Capturing 2½D Depth and Texture of Time-Varying Scenes Using Structured Infrared Light

Christian Frueh and Avideh Zakhor

Department of Computer Science and Electrical Engineering
University of California, Berkeley

{frueh,avz}@eecs.berkeley.edu

In this paper, we describe an approach to simultaneously capture visual appearance and depth of a time-varying scene. Our approach is based on projecting structured infrared (IR) light. Specifically, we project a combination of (a) a static vertical IR stripe pattern, and (b) a horizontal IR laser line sweeping up and down the scene; at the same time, the scene is captured with an IR-sensitive camera. Since IR light is invisible to the human eye, it does not disturb human subjects or interfere with human activities in the scene; in addition, it does not affect the scene's visual appearance as recorded by a color video camera. Vertical lines in the IR frames are identified using the horizontal line, intra-frame tracking, and inter-frame tracking; depth along these lines is reconstructed via triangulation. Interpolating these sparse depth lines within the foreground silhouette of the recorded video sequence, we obtain a dense depth map for every frame in the video sequence. Experimental results corresponding to a dynamic scene with a human subject in motion are presented to demonstrate the effectiveness of our proposed approach.

Keywords: dynamic scene, motion capture, structured light

I. INTRODUCTION

The 4D capture of time-varying, dynamic scenes is useful for applications such as scene analysis, motion analysis, human action recognition, and interactive scene viewing. There are several approaches to acquire appearance and/or geometry of a scene from multiple angles using a set of surrounding cameras. [5, 6] propose image-based approaches for visualization that do not explicitly capture geometry. Among others, [3, 10, 11, 13] reconstruct voxel-based or surface-based scene representation based on space carving, shape-from-silhouette, stereo matching or combinations thereof. Used in conjunction with body models, these approaches have been successful for human motion estimation and analysis, aiming to replace the inconvenient but still widespread use of markers. However, they require (a) a large number of precisely calibrated and registered cameras, (b) the capability of recording and storing multiple video streams simultaneously, and (c) special scene background; in addition, they make restrictive assumptions about the object in the scene.

Other approaches attempt to directly acquire depth from one viewpoint at video frame rate. [4] offers a commercial laser-based system which provides both color and depth at video frame rate, but is currently quite expensive. [7, 9, 12] use projector/camera configurations and structured color light to reconstruct depth. While these off-the shelf components are inexpensive, and the depth reconstruction is fast and sufficiently accurate, these systems have the inherent disadvantage that the projected visible pattern interferes with the texture/appearance acquisition; thus these approaches are not applicable to photo-realistic scene reconstruction. Moreover, the visible light pattern could potentially distract or disturb humans or animals in the scene or change their behavior; for some applications, e.g. military surveillance, it is desirable to capture the 3D motion of a scene in a covert manner.

In this paper, we propose architecture and associated algorithms for a system that is capable of capturing 3D depth of a time-varying scene using structured infrared (IR) light. Since IR light is invisible to the human eye, our capture process is done in a minimally disturbing fashion. At the same time, the visual appearance of the scene is captured by a visible-light (VIS) color camera insensitive to the IR light; we propose algorithms to enhance the recorded VIS video stream with a 2½D depth ("z") channel. In doing so, we combine advantages of passive systems, i.e. minimal interference and proper texture acquisition, with advantages of active systems, i.e. low calibration effort and robustness to absence of features.

The outline of this paper is as follows: Section II introduces the setup of our acquisition system, and Section III describes the IR line detection and identification procedure. In Section IV, we describe the algorithms for reconstruction of a dense depth map, and in Section V, we show results for a video sequence of a moving person.

II. SYSTEM SETUP

We propose an active acquisition system, in which the 3D depth estimation is performed entirely in the IR domain, and thus invisible to the human eye. In this system, shown in Figure 1, an invisible static pattern of equally spaced vertical IR stripes is projected onto the scene, and is captured by an IR sensitive camera at a different angle. Our approach is to reconstruct the depth along the resulting vertical intensity discontinuities (V-lines) in the IR image via triangulation.

Since there are no off-the-shelf IR "color" cameras available commercially, i.e. cameras that distinguish between different IR wavelengths, it is not possible to exploit color-coding techniques to identify and distinguish individual lines from each other. Rather, we use an additional horizontal IR laser stripe sweeping the scene up and down in a vertical manner; this stripe is easy to identify because it is the only horizontal line in the captured IR image. Because we know the vertical displacement between the rotating mirror and the camera, and because we can determine the current plane equation for the sweeping light plane for each frame using a reference object as described below, we can compute the depth along this horizontal line (H-line) via triangulation. We then compute the plane equation for the perpendicular V-lines that intersect with the H-line by using the 3D coordinates of the intersection point, the center-of-projection of the IR light projector, and exploiting the fact that the light planes are vertical.

