High-Resolution Three-Dimensional Sensing of Fast Deforming Objects

Philip Fong and Florian Buron

Department of Computer Science
Stanford University, Stanford, California, USA
{fongpwf, fburon}@ai.stanford.edu

***Abstract* – In applications like motion capture, high speed collision testing and robotic manipulation of deformable objects there is a critical need for capturing the 3D geometry of fast moving and/or deforming objects. Although there exist many 3D sensing techniques, most cannot deal with dynamic scenes (e.g., laser scanning). Others, like stereovision, require that object surfaces be appropriately textured. Few, if any, build high-resolution 3D models of dynamic scenes. This paper presents a technique to compute high-resolution range maps from single images of moving and deforming objects. This method is based on observing the deformation of a projected light pattern that combines a set of parallel colored stripes and a perpendicular set of sinusoidal intensity stripes. While the colored stripes allow the sensor to compute absolute depths at coarse resolution, the sinusoidal intensity stripes give dense relative depths. This twofold pattern makes it possible to extract a high-resolution range map from each image in a video sequence. The sensor has been implemented and tested on several deforming objects.**

Index Terms* – *3D sensing, deformable objects, structured light, phase unwrapping, snake

I. Introduction

A 3D representation of the world is useful for many tasks in robotics. Considerable research has gone into developing techniques to sense the geometry of objects. In particular, range sensing has been used in many tasks including navigation and object recognition. However, such techniques have focused on rigid objects and quasi-static scenes.

In many applications, there is a need for sensing fast moving and/or deforming objects. For example, capturing the deformation of real objects could be used in surgical simulation to model human tissues and organs. Modeling fabric and rope deformation is of great interest for the video games and film industries. Another application is vehicle crash tests where analysis of the deforming geometry is of direct importance. Real-time sensing of moving and deforming objects can also be used for feedback in robotic manipulation and navigation.

These applications require a high-resolution range sensing technique that makes minimal assumptions about the type of objects sensed and the types of motions occurring in the scene. This paper describes a range sensing technique that extracts depth information in the entire field of view from single images and is suitable for untextured objects.

II. Related Work

Previous research in range sensing includes work on laser stripe scanning, stereovision, and pattern-based structured light methods. These techniques are all based on depth from triangulation, which requires at least two viewpoints. By identifying corresponding features in different viewpoints, their 3D positions can be found by intersecting rays from each viewpoint. These methods are distinguished by how they solve the correspondence problem.

Laser scanners project a single plane of light and scan across the scene. This limits their application to rigid static objects. Stereovision, which finds correspondences by matching pairs of features from two images, often fails on objects with no texture or repeating textures. It has been successfully applied to deforming textured cloth [1].

Pattern-based structured light techniques are capable of producing a dense range map of an entire scene by projecting a pattern and observing how it is changed from another viewpoint. Here finding correspondences is reduced to finding spatially and/or temporally unique features encoded into the pattern. In methods that only rely on single images, range maps can be produced at the video frame rate (in real-time or post processed). Multi-image techniques typically give higher quality range maps, but rely on temporal coherence limiting the speed of motions in the scene and often produce range maps a rate lower than the video frame rate.

Pattern-based techniques differ in the way the information is encoded in the pattern. Intensity ramp patterns [2] are sensitive to noise and reflectance variations. Techniques using Moiré gratings and sinusoids [3,4,5,6] are much more resistant to noise and reflectance variations, but requires both knowledge of the topology and points of known depth in the scene. In systems using black and white stripes [7] or colored stripes [8,9] the resolution of the range map obtained from a single image is limited by the discrete number of possible pattern encodings. Higher resolution is attained by using a temporal sequence of images and patterns. This limits their use to rigid or slowly moving objects. In contrast, the technique presented here scales with camera resolution even while using single images. Spacetime stereo [10] is a method that performs stereo matching over space and time windows on scenes illuminated by arbitrarily varying patterns. Again, the reliance on temporal coherence limits the types of motion it can sense.

In [11], a series of vertical stripes are used with a few diagonal stripes to disambiguate them. The technique presented here also uses a pattern with repeating features combined with additional unique features.

