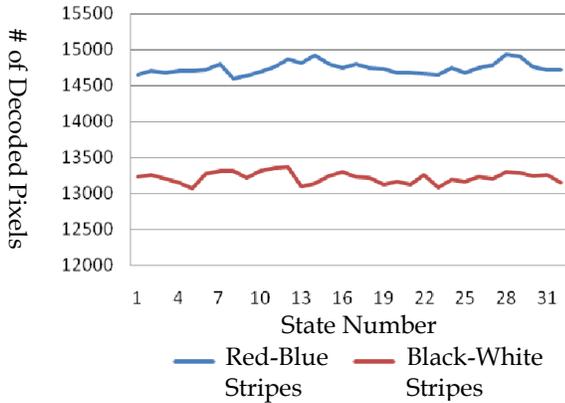


Fig. 13. Comparison of Decoded Pixels. Because of better state matching, our two-color structured light patterns decode more pixels than black-white patterns.

the optimal temporal window is 8 frames for a slow 0.3 degrees per frame motion and only 2 frames for a faster 3 degrees per frame. Hence, as the motion speed increases, the optimal window size for space-time stereo analysis tends to reduce. Upon reaching one frame, space-time stereo becomes equivalent to either standard stereo (i.e., depends on passive feature detection) or to one-frame structured-light methods (see Section 2.1).

In contrast, our method is able to support faster motions and still produce high-quality and dense reconstructions. To compare our method with such multi-frame methods, we perform reconstructions of scene motion occurring at different speeds. We place a bear object on the aforementioned turntable and alter the turntable speed. The object is placed off-center to simulate both translational and rotational object motion. Fig. 14a shows the maximum pixel motion for consecutive frames in the original video sequence and in the image cluster for a particular motion state computed by our method. Our approach is able to group pattern images of the same motion state together, despite the large object motion speed (e.g., up to nearly 5 degrees per frame) and produce reconstructions. Fig. 14b shows the reconstruction of the bear when it is static and Fig. 14c shows various reconstructions when the bear is rotating at increasing speed. While our method shows improved resilience to faster motion speeds, very high motion speeds can eventually cause problems because of an increase in the number of spurious states (Section 3.1.2) and motion blur (the latter

also being a limitation for space-time stereo).

An alternative design of our system could be to incorporate the mentioned multi-frame methods with our repetitive motion recognition. For example, instead of using a full Gray code pattern, we could project high-frequency stripe patterns such as those used in space-time stereo [7][39] (though still interspersed with images captured under constant room lighting). When there are motion repetitions, we would use the repetition to cluster pattern images seeing the same motion state together. This would enable using longer temporal windows for space-time stereo processing and potentially result in improved results. This design does not assume the images in the same cluster are exactly repeating. However, it does not have the advantages of a highly controlled Gray code, such as dense acquisition and robustness against depth discontinuity, and would require an additional calibrated camera.

7. CONCLUSIONS AND FUTURE WORK

We have introduced an image-based methodology for modeling repetitive motion as a set of smoothly changing motion states and for obtaining a dense and robust space-time reconstruction of the scene. The repetition is exploited in order to convert the dynamic scene acquisition problem into that of capturing many static scenes. Further, our approach obtains dense reconstructions without needing an infrastructure of many cameras and projectors (as little as one camera and one projector is sufficient). Our method reliably detects the motion state, adapts the sampling parameters, and is able to use time-multiplexed codes enabling a robust active acquisition of the dynamic scene. The observed object and its motion can be played back in the original order or rearranged to produce new motion and in both cases observable from novel viewpoints. Finally, our active method uses only one camera and one projector, yet is able to acquire both dense color and dense depth samples at the same time.