Sweeping the H-stripe periodically up and down the scene along the vertical direction, we obtain the depth along a horizontal line at different locations in different IR frames, resulting in a different set of identified V-lines for each frame; in doing so, almost all V-lines are likely to be identified at some point in time. For the V-lines that do not intersect with the H-line in a given frame, we utilize the plane equation from a previous or future frame. Specifically, for each unidentified V-line, we search the previous or future frame for an identified line at close proximity. This is possible because the plane equations are constant, and the motion between 2 consecutive frames, i.e. within 33 ms, is assumed to be small. In this fashion, the plane equation for each V-line is either determined directly via intersection points with the H-line, i.e. intra-frame line tracking, or indirectly by carrying over the plane equation across frames, i.e. inter-frame line tracking. In Addition, as will be explained shortly, some of the remaining V-lines can be identified via line counting, i.e. by using the light plane equation of neighboring lines.

Figures 1 and 2 show the main components of our system: We create the static vertical stripe (V-stripe) IR pattern using a 500W light bulb and a stripe screen in front of it. For IR camera, we use an off-the-shelf digital camcorder in 0-lux ("nightvision") mode with an additional infrared-filter, and record the IR video stream at 30 Hz in progressive scan mode. The visible camera is mounted next to the IR camera, and is connected to the PC via a FireWire cable. It is triggered by every third pulse of the camcorder's video sync signal; thus, it captures the visible appearance of the scene at 10 Hz. The horizontal stripe is generated by a 30 mW IR line laser with a 90 degree fan angle, and is swept across the scene at about 2 Hz using a rotating polygonal mirror. For simplicity in our current experimental setup, we do not use an angular encoder in order to obtain the precise orientation of this horizontal light plane; rather, we compute its plane equation by 'reversing' triangulation, i.e. by using the pre-calibrated depth of a vertical reference strip to the left side of the scene.

Due to its off-the-shelf components, the overall cost of our system is low, and the required pre-calibration besides the reference strip is limited to the camera parameters, and the two baselines, namely the vertical displacement between the IR camera and the polygonal mirror, and the horizontal displacement between the IR camera and the light source for the V-stripes. As we will see shortly, while the stripe pattern does not need to be calibrated, we do assume it to be vertical; we achieve this by mounting the screen using a water gauge and gravity.
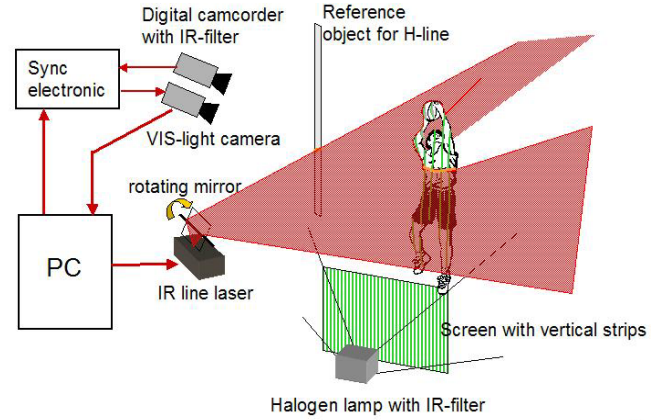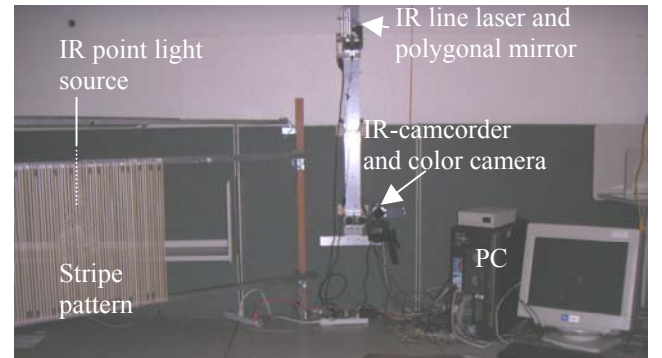

Figure 1: System setup

Figure 2: Acquisition system

III. IR-LINE DETECTION AND IDENTIFICATION

In this section, we describe the extraction of basic features such as the horizontal line, vertical lines, and the silhouette from the IR video sequence. The goal is to identify the individual patterns and distinguish them from each other.

A. H-Line Detection

The horizontal line could potentially be detected by applying a standard horizontal edge filter on each IR frame. However, there are two main problems: (1) the H-line is typically 7 to 8 pixels wide, due to the finite exposure time and its fast-sweeping motion up and down the scene. This problem can be solved by designing a filter for maximal response to an 8 pixels wide horizontal edge. (2) on thin horizontal objects such as bookshelves and fingers, and on complicated surfaces such as wrinkles of a shirt, vertical

stripes can appear as horizontal edges and result in a significant response to the horizontal filter, as seen in Figure 3 for wrinkles in the shirt and on the collar.
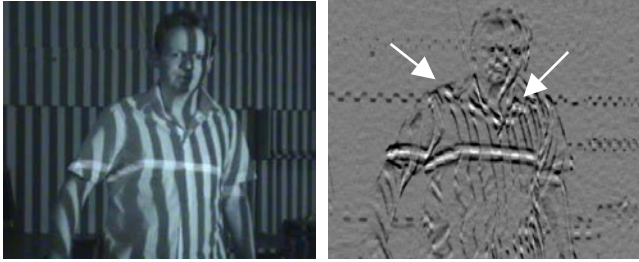