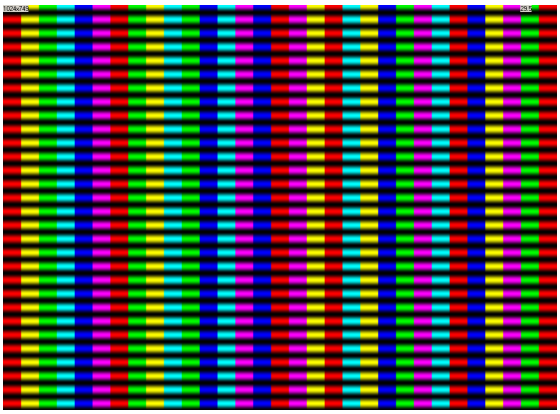

Fig. 1 Projected pattern

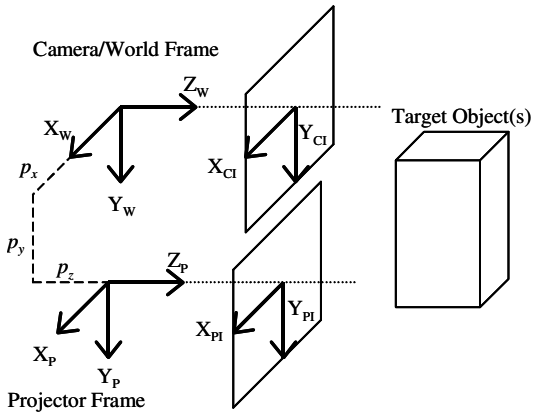

Fig. 2 System geometry

III. Overview

A. Basic Operation

The technique described here combines features of color stripe and sinusoidal triangulation systems. We use color stripes to get sparse but absolute depth values and a sinusoid to propagate these values and get a high-resolution range map. To guide the propagation we segment the image into regions that do not contain any depth discontinuities.

Compared to patterns used in previous work, which employ either color or luminance alone, this pattern is more robust because the color and sinusoidal components provide complementary information. The colored stripe boundaries give sparse, but absolute depths, while the sinusoid gives dense relative depths. These points are used as known points in the sinusoid processing. The sinusoidal component is resistant to reflectance variations in the scene and only a few correctly sensed color transitions (also known as color edges) are needed as start points. A sinusoidal pattern also avoids the need to do any sub-pixel location estimation of pattern features (e.g. transitions). The depths are computed at pixel locations.

A pattern of vertical colored stripes with a horizontal sinusoid overlaid (see Fig. 1) is projected on to the scene by a standard LCD projector and images are captured from a different viewpoint with a color camera. The sinusoid of intensity is embedded in the value channel and the colored stripes are embedded in the hue channel in HSV color space.

Each frame can be processed independently to recover depth information. Pixels are given color labels and scores by a Bayesian classifier. The stripe transitions are identified, labeled, and scored according to the color labels of pixels adjacent to each boundary. The scene in segmented into regions using a snake technique [12] and the luminance component in each region is processed to recover the dense depth map using the color transition depths as starting points.

Processing each frame independently eliminates the need for temporal coherence. The scene can change significantly between frames without affecting the depth recovery.

B. System Geometry

Fig. 2 shows the geometric configuration and coordinate frames of the camera and projector. The world coordinate frame, W, coincides with the camera coordinate frame which is centered at the camera lens optical center with X_W pointing to the right, Y_W pointing down, and Z_W pointing out of the camera lens along the optical axis. The projector frame, P, is centered at the projector lens optical center which is located at ${}^{W}(p_x, p_y, p_z)$ and the axes are aligned with those of the world frame. In practice, the optical axis of the projector and camera may not be parallel but rectifying the projected pattern makes a parallel virtual axis. In fact, it is often useful for them to be verging together at a point near the object of interest.

Points in the world are mapped onto the projector and camera normalized virtual image planes (located at z=1) using a pinhole lens model:

$$
\begin{aligned}
{}^{CI}x &= \frac{{}^{W}x}{{}^{W}z}, \quad {}^{CI}y = \frac{{}^{W}y}{{}^{W}z} \\
{}^{PI}x &= \frac{{}^{P}x}{{}^{P}z}, \quad {}^{PI}y = \frac{{}^{P}y}{{}^{P}z}
\end{aligned}
\tag{1}
$$

Note that ${}^{CI}(x,y) = {}^{W}(x,y,1)$ and ${}^{PI}(x,y) = {}^{P}(x,y,1)$.