There are a few current limitations. First, our method does not target acquiring general dynamic scenes. Instead, when the scene motion has repetitions, we can exploit this property. Our image-based motion state method requires almost exact repetition; thus handling more natural motion is difficult. Second, we assume moving objects remain in a compact working volume and do not have global translation and rotation. Third, our approach assumes single object motion. Multiple objects with repe-

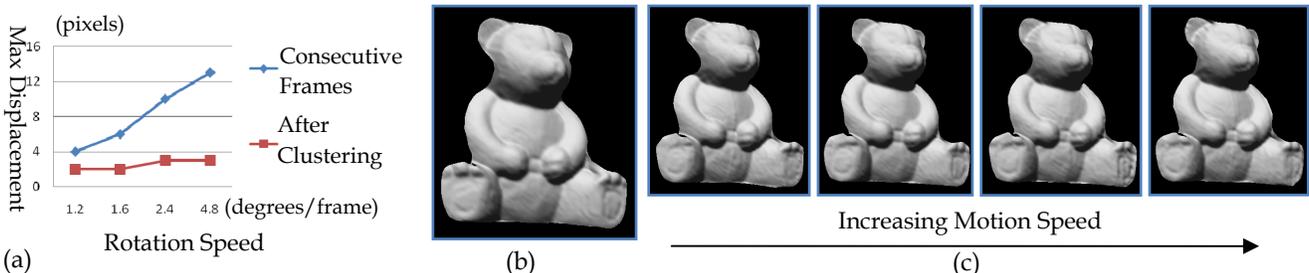


Fig. 14. Increasing Motion Speed. a) We manually find the maximum pixel movement for consecutive frames both in the captured video and in the image cluster for one motion state and for four increasing motion speeds. b) The reconstruction when the bear object is static. c) Reconstructions of the bear object rotating at the increasing speeds shown in the graph.

titive motions can be handled only if they do not occlude each other and a prior knowledge of segmentation is given. There are also some additional limitations for the unsynchronized method. The red-blue patterns cannot be used to faithfully recover dark or green-only surfaces. In the presence of cyan or yellow surfaces, the red-blue patterns degenerate to the same restrictions as the black-white Gray code. In these cases, the synchronized acquisition can be used at the cost of a little bit more complicated hardware setup and longer capturing time.

With regard to future work, we would like to investigate how to capture local repetitive motion but under global translation and rotation. One option is to use feature tracking and image warping as a means to align translating objects. Second, we would like to study the theoretical aspect of how many static views are enough to uniquely identify all repetitive motions. Third, we would like to investigate other sampling parameters, which can be adjusted when the motion state repeats to enable additional applications.

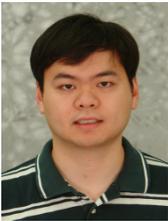
ACKNOWLEDGMENTS

This work was supported by NSF CCF 0434398 and by a Purdue Research Foundation grant. We would also like to thank the reviewers for their suggestions.