Figure 3: Vertical stripes can result in a response to the horizontal edge filter

However, "true" H-line pixels do not appear at the same image location in consecutive frames due to the sweeping motion of the H-line, while "false" ones tend to be at the same location due to the limited motion of objects in the scene; thus we apply an XOR operation between the H-pixels found in consecutive frames to eliminate false edges. Furthermore, since false H-lines are typically short, we remove all short H-line segments below a threshold length to obtain the final H-line pixels, as shown in Figure 4.

Figure 4: Detected H-line, marked red on the object and green on the reference stripe.

The plane equation for the H-line is determined using the reference strip. Specifically, the 3D coordinates of the H-line points on the reference strip can be computed from the ray of the corresponding IR camera pixel and the known depth; this point and the location of the polygonal mirror is enough to determine the plane, because it is parallel to the horizontal axis. Then, the 3D coordinates and depth of each H-line pixel is computed as the intersection point between the computed light plane and each pixel's ray.

B. V-Line Detection

V-lines are detected by applying a standard vertical edge filter on each IR frame, thinning the obtained edges and tracking them within the frame. However, it is conceivable for two separate V-lines, one on the background and one on the foreground, to align by coincidence, and hence be tracked as a single one. To avoid this, and also to reduce processing time, we clip the obtained edges to the IR foreground silhouette. We assume a fixed but possibly complex scene background, rather than imposing severe restrictions such as a special color as it is commonly done in silhouette-based techniques. To determine the silhouette, we compute the difference between each frame and the original background image acquired without any objects, and then threshold it; in order to cope with noise and objects that by coincidence have a similar gray value, we apply median filtering, segment marked areas, and remove isolated small segments. Since the background difference is small for dark stripes, the outline of the silhouette appears somewhat jagged; as it turns out, this does not cause any problems for clipping the V-lines because it essentially occurs between the V-lines. Furthermore, due to its difference with the original background, the shadow created by a person moving in the stripe pattern is typically also detected as foreground; this is not of concern either because it does not contain any V-lines. Figure 5 shows steps for the V-line detection and clipping procedure.
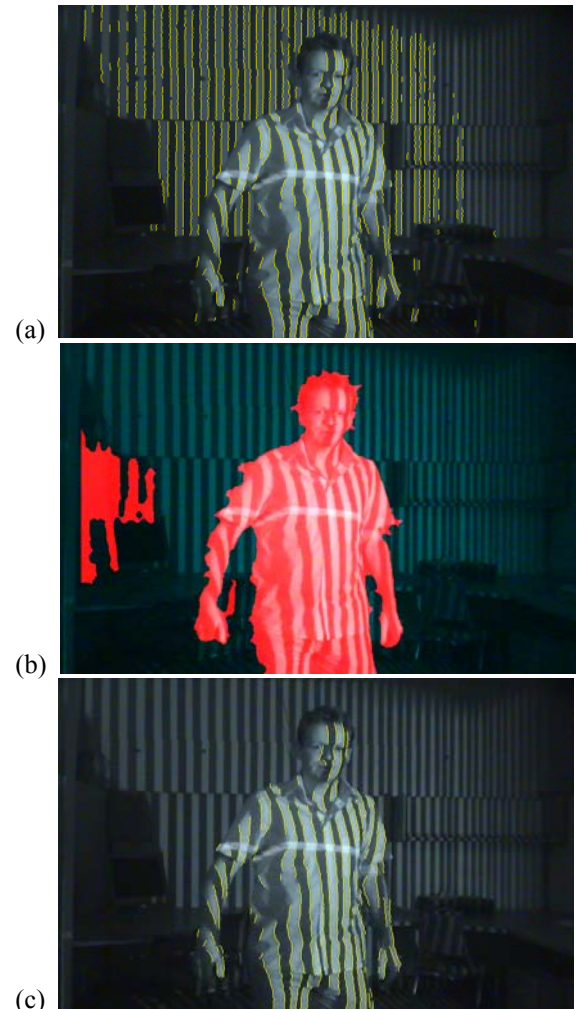

(a)

(b)

(c)

Figure 5: (a) Detected V-lines; (b) Detected IR foreground silhouette shown in red; (c) V-lines clipped to the foreground silhouette.