The mapping from real image plane coordinates to the normalized virtual coordinates is done with a projection model that accounts for lens distortion [13] and by warping the projected pattern and captured images.

Computing the range map is then the same as computing the z value for each pixel on the image plane of the camera.

IV. Computing Depth

A. Depth from Colored Stripes

The absolute depths of points in the scene can be found by detecting the color stripe transitions in the acquired image and triangulating with the locations of the stripe transitions in the projected pattern.

The color part of the projected pattern consists of a set of colors, $C=\{c_0,\ldots, c_n\}$ arranged into a sequence $(b_0, \ldots, b_m)$ $b_i \in C$ of vertical stripes such that:

1. $b_j \neq b_{j+1}$, i.e. consecutive colors are different.
2. $b_j \neq b_k$ or $b_{j+1} \neq b_{k+1}$ for $j \neq k$, i.e. each transition appears once.

Using more colors leads to a greater number of possible transitions, but the colors are more difficult to distinguish. In our case, since the sinusoid is used to get the dense range map, having a large number of transitions is not critical. In the results presented in this paper, six colors were used allowing 30 unique transitions.

In the projected pattern, each color transition uniquely identifies a vertical plane. On each scanline (constant ${}^{CI}y$), if a transition is found at ${}^{CI}x$ and corresponds to the transition located at ${}^{PI}x$ in the projected pattern, the z value of the point in the scene corresponding to the pixel ${}^{CI}(x,y)$ is:

$$ {}^{W}z({}^{CI}x,{}^{CI}y) = \frac{{}^{W}p_X - {}^{PI}x\,{}^{W}p_Z}{{}^{CI}x - {}^{PI}x}. \quad (2) $$

B. Recovering the Color Transitions

In captured images, a Bayesian classifier determines the colors of the pixels. The probability distributions are estimated by fitting gaussians to training images. The training images for the classifier are images of the scene with a projected "pattern" of only one color. The images are converted to HSV space giving hues $h(x,y)$, saturations $s(x,y)$, and values $v(x,y)$. For each color, c, the mean, μ_c, and variance σ_c of the hues are computed and stored. Colors in the pattern should be chosen so that the overlap in the gaussians is minimized. Our technique is robust to incorrectly classified pixels because all transitions detected in a region are aggregated in phase unwrapping (see section E).

A captured image is processed one horizontal line at a time. There are several techniques to locate color edges, such as the consistency measure used in [8]. A simpler method is to look at the difference in hue between consecutive pixels. For each group of pixels whose absolute hue difference (taking into account the circular nature of hue) exceeds a set threshold, pick the pixel with the maximum difference to be the location of a transition.

Once an edge is found, it is given a label and score based on the color scores of windows of pixels on each side of the transition. For an edge at pixel k, color score of the left and right pixel windows are:

$$ LCS_c = \sum_{x=k-win}^{k} \frac{CS_c(x,y)}{\sum_{h\in C} CS_h(x,y)},\ RCS_c = \sum_{x=k+1}^{k+win} \frac{CS_c(x,y)}{\sum_{h\in C} CS_h(x,y)}. \quad (3) $$

The color score of a pixel is the estimated probability of observing the hue given that the projected color is c:

$$ CS_c(x,y) = \frac{1}{\sigma_c\sqrt{2\pi}} e^{-(h(x,y)-\mu_c)^2/(2\sigma_c^2)}. \quad (4) $$

The colors of the left and right windows are chosen to be:

$$ lcolor = \underset{c\in C}{\operatorname{argmax}}\ LCS_c,\ rcolor = \underset{c\in C}{\operatorname{argmax}}\ RCS_c. \quad (5) $$

If ($lcolor$, $rcolor$) is consistent with an edge in the projected pattern (b_i=$lcolor$ and b_{i+1}=$rcolor$ for some i) and LCS_C and RCS_C are greater than a set threshold, the edge is labeled with the x location of that edge in the projected pattern, ${}^{PI}x$, and given a score, ES=LCS_c + RCS_c. Validating against the projected pattern eliminates edges detected as a result of noise and reflectance variations in the scene.