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), pp. 821-827, 2005.
- [2] C. Buehler, M. Bosse, L. McMillan, S. Gortler, and M. Cohen, "Unstructured Lumigraph Rendering", *Proc. of ACM SIGGRAPH*, pp. 425-432, 2001.
- [3] D. Bradley, T. Popa, A. Sheffer, W. Heidrich, and T. Boubekeur, "Markerless Garment Capture", *ACM Trans. on Graphics*, 27(3), 2008.
- [4] J. Carranza, C. Theobalt, M.A. Magnor, and H.P. Seidel, "Free-Viewpoint Video of Human Actors", *ACM Trans. on Graphics*, 22(3), pp. 569 - 577, 2003
- [5] D. Caspi, N. Kiryati, and J. Shamir, "Range Imaging with Adaptive Color Structured Light", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(5), pp. 470 - 480, 1998.
- [6] R. Cutler and L. Davis, "Robust Real-Time Periodic Motion Detection, Analysis, and Applications", *IEEE Trans. on IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8), pp. 781-796, 2000.
- [7] J. Davis, D. Nehab, R. Ramamoorthi, S. Rusinkiewicz, "Space-time Stereo: A Unifying Framework for Depth from Triangulation", *IEEE Trans. on Pattern Analysis and Machine Intelligence* 27(2), pp. 296-302, 2005.
- [8] E. de Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H. Seidel, and S. Thrun, "Performance Capture from Sparse Multi-View Video", *ACM Trans. on Graphics*, 27(3), article 98, 2008.
- [9] P. Einarsson, C.F. Chabert, A. Jones, W.C. Ma, B. Lamond, T. Hawkins, M. Bolas, S. Sylvania, and P. Debevec, "Relighting Human Locomotion with Flowed Reflectance Fields", *Proc. of Eurographics Symposium on Rendering*, pp. 183-194, 2006.
- [10] P. Fong and F. Buron, "High-resolution 3-dimensional Sensing of Fast Deforming Objects", *Proc. Intelligent Robots and Systems*, pp. 1606-1611. 2005
- [11] C. Frueh and A. Zakhor, "Capturing 2½D Depth and Texture of Time-varying Scenes using Structured Infrared Light", *Proc. of Intl. Conf. on 3D Imaging and Modeling*, pp. 318 - 325, 2005.
- [12] C. Hernandez, G. Vogiatzis, G.J. Brostow, B. Stenger, R. Cipolla, "Non-rigid Photometric Stereo with Colored Lights", *Proc. of Intl. Conf. on Computer Vision*, pp. 1-8, 2007.
- [13] G.J. Iddan and G. Yahav, "Three-dimensional Imaging in the Studio and Elsewhere", *Proc. SPIE Vol. 4298*, pp. 48-55, 2001.
- [14] S. Inokuchi, K. Sato, and F. Matsuda, "Range Imaging System for 3-D Object Recognition", *Proc. Intl. Conf. of Pattern Recognition*, pp. 806-808, 1984.
- [15] A. Jones, A. Gardner, M. Bolas, I. McDowall, P. Debevec, "Simulating Spatially Varying Lighting on a Live Performance", *3rd European Conf. on Visual Media Production*, pp. 127-133, 2006.
- [16] S.B. Kang, J. Webb, C. Zitnick, and T. Kanade, "A Multibaseline Stereo System with Active Illumination and Real-time Image Acquisition", *Proc. of Intl. Conf. on Comp. Vision*, pp. 88 -93, 1995.
- [17] 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 Intl. Symposium on 3D Data Processing, Visualization and Transmission*, pp. 413 - 421, 2004.
- [18] T.P. Koninckx, A. Griesser, and L. van Gool, "Real-time Range Scanning of Deformable Surfaces by Adaptively Coded Structured Light", *Proc. of Intl. Conf. on 3D Imaging and Modeling*, pp. 293 - 300, 2003.
- [19] T. Koninckx, P. Peers, P. Dutre, and L. Van Gool, "Scene-adapted Structured Light", *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 611- 618, 2005.
- [20] L. Kovar, M. Gleicher, and F. Pighin, "Motion Graphs", *ACM Trans. on Graphics*, 21(3), pp. 473 - 482, 2002.
- [21] I. Laptev, S. Belongie, P. Perez, and J. Wills, "Periodic Motion Detection and Segmentation via Approximate Sequence Alignment", *Proc. of Intl. Conf. on Comp. Vision*, pp. 816-823, 2005.
- [22] C. Lee, S. Ghyme, C. Park, and K. Worn, "The Control of Avatar Motion using Hand Gesture", *Proc. of the ACM Symp. on Virtual Reality Software and Technology*, pp. 59-65, 1998.
- [23] W. Matusik, C. Buehler, R. Raskar, S.J. Gortler, and L. McMillan, "Image-based Visual Hulls", *Proc. of ACM SIGGRAPH*, pp. 369-374, 2000.
- [24] C. Rocchini, P. Cignoni, C. Montani, P. Pinci, and R. Scopigno, "A Low Cost 3D Scanner Based on Structured Light", *Computer Graphics Forum* 20(3), pp. 299-308, 2001.
- [25] S. Rusinkiewicz, O. Hall-Holt, and M. Levoy, "Real-time 3D Model Acquisition", *ACM Trans. on Graphics*, 21(3), pp. 438 - 446, 2002.
- [26] F. Sadlo, T. Weyrich, R. Peikert, and M. Gross, "A Practical Structured Light Acquisition System for Point-based Geometry and Texture", *Proc. of Symp. on Point-Based Graphics*, pp. 89-98, 2005.
- [27] P. Sand, L. McMillan, and J. Popovic, "Continuous Capture of Skin Deformation", *ACM Trans. on Graphics*, 22(3), pp. 578-586, 2003.
- [28] A. Schödl, R. Szeliski, D. Salesin, and I. Essa, "Video Textures", *Proc. of ACM SIGGRAPH*, pp. 489-498, 2000.
- [29] S. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms", *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 519-526, 2006.
- [30] S. Seitz and C. Dyer, "View-Invariant Analysis of Cyclic Motion", *Intl. Journal of Computer Vision*, 3, pp. 231-251, 1997.
- [31] J. Starck, G. Miller, and A. Hilton, "Video-based Character Animation", *Proc. of Symp. on Comp. Animation*, pp. 49-58, 2005.
- [32] S. Vedula, S. Baker, and T. Kanade, "Image-based Spatio-temporal Modeling and View Interpolation of Dynamic Events", *ACM Trans. on Graphics*, 24(2), pp. 240-261, 2005.
- [33] M. Waschbüsch, S. Würmlin, D. Cotting, F. Sadlo, and M. Gross, "Scalable 3D Video of Dynamic Scenes", *The Visual Computer*, 21(8-10), pp. 629-638, 2005.
- [34] T. Weise, B. Leibe, L. van Gool, "Fast 3D Scanning with Automatic Motion Compensation", *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1-8, 2007.