C. V-Line Identification

Rather than using calibrated vertical light planes and identifying them with an ordinal number, we identify a V-line in the image directly by its plane equation. This has two advantages: (a) no precise pre-calibration step of the IR-stripes is necessary, since their corresponding plane equation is obtained on-the-fly using the H-line, and (b) there is no possibility of assigning an incorrect ordinal number and thus an incorrect pre-calibrated plane to a stripe. Conceptually, the depth originally obtained from the H-line is locally updated; in this sense, it is similar to video coding, with the H-line corresponding to an Intra-frame (I-frame), and the V-line movements, i.e. incremental depth changes, corresponding to Prediction-frames (P-frames).

We define a coordinate system aligned with the IR camera, with the camera's center-of-projection as origin, the x-axis horizontal, the y-axis vertical, and the z-axis pointing into the image. Since V-planes are all vertical, i.e. the y-component n_y of their normal vector $\mathbf{n}=(n_x,n_y,n_z)$ is 0, and all pass through the light source at $\mathbf{S}=(S_x,S_y,0)$, we can simplify the V-plane equation $\mathbf{p}\cdot\mathbf{n}+d = 0$ to

$$x + \alpha\cdot z = S_x \qquad \text{Eq. (1)}$$

with a single "gradient" parameter $\alpha = n_z/n_x$ describing the plane. If one 3D point on the light plane is known, it is simple to compute its α-value; vice versa, if the α-value for a V-line in the IR image is known, the depth z_k along all of its pixels P_k can be computed to

$$z_k = S_x / (v_{k,x}/v_{k,z} + \alpha) \qquad \text{Eq. (2)}$$

where $\mathbf{v_k}=(v_{k,x},v_{k,y},v_{k,z})$ is the direction of the ray defined by the IR camera's center of projection and the pixel P_k. The reason for the simplicity of the above equations is the selection of the IR camera as origin, and the alignment of the baseline and the stripes with the x- and y-axes, respectively.

We determine the α-value for each V-line in a given frame using the following methods:

1. Intra-frame tracking

α-values of V-lines that intersect with the H-line are directly computed from the 3D coordinates of the intersection point according to Eq. (1). Depth is assigned to every line pixel according to Eq. 2; we refer to this procedure as intra-frame tracking because the depth assignment is based on tracking a V-line within a given frame.

2. Inter-frame tracking

For remaining V-lines, we search the previous frame for similar lines, i.e. lines that are close enough as compared to the average line spacing, and that correspond to the same type of edge, i.e. either a transition from a dark to a bright or from a bright to a dark stripe. We compute a score for that V-line based on proximity and line length, and if the

score is above a credibility threshold, we choose the previous-frame V-line with the highest score as parent, and assign the α-value to its child, namely the corresponding V-line in the current frame. In this manner, the plane equation is passed on to future frames; we refer to this procedure as inter-frame tracking, because V-lines are tracked across consecutive frames. For offline processing, future frames are also available, and can be used to reverse the tracking direction in time in an anti-causal manner.

3. Line counting

Another way to determine α-values for V-lines that could not be determined using either of the previous approaches is to use their neighboring lines. However, due to horizontal depth discontinuities, non-consecutive V-lines may erroneously appear as consecutive neighbors in the frame. Therefore, we assign an α-value only if (a) both left and right neighbor lines are identified and suggest approximately the same α-value; in this case we choose the arithmetic mean between the two values; or (b), if a line has a similar length to its identified neighbor, and runs roughly parallel to it at the distance to be expected according to the line on the other side of the identified neighbor.
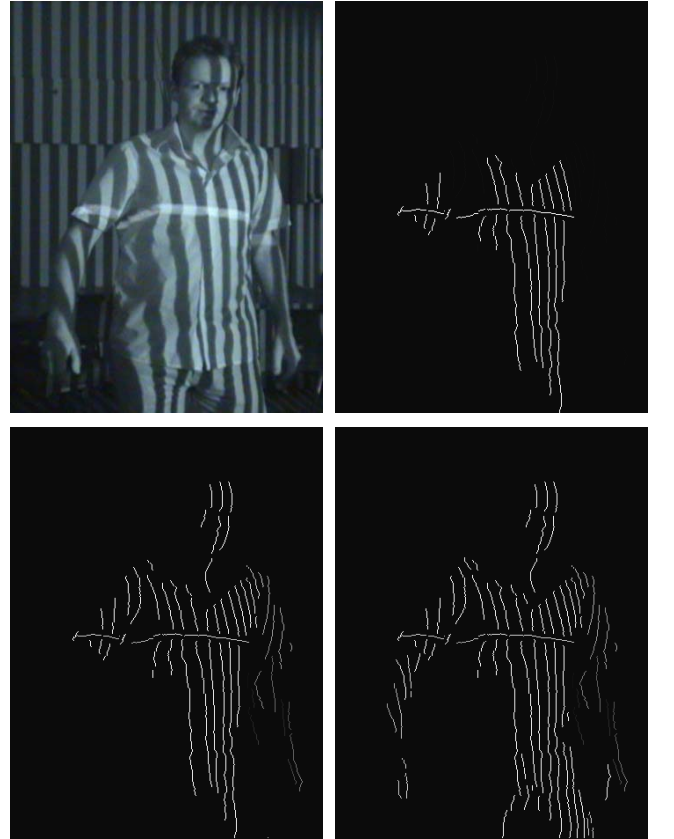

Figure 6: Reconstructing the depth along V-lines. (a) IR frame; (b) V-lines from intra-frame tracking only; (c) V-lines with additional forward inter-frame tracking, (d) final result after V-lines with both forward and backward inter-frame tracking, and line counting.