C. Depth from Sinusoids

The depth at each pixel is computed from the apparent phase shift between the projected and observed sinusoids.

On the projector image plane, the luminance of the projected pattern is:

$$ v({}^{PI}x,{}^{PI}y) = A\cos(\omega\,{}^{PI}y) + C \quad (6) $$

where ω is the frequency. A and C are chosen to maximize the use of the projector's dynamic range and ω is set smaller than the camera's and projector's Nyquist frequencies. Converting to world coordinates and projecting into the world yields:

$$ v({}^{W}x,{}^{W}y,{}^{W}z) = A\cos\left(\omega\frac{{}^{W}y - {}^{W}p_Y}{{}^{W}z - {}^{W}p_Z}\right) + C. \quad (7) $$

The camera sees on its image plane:

$$ v({}^{CI}x,{}^{CI}y) = R({}^{W}x,{}^{W}y,{}^{W}z({}^{CI}x,{}^{CI}y)) \times \cos\left(\omega\frac{{}^{W}z({}^{CI}x,{}^{CI}y)\,{}^{CI}y - {}^{W}p_Y}{{}^{W}z({}^{CI}x,{}^{CI}y) - {}^{W}p_Z}\right) + N({}^{W}x,{}^{W}y,{}^{W}z({}^{CI}x,{}^{CI}y)) \quad (8) $$

where $R(x,y,z)$ is due to the reflectance of the surface at (x,y,z) and $N(x,y,z)$ is due to noise and ambient light. Computing the depth map is recovering ${}^{W}z({}^{CI}x,{}^{CI}y)$. Eq. (8) gives:

$$ v({}^{CI}x,{}^{CI}y) = A\cos\left(\omega\,{}^{CI}y - \theta({}^{CI}x,{}^{CI}y)\right), \quad (9) $$

$$ \theta({}^{CI}x,{}^{CI}y) = \omega\left(\frac{-{}^{W}p_Z\,{}^{CI}y + {}^{W}p_Y}{{}^{W}z({}^{CI}x,{}^{CI}y) - {}^{W}p_Z}\right). \quad (10) $$

In this formulation, θ can be seen as modulating the phase or frequency of the original sinusoid. We will refer to it as the phase. $\theta_w = \arctan 2(\sin(\theta),\cos(\theta))$, the wrapped version of θ, can be recovered by multiplying $v({}^{CI}x,{}^{CI}y)$ by $\cos(\omega\,{}^{CI}y)$ and low-pass filtering (digital demodulation) [4,5]. θ is recovered from θ_w by phase unwrapping as described in section E.

Given $\theta({}^{CI}x,{}^{CI}y)$, ${}^{W}z({}^{CI}x,{}^{CI}y)$ is given by:

$$ {}^{W}z({}^{CI}x,{}^{CI}y) = \frac{\omega(-{}^{W}p_Z\,{}^{CI}y + {}^{W}p_Y)}{\theta({}^{CI}x,{}^{CI}y)} + {}^{W}p_Z \quad (11) $$

The 3D locations of the points are found by projecting back into the world coordinate frame.

D. Segmentation Using Snakes

The depth cannot be computed from the θ_w found with the sinusoid pattern processing because we need $\theta = \theta_w + k2\pi$ to compute the depth and the integer k cannot be recovered from the sinusoid. This ambiguity is due to the periodic nature of a sinusoid. However, if the phase (or the depth) of one pixel is known (for example from using the color stripes), we can assume that the adjacent pixels have the same offset ($k2\pi = \theta - \theta_w$) and compute the θ value of the adjacent pixels. Performing this propagation is called phase unwrapping [14].

The main assumption made in phase unwrapping is that between two adjacent pixels along the unwrapping path the phase changes by less than 2π. So, we need to segment our image in regions such that in each region the difference in θ between two adjacent pixels is always less than 2π.

