Viewpoint-Coded Structured Light

Mark Young
University of California, Santa Cruz
meyoung@ucsc.edu

Erik Beeson
University of California, Santa Cruz
ebeeson@cs.ucsc.edu

James Davis
University of California, Santa Cruz
davis@cs.ucsc.edu

Szymon Rusinkiewicz
Princeton University
smr@cs.princeton.edu

Ravi Ramamoorthi
Columbia University
ravir@cs.columbia.edu

Abstract

We introduce a theoretical framework and practical algorithms for replacing time-coded structured light patterns with viewpoint codes, in the form of additional camera locations. Current structured light methods typically use $\log(N)$ light patterns, encoded over time, to unambiguously reconstruct N unique depths. We demonstrate that each additional camera location may replace one frame in a temporal binary code. Our theoretical viewpoint coding analysis shows that, by using a high frequency stripe pattern and placing cameras in carefully selected locations, the epipolar projection in each camera can be made to mimic the binary encoding patterns normally projected over time. Results from our practical implementation demonstrate reliable depth reconstruction that makes neither temporal nor spatial continuity assumptions about the scene being captured.

1. Introduction

Range imaging has proven useful in a large number of application areas. Nearly all high quality range estimation methods use some sort of actively projected light to code space. These light codes are later observed by cameras and decoded to determine depth. If a particular system is to obtain a range resolution of $1/N^{th}$ of the working volume, then it will need N codes to uniquely code space. If the method uses binary codes then $\log(N)$ bits of information will need to be encoded by some method.

Some methods code these $\log(N)$ bits of information temporally using a succession of changing patterns so that each pixel receives a unique code over time. Unfortunately this places a temporal coherence constraint on the scene: it cannot move too fast or successive codes will not fall on the same scene elements. Other methods code the $\log(N)$ bits of information into a small spatial neighborhood, so that each point in the scene has a unique coded pattern projected onto it. Since the pattern is static, fast moving objects can be measured. Unfortunately, detection of the correct code requires an assumption of surface continuity in a small neighborhood around each pattern. If an object edge or depth discontinuity breaks the pattern, range cannot be sensed.

In summary, all existing methods are limited in either spatial or temporal resolution—this limitation is fundamental to the techniques used to encode and reconstruct depth. No methods yet proposed have provided a unique coding of space while requiring neither spatial nor temporal continuity constraints. Where would the bits of information go, if not into spatial or temporal neighborhoods?

This paper proposes that a unique coding of space can be induced by a single high frequency projected pattern and enough appropriately-positioned cameras. Although it is counterintuitive that a single projected pattern can induce different codes from different viewpoints, it can be shown using an analysis that replaces the concept of epipolar lines with a new construct called the epipolar segment. The epipolar segment is a restricted subset of the epipolar line that covers only the working volume, rather than all of space. Careful positioning of cameras ensures that the epipolar segment's projection onto each camera has a different code. The actual computation of depth is performed using a variant of existing multibaseline stereo methods that robustly accounts for occlusion.

This paper makes two specific contributions:

- It proposes a new theory for range imaging that allows the unique bits of depth information to be precisely coded using camera viewpoints rather than using time or space.
- It uses the theory to demonstrate a range imaging system design that guarantees correct results and that does not rely on spatial or temporal coherence.

2. Related Work

A large number of methods have been proposed to acquire 3D shape information, and many good surveys ex-

ist [2, 3, 7, 14, 23, 27, 31]. This section discusses several classes of techniques most closely related to our work.

Temporal coding: Many range estimation methods utilize time-varying patterns of light to robustly recover depth. Triangulation range scanners are perhaps the most commonly used commercial method in this category. For example, laser stripe scanners sweep a single plane of light over a scene [1, 26, 30]. The plane of light is in a unique location at each time instant, and can be used to recover depth. Variations include noting that a limited working volume allows two simultaneous light stripes to be disambiguated [21]. A number of other strategies for coding space with a time sequence of light patterns exist, including: binary codes [22], gray codes [6, 13], sinusoidal patterns [12], and intensity ratios [5]. Some of these triangulation methods have been implemented in hardware and run at real-time rates [15, 20]. There also exist temporal coding methods based on principles other than triangulation. For example, time-of-flight imagers sense the phase of a returning light pulse using a computation based on several measurements over time [11, 16, 18]. Although some have been implemented at real time rates, all of the methods in this category fundamentally require temporal continuity in the scene being imaged.

Spatial coding: Some structured light techniques code space using a pattern that varies spatially, rather than temporally. Stripe patterns [4], grid patterns [24], and moiré fringes [32] are examples of methods that project high-frequency information and use a phase unrolling or line counting step to track depth change across a surface. Unfortunately, phase unrolling requires integrability and leads to ambiguities near depth discontinuities. These ambiguities may be addressed by projecting lower-frequency patterns that can be uniquely identified [10, 19, 33]. However, this requires the surface to be continuous on the scale of the pattern. These methods all make assumptions about the spatial continuity of objects, either locally or globally.

Spatio-temporal coding: Researchers have attempted to merge the advantages of temporal and spatial coding. These methods improve temporal coding by reducing the number of patterns required and improve the accuracy of spatial coding using the temporal information [9, 25, 34, 35]. These systems produce high-quality depth reconstructions suitable for both animation [36] and high resolution reconstruction of static objects [8]. All of these methods require both spatial and temporal coherence.

Multiview reconstruction: Viewpoint coding relies on placing cameras at multiple locations to reduce ambiguity. This methodology is usually called multi-baseline stereo by the machine vision community. A good survey of modern methods is due to Seitz et al. [28]. With many cameras, the probability that correspondences arise by chance,

hence result in incorrect 3D estimates, can be greatly reduced. Our method differs from prior work in that it is designed to turn this probabilistic argument into a deterministic one: we select camera positions together with an easily-distinguishable projected light pattern (e.g., alternating black and white stripes) such that each subsequent camera systematically constrains