To determine α-values, we first apply intra-frame tracking, followed by bi-directional inter-frame tracking, followed by line counting, and finally inter-frame tracking again, in order to track the lines identified via line counting across frames. Using the obtained α-values, we compute depth values z for the identified lines according to Eq. 2. Figure 6 shows the resulting depth lines, with depth coded as gray value.

IV. DENSE DEPTH FRAME RECONSTRUCTION

Using the techniques described in the previous section, we obtain depth along sparse lines in the IR image. However, our final goal is to obtain dense depth for foreground objects for every frame of the video sequence. Hence, we need to interpolate between depth lines and extrapolate the depth within the entire moving foreground object. To this end, we apply the following steps to the sequence of VIS frames:

A. Project the IR depth lines onto the corresponding VIS frame

Since both the IR and the VIS camera are calibrated, this step is straightforward. From the depth value in the IR frame, a 3D vertex is computed using the internal and external camera parameters. This vertex is then projected into the VIS image, and depth is accordingly assigned to the corresponding pixel. The result is a sparse depth image which looks very similar to the one for the IR frame, but which is now superimposed on top of the VIS frame on a pixel by pixel basis.

B. Determine the foreground silhouette in the VIS frames

In order to assign depth values to the entire silhouette of a foreground object, it is necessary to precisely determine its boundaries in the VIS frame. In principle, we could apply the same steps that we used to compute the silhouette in the IR domain shown in Figure 5(b), i.e. background differencing followed by median filtering and small region removal. However, this procedure would inevitably result in a silhouette which also includes the object's shadow, as seen in Figures 7(a) and (b).

The silhouette however can be substantially improved, if IR frames are also taken into account. Because we place the IR and the visible light source at different locations within the scene, the IR shadow appears on the opposite side of the object as compared to the visible shadow, and can thus be identified as such. This is shown in Figures 7(b) and 7(c), which show the silhouette in the visible and IR domain, respectively. Specifically, we first determine correspondence between foreground pixels in IR and VIS images by projecting the reconstructed 3D points of the IR V-lines into the VIS image; this step is necessary due to parallax caused by the slight offset between the two camera locations. Using this correspondence, we modify foreground regions identified in the VIS images according to foreground regions detected in the IR images. Specifically, in order to eliminate shadows, we remove all pixels in the VIS image that are darker than the reference

background if the corresponding IR frame pixel is not marked as foreground. Additionally, we mark pixels in the VIS image for which the corresponding IR frame pixel is substantially brighter than the reference IR background.
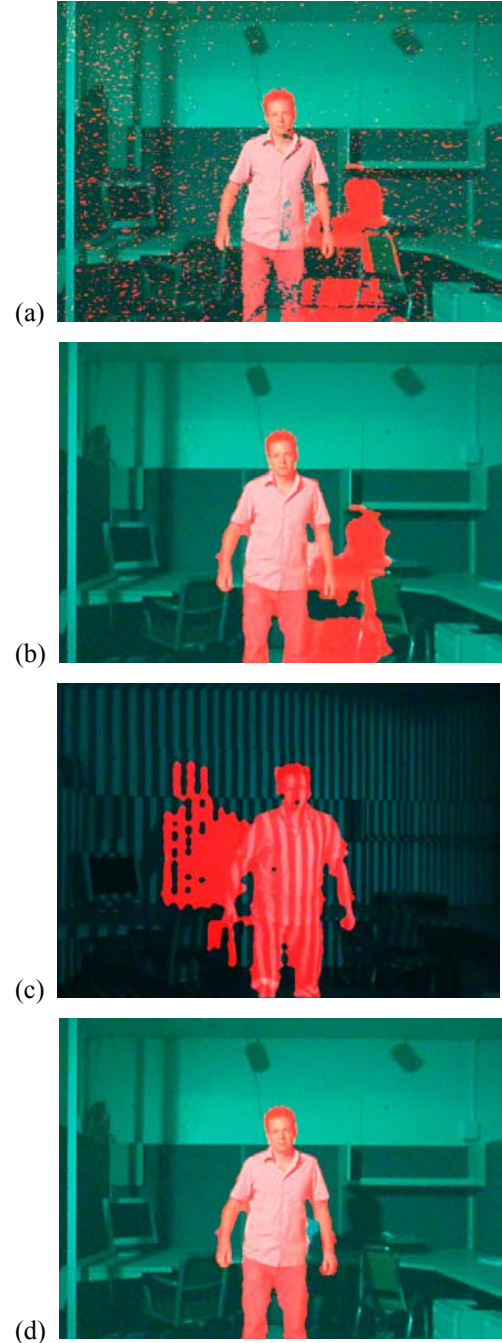

(a)

(b)

(c)

(d)

Figure 7: Determining the foreground silhouette in the VIS frames. (a) Foreground areas identified in the VIS image by using background differencing and thresholding; (b) part (a) followed by median filtering and removal of small regions, (c) corresponding silhouette in the IR image computed in the previous section, (d) VIS silhouette after combining IR and VIS background difference, median filtering and removal of small regions.