There exist various techniques to segment images. We chose the snake (or active contour) technique [12]. A snake is a closed contour, which is deforming to fit the borders of the region we want to segment. Each point of the snake is moved iteratively based on the various forces applied to it until it reaches an equilibrium. This technique is known to give good

results for many problems and has the advantage of being able to bridge across small gaps in the boundary of a region we want to segment. This property is very important in our case.

We decided to base the segmentation on the phase variance (defined in [14]) of the image. The phase variance is a measure of how fast the phase of a pixel changes compared to its neighbors. A large change in phase corresponds to a discontinuity in depth and is interpreted as a boundary in our image. Phase variance has been effectively employed in phase unwrapping for applications such as synthetic aperture radar and magnetic resonance imaging [14].

To start the segmentation we initialize the snake to a small circle centered randomly in the yet unsegmented region of the phase variance image. We then adjust the location of the circle so that the circle covers a uniform area of small phase variance values. Notice the tradeoff on the size of the initial circle: a large circle will guarantee that we are starting inside a uniform region, but will prevent us from segmenting small or narrow regions.

Then, each snake point moves under the influence of three forces applied to it until equilibrium is reached:

- An *expansion* force: Contrary to most snakes in the literature, our snake is expanding. Each snake point P is subject to a constant force pointing toward the outside of the region encircled by the snake and orthogonal to the tangent at P.
- An *image* force: The snake expansion is locally stopped when it reaches regions of high phase variance. The force is computed by taking the gradient of the phase variance. To avoid having discontinuities in this force (due to the pixelization of the phase), it is interpolated on the four pixels surrounding the current snake point.
- A *rigidity* force: Each snake point is subject to a force that resists high contour curvatures preventing the snake from flooding into other regions through the small gaps.

Besides these forces, other events affect the deformation of the snake. Snake points are added or removed when they get too far or too close to each other. The snake is not allowed to self intersect. Regions that do not contain any detected color edges are merged with neighboring regions.

Since this segmentation is based only on phase variance, it could fail if there is a long depth discontinuity that corresponds to a phase difference of a multiple of 2π. But this is very unlikely to happen because the depth difference corresponding to a phase difference of 2π varies with position. We have never encountered this problem in our experiments.

E. Phase Unwrapping

Before unwrapping begins, guesses for θ are computed from the detected color edges by applying (2) and (10) to each edge. The score of each guess is the score, *ES*, of the edge.

The unwrapping of each region starts at the center of the initial circle of the snake. Then the pixel with the lowest phase variance among pixels that border unwrapped pixels is unwrapped. This proceeds until the entire region is unwrapped in a way similar to the quality map algorithm in [14].

Once a region is unwrapped, the offset for the region needs to be computed from the guesses encountered in the region. We have θ_u, which is offset from θ by an integral multiple of 2π.

$$\theta(x, y) = \theta_u(x, y) + 2\pi k \tag{12}$$

k is computed by rounding to the nearest integer the weighted median of the difference between encountered guesses and θ_u divided by 2π. Using the weighted median provides robustness against bad guesses from misclassified colors or falsely detected edges.

V. Experimental Results

The results shown here were obtained with a Viewsonic PJ551 LCD projector at a resolution of 1024x768. Images were captured using a Basler A602fc camera at 640x480 @ 60 fps. A pattern similar to Fig. 1 was used. In each case, the colors were adjusted as described in section IV.B and ω was selected based on the distance the object of interest was from the sensor.

Calibrating a projector-camera system is more difficult than calibrating a dual camera system. Standard camera calibration techniques require known 3D points in world space and finding corresponding points in the image plane [13]. In projector calibration, the known points are on the image plane and the corresponding points are found in the world. This often leads to many more sources of error in the calibration. In this work, the 3D world coordinates were found by calibrating the camera first and using the camera to locate the points of a projected pattern overlaid on a planar checkerboard pattern.