- [35] A. Wenger, A. Gardner, C. Tchou, J. Unger, T. Hawkins, P. Debevec, "Performance Relighting and Reflectance Transformation with Time-multiplexed Illumination", *ACM Trans. on Graphics*, 24(3), pp. 756-764, 2005.
- [36] Y. Xu and D. Aliaga, "Efficient Multi-viewpoint Acquisition of 3D Objects Undergoing Repetitive Motions", *Proc. of ACM Symp. on Interactive 3D Graphics and Games*, pp. 113-120, 2007.
- [37] Y. Xu and D. Aliaga, "Dense Depth and Color Acquisition of Repetitive Motions", *Proc. of Intl. Conf. on 3D Imaging and Modeling*, pp. 141-148, 2007.
- [38] L. Zhang, B. Curless, and S. Seitz, "Rapid Shape Acquisition using Color Structured Light and Multi-pass Dynamic Programming", *Proc. of Intl. Symposium on 3D Data Processing, Visualization and Transmission*, pp. 24 - 36, 2002.
- [39] L. Zhang, B. Curless, and S. Seitz, "Spacetime Stereo: Shape Recovery for Dynamic Scenes", *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 367-374, 2003.
- [40] L. Zhang, N. Snavely, B. Curless, S.M. Seitz, "Spacetime Faces: High-resolution Capture for Modeling and Animation", *ACM Trans. on Graphics*, 23(3), pp. 548-558, 2004.
- [41] C. 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), pp. 600-608, 2004.



Yi Xu is a Ph.D. student in Computer Science at Purdue University. His research interests are in image-based modeling and rendering, 3D reconstruction, interactive computer graphics, and computational photography. He obtained a B.Eng. degree from Zhejiang University, China and a M.S. degree from University of Alberta, Canada.



Dr. Daniel G. Aliaga is an Assistant Professor of Computer Science at Purdue University. He obtained his Ph.D. degree and M.S. degree from the University of North Carolina at Chapel Hill. Dr. Aliaga's research is primarily in the area of 3D computer graphics but overlaps with visualization and with computer vision. In particular he develops novel 3D model acquisition methods, and combines his 3D acquisition methods with additional new algorithms to produce pioneering modeling and visualization frameworks. To date Prof. Aliaga has published nearly 60 peer reviewed publications and chaired and served on numerous ACM and IEEE conference and workshop committees, including being member of 25 program committees, conference chair, papers chair, invited panelist, and a frequent reviewer of papers, journal articles, and technical courses. He also holds several patents related to 3D acquisition and modeling. In addition, Dr. Aliaga has served on several NSF panels, is on the editorial board of *Graphical Models*, and is a member of ACM SIGGRAPH. His research has been whole or partially funded by NSF, Microsoft Research, and Adobe Inc.