The later step is intended to fill areas where foreground object and reference background image have by coincidence the same color in the visible domain; we only use pixels that are brighter than the original background because they are certainly caused by the presence of a foreground object, while darker IR pixels can also result from shadows. Finally, median filtering and small region removal is again used to smooth the silhouette shape, resulting in a silhouette without shadow as seen in Figure 7(d).

C. Dense Depth Interpolation

All pixels in the obtained VIS silhouette are subsequently assigned a depth value either directly from the V-lines, or via interpolation and extrapolation. Specifically, to each pixel without depth, we assign a depth value computed from the depth of surrounding V-line pixels, weighed inversely proportional to their distance.

V. RESULTS

Using the acquisition system described in Section II, we have recorded a 30 second movie of a person moving around in an office environment. The sequence consists of 900 NTSC video frames from the IR-camcorder, and 300 VIS frames from the FireWire camera. Figure 8 shows examples of IR frames and their corresponding VIS frames; as seen, the IR stripe pattern does not affect the VIS frames at all.
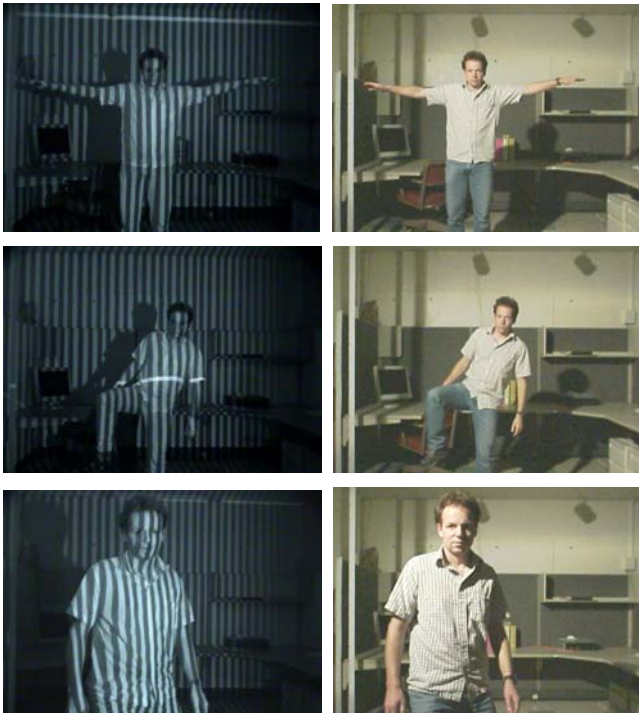

Figure 8: IR frames and corresponding VIS frames.

Applying the algorithms described in Section III, we reconstruct the depth along the detected V-lines in the IR frames using inter-frame and intra-frame tracking in both

forward and backward directions. We then create a dense depth map as described in Section IV. Figure 9 shows examples of the resulting sparse and dense depth frames. As seen in 9(b), we are able to recover the depth along most of the V-lines on the foreground object; the reconstructed 3D vertices along these V-lines are shown from different viewpoint in Figure 9(c). Figure 9(d) shows the VIS frames of the sequence, and Figure 9(e) the corresponding dense depth frames reconstructed from the VIS silhouette and the V-lines. There is a one-to-one correspondence between the pixels in 9(d) and depth pixels in Figures 9(e). Finally, Figure 9(f) shows renderings of the scene as seen from different viewpoints.

While Figure 9 demonstrates that our approach is capable of reconstructing a dynamic scene, it also illustrates some limitations of our current setup: As seen in the leftmost example in 9(b), objects that are small compared to the stripe pattern such as the hands of the subject cannot be properly detected and reconstructed. This is because (a) they produce short V-lines that do not necessarily intersect with the sweeping H-line in any of the frames, or (b) do not overlap enough with the V-lines of the previous frame to be tracked from frame to frame, if they move fast in the vertical direction. As the density of vertical lines decreases with the distance of the object, the minimum size of a detectable object increases with distance. In the rightmost example of Figure 9(b), there are only a few lines on the entire foreground object, and the arm could not be detected at all. The depth values at these locations can only be estimated by extrapolating from nearby depth values; while this is often a good estimate, it is obviously not always correct

There are other open issues inherent to our approach that need to be dealt with: First, the depth for "uncooperative" materials which appear dark in the IR domain, such as hair, cannot be recovered since IR lines are invisible on them. Similarly, clothing with strong patterns could potentially interfere with the projected IR pattern; however, in practice, we have found that a surprising number of textile colors, in particular green and dark colors, appear white in the IR domain; thus, patterns that are well visible in the VIS domain are often subtle or invisible in the IR domain. Second, other IR light sources in the scene could interfere with the projected pattern. In indoor scenes, this can be solved by placing IR-blocking filters in front of each light source; however, this is not feasible for outdoor scenes with sun light. One possible solution is to operate our IR projection system within a very narrow spectral bandwidth, e.g. by using laser as the light source for the vertical stripes, and a narrow-bandwidth interference IR filter for blocking all other IR wavelengths at the capture devices.