Fig. 3. Speaker. Photo (left), video frame (mid), and corresponding reconstructed 3D model

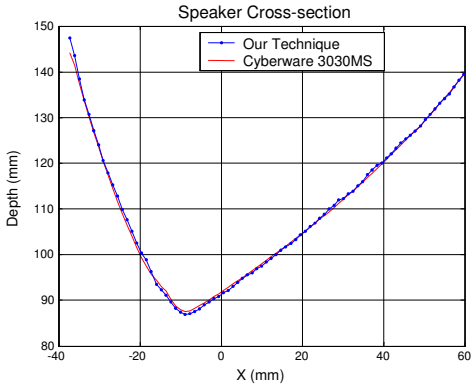

Fig. 4 Speaker cross-section corresponding to red line in Fig. 3.

A. Moving Speaker Sequence

We captured a sequence showing a computer speaker being rotated and translated. The speaker was about 65 cm from the sensor. Fig. 3 shows a photograph of the speaker, a video frame, and its corresponding 3D reconstruction. Fig. 4 shows a cross section of the speaker along the line marked in red in Fig. 3. For reference, a scan from a Cyberware 3030MS laser scanner is also shown. The range map shows the bend in the left side and the hole on the front. Note that no surfaces on the real speaker are flat. There are small ridge artifacts along colored stripe boundaries. These are likely the result of misalignment of the subpixel color elements in the projector and chromatic aberration. The ridges on the front appear to be a Moiré like pattern caused by interaction between the pattern and speaker mesh surface. In all cases, the ridges and noise are less than +/- 1 mm from the surface. The root mean squared error is about 0.7 mm.

B. Deforming Foam

We constructed a simple deforming object consisting of a piece of soft foam held in place by a more rigid piece of foam. The foam was deformed from the backside by pulling on wires attached to the foam. Fig. 5 shows a photograph of the foam and a frame from a sequence captured while deforming the foam at about 70 cm from the sensor. The sensor is able to correctly separate the deforming foam from the background foam. The rougher appearance of the holder foam is due to the actual surface being rougher and porous which caused subsurface scattering. Also, notice that the colors in the stripes look different when reflected by two types of foam. The sensor is able to deal with this through a combination of the robustness of Bayesian color classifier and the aggregation of the guesses to compute the offset.

C. Waving Flag

Fig. 6 shows six frames from a sequence of a waving flag. A piece of fabric attached to a wooden board was waved in a side to side motion. The noise is larger than in the foam and speaker sequences because in order to fit the flag in the field of view it was about 130 cm from the sensor. In frames 127, 131, and 147 the wooden board is moving leftward and in 139 and 143 it is moving rightward. In 135, one can see the folds in the fabric making a C-like shape, which is a result of the board changing directions.

D. Discussion

The range maps obtained from our sensor are very good for a single frame technique, showing the effectiveness of a combination of a discrete color pattern and continuous sinusoidal intensity pattern.

Using a static pattern allows the range map frame rate to exactly match the camera frame rate. Systems using pattern sequences are limited by the speed of both the projector and camera. Cameras faster than 500 fps are commercially available, while it is difficult to find projectors over 60 fps. Additionally, in cases where adapting the projected pattern to the scene is unnecessary, the LCD projector can be eliminated potentially saving cost and weight.

However, the practical implementation has a few drawbacks. The unwrapping process requires at least one correct phase guess in each region. The colored stripe process often produces many more than needed. But in objects that have strongly saturated colors there will be fewer guesses making the system more sensitive to segmenting errors. For example, the sensor will have difficulties on a bright red object. This can be mitigated by choosing the right set of colors in the pattern in some cases. However, if the scene contains many saturated colors it then could be very difficult to find a good set of pattern colors.

The use of color imposes stricter constraints on the quality of the camera, projector, and lenses used. Color misalignment occurs from misalignment of the different sub-pixel color elements in the projector and chromatic aberration in the camera and/or projector lenses. In many cameras (including ours), color is sensed by a Bayer pattern sensor which reduces the effective resolution. This sometimes leads to a darker region along color stripe boundaries. The colored stripe processing is robust to some color misalignment because the results are only used as guesses for the phase unwrapping and only need to be within 2π of the correct result. However, if the misalignment causes the sinusoidal pattern to be shifted it will result in a systematic error in the depth. This is especially evident when the shift varies in the different colored stripes causing ridges to appear in the range map. These effects may be mitigated by a camera/projector model and calibration that takes into account differences for each color sub-component in the camera and projector [15].