Some system parameters could be traded off against each other: In previous experiments, we tested our system with a wide horizontal baseline of 130 cm, rather than the current baseline of 60 cm between camera and stripe projector. While this results in a higher accuracy and less noise for the depth estimates, it can create problems if the object is too

close to the camera and/or moves too fast. Specifically, if the subject gets too close to the system, the difference between projection and viewing angle become too large, thus creating "projection shadows", i.e. large object regions without any lines. Additionally, V-lines appear and disappear within very few frames upon body turns, and move rapidly upon body motion, and can therefore not be tracked reliably. These problems do not occur in our current system with narrower baseline. Another system parameter that could be modified is the capture frame rate of the IR camera. If the off-the-shelf camcorder is replaced by a high-speed camera, the amount of V-line movement between consecutive frames becomes smaller. That would also allow us to increase the number of V-lines and the speed of the H-line sweep, though at the expense of increased cost. Depending on the a priori knowledge of the scene, depth estimation results could also be substantially improved by imposing additional constraints such as smoothness of surfaces and motions, or human body models.

VI. CONCLUSION AND FUTURE WORK

We have presented a minimally invasive system for capturing dynamic scenes without projecting visible light pattern and changing the visual appearance of the scene. While we have demonstrated the principle feasibility of our approach, there are several open issues to be addressed in future work: First, we believe the parameters of our initial system, e.g. horizontal and vertical baseline, capture rate, number and speed of sweeping horizontal lines, and number and density of vertical lines, need to be optimized. Second, a probabilistic framework could be implemented to assess and cope with uncertainty of the reconstructed depth estimates. Third, more sophisticated techniques for detecting the moving silhouette, could be used in order to handle situations where multiple objects or persons occlude each other, or the background is changing. Forth, processing speed can be substantially improved to allow real-time or near-real-time depth estimation. Finally, two or more of the described systems could be combined to capture a scene from multiple angles; this could pose additional problems due to possible interference of the projected patterns with each other.

VII. ACKNOWLEDGEMENTS

VIII. REFERENCES

[1]   J. Carranza, C. Theobalt, M. Magnor, H.-P. Seidel, "Free-Viewpoint Video of Human Actors," Proc. ACM Conference on Computer Graphics (SIGGRAPH'03), San Diego, USA, pp. 569-577, July 2003.

[2]   G. Cheung, S. Baker, and T. Kanade. Visual hull alignment and refinement across time: A 3D reconstruction algorithm combining shape-frame-silhouette with stereo. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'03), Madison, MI, June 2003..

[3]   B. Goldlücke, M. Magnor, "Space-Time Isosurface Evolution for Temporally Coherent 3D Reconstruction.", Proceedings of CVPR 2004, Washington, July 2004, p. 350-355.

[4]   http://www.3dvsystems.com

[5]   Itaru Kitahara, Yuichi Ohta, "Scalable 3D Representation for 3D Video Display in a Large-scale Space", Proc. of IEEE Virtual Reality 2003 Conference (VR2003), pp.45-52, (2003)

[6]   T. Kanade, H. Saito, and S. Vedula: "The 3D Room: Digitizing Time-Varying 3D Events by Synchronized Multiple Video Streams", Tech. report CMU-RI-TR-98-34, Robotics Institute, Carnegie Mellon University, December, 1998

[7]   T. Koninckx, I. Geys, T. Jaeggli, L. Van Gool: "A Graph Cut based Adaptive Structured Light Approach for Real-Time Range Acquisition", 3DPVT 2004, p. 413-421, Thessaloniki, 2004

[8]   T. Naemura, J. Tago, and H. Harashima, "Real-Time Video-Based Modeling and Rendering of 3D Scenes", IEEE Computer Graphics and Applications, Vol. 22, No. 2, pp. 66-73, Mar-Apr 2002

[9]   S. Rusinkiewicz, O. Hall-Holt, M. Levoy,   "Real-time 3D model acquisition ", Siggraph' 02, San Antonio, p. 438-446, 2002

[10]   S. Seitz and C. Dyer, "Photorealistic Scene Reconstruction by Voxel Coloring , International Journal of Computer Vision, 35 (2): 151-173, Nov. – Dec., 1999