Finally, this system is subject to a limitation common to

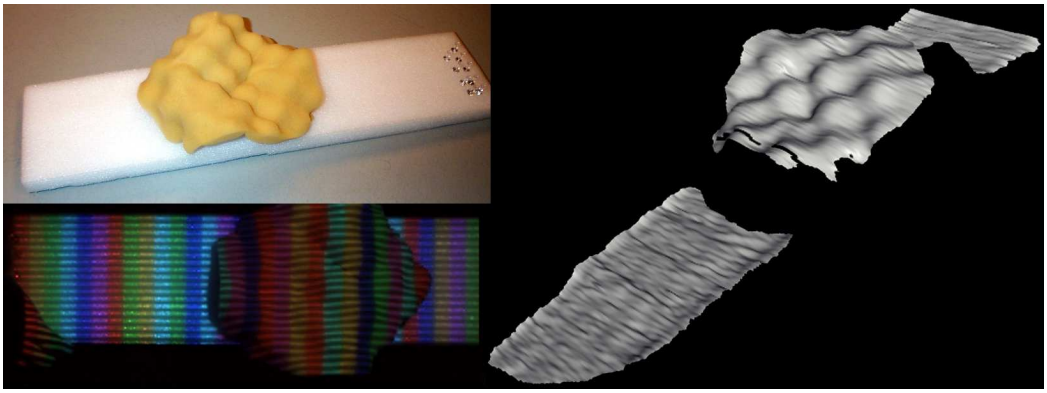

Fig. 5 Deforming foam. Photo (left top), video frame (left bottom), and 3D reconstruction (right).

all structured light approaches: the projected pattern must be brighter than the ambient light.

VII. Conclusion

The proposed system produces good quality dense range maps of scenes composed of moving and deforming objects at camera frame rates by using a pattern that combines color and luminance information. It makes no temporal coherence assumption and a reduced spatial coherence assumption. The spatial resolution scales with camera resolution and the temporal resolution depends on camera speed and not on projector speed.

At this stage, the sensor can still be improved, but works well enough to begin exploring some application areas. In particular, we are investigating how to extract from a sequence of range maps a moving and deforming mesh. A dynamic mesh representation is potentially more efficient and useful than a series of range maps.

Acknowledgments

This work is supported in part by in part by NIH grant R33 LM07295 and NSF grant ACI-0205671. We would like to thank Jean-Claude Latombe for his support, assistance, and suggestions, James Davis for his invaluable help and discussions, Ken Salisbury for his support and inspiring some potential applications, Justin Durack for implementing the Bayesian classifier, and Keenan Wyrobeck for his assistance in constructing the experimental system.

References

[1] D. Pritchard and W. Heidrich. "Cloth Motion Capture," *Computer Graphics Forum (Eurographics 2003),* 22(3):263-271, September 2003.

[2] B. Carrihill and R. Hummel. "Experiments with the intensity ratio depth sensor," *Computer Vision, Graphics, and Image Processing,* 32:337-358, 1985.

[3] M. Takeda and M. Kitoh. "Fourier transform profilometry for the automatic measurement of 3-D object shape," *Applied Optics*, 22:3977-3982, 1983.

[4] S. Tang and Y. Y. Hung. "Fast Profilometer for the Automatic Measurement of 3-D Object Shapes," *Applied Optics*, 29:3012-3018, 1990.

[5] G. Sansoni, L. Biancardi, F. Docchio, and U. Minoni. "Comparative Analysis of Low-Pass Filters for the Demodulation of Projected Gratings in 3-D Adaptive Profilometry," *IEEE Tr. Instrumentation and Measurement*, 43:50-55, 1994.

[6] P. S. Huang, Q. Hu, F. Jin, and F. P. Chiang. "Color-encoded digital fringe projection technique for high-speed 3-D surface contouring," *Optical Engineering*, 38(6), 1065-1071, 1999.

[7] G. Sansoni, M. Carocci, and R. Rodella. "Calibration and performance evaluation of a 3-D imaging sensor based on the projection of structured light," *IEEE Tr. Instrumentation and Measurement*, 49(3):628-636, 2000.