[11]   S. Vedula, S. Baker, S. Seitz, and T. Kanade, "Shape and Motion Carving in 6D " Proc. Computer Vision and Pattern Recognition Conf. (CVPR) , 2000

[12]   J. Zhang, B. Curless, S. Seitz, "Rapid Shape Acquisition Using Color Structured Light and Multi-pass Dynamic Programming", 3DPVT 2002, Padua, pp.24-36, 2002

[13]   C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. A. J. Winder, and R. Szeliski, "High-quality Video View Interpolation Using a Layered Representation", ACM SIGGRAPH 2004 and ACM Trans. on Graphics, Aug. 2004
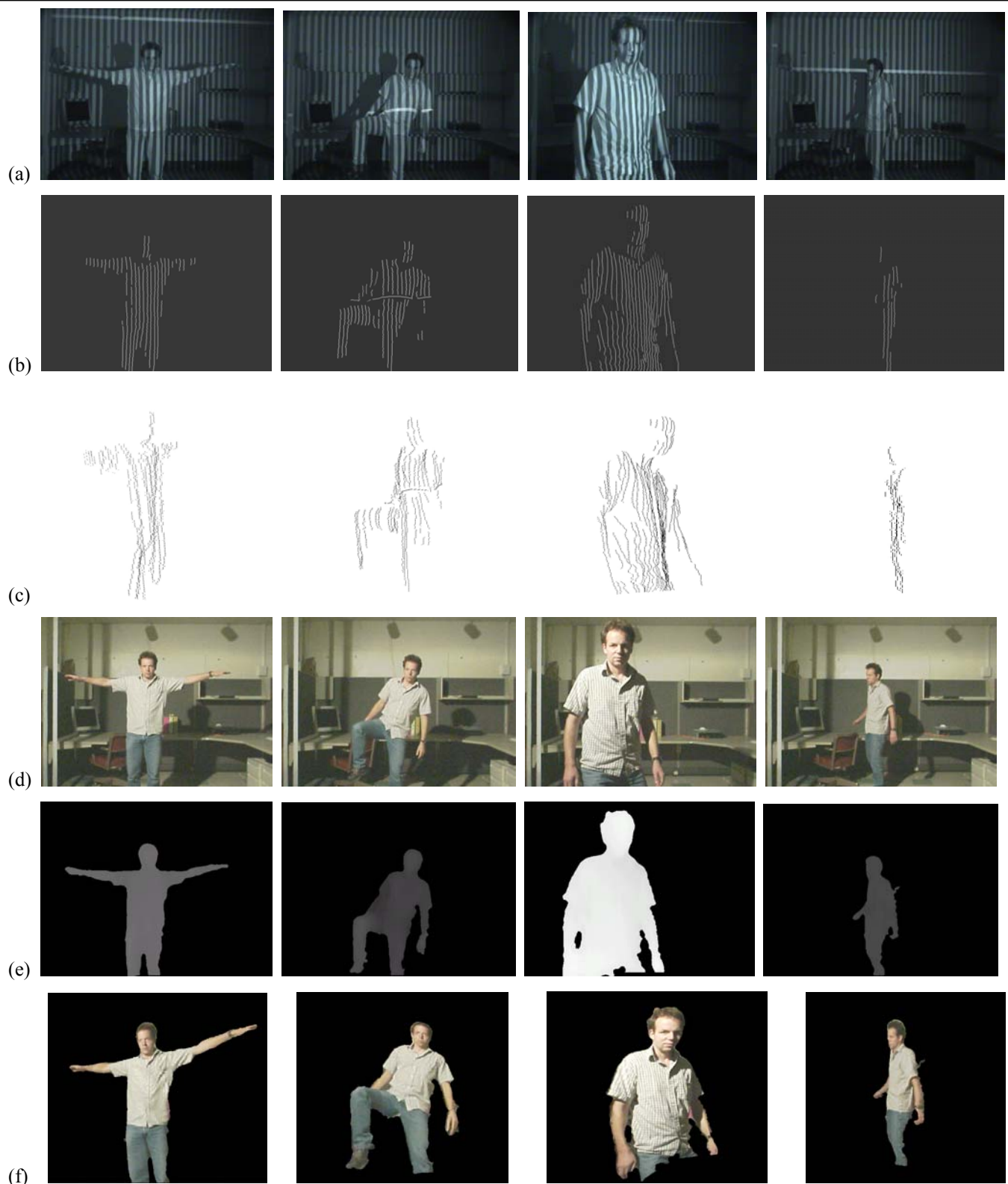
Figure 9: Sample frames of a recorded 30-second sequence. (a) IR video frames; (b) reconstructed sparse depth along H- and V-lines; (c) 3D points from V-lines rendered from a different viewpoint; (d) color video frames; (e) dense depth frames; (f) rendered views of a texture-mapped VRML model from a different viewpoint. The pixels of the VIS images in (d) and the depth frames in (e) correspond to each other one-to-one.