[8] L. Zhang, B. Curless, and S. M. Seitz. "Rapid Shape Acquisition Using Color Structured Light and Multi-pass Dynamic Programming. *Proc. 1st Int. Symp. on 3D Data Processing, Visualization, and Transmission*," Padova, Italy, June 19-21, 2002, 24-36.

[9] D. Caspi, N. Kiryati, and J. Shamir. "Range Imaging with Adaptive Color Structured Light. *IEEE Trans. Pattern Analysis and Intelligence*," 20:470-480, 1998.

[10] J. Davis, R. Ramamoothi, S. Rusinkiewicz. "Spacetime Stereo: A Unifying Framework for Depth from Triangulation," *IEEE Conf. on Computer Vision and Pattern Recognition*, 2003.

[11] T. P.Koninckx, A. Griesser, and L. Van Gool. "Real-time Range Scanning of Deformable Surfaces by Adaptively Coded Structured Light," *Fourth Int. Conf. on 3-D Digital Imaging and Modeling - 3DIM03*, 293-300, October 6-10, 2003.

[12] M. Kass, A. Witkin, D. Terzopoulos. "Snakes, Active Contour Models," *Int. J. Computer Vision*, 1:321-331, 1988.

[13] J.Y. Bouguet. *Camera Calibration Toolbox for Matlab.* http://www.vision.caltech.edu/bouguetj/calib_doc/index.html, 2004.

[14] D. Ghiglia, M. D. Pritt. *Two-dimensional Phase Unwrapping: Theory, Algorithms, and Software*. New York: Wiley, 1998.

[15] P. Fong, F. Buron. "Sensing Moving and Deforming Objects with Commercial Off the Shelf Hardware," *IEEE Int. Wrkshp on Projector-Camera Systems (PROCAMS)*, San Diego, June 2005.

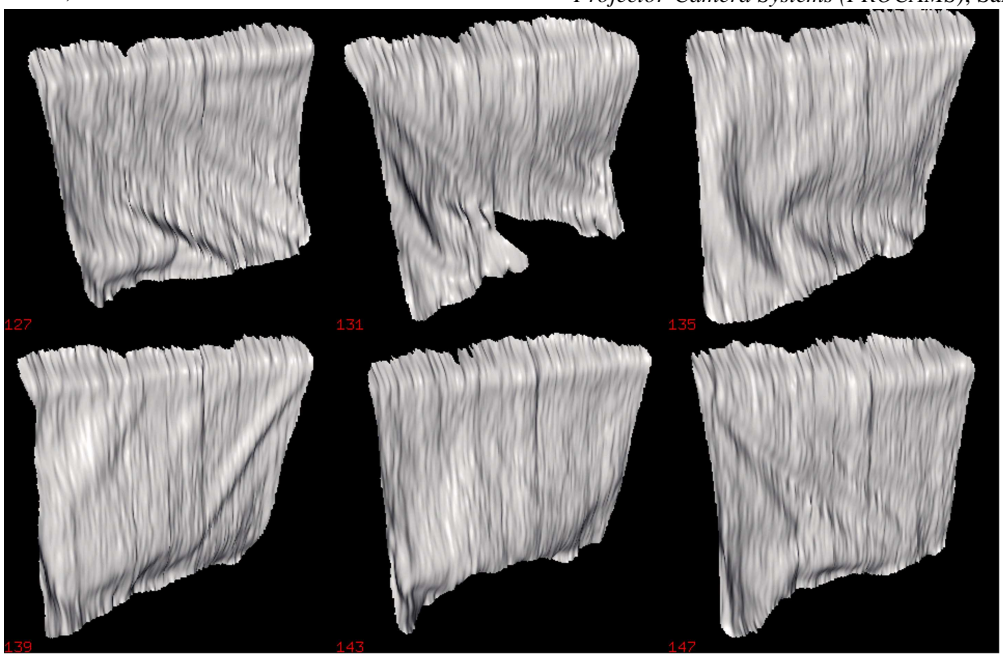

Fig. 6 Frames 127, 131, 135 (top row), 139, 143, and 147 (bottom row) of a waving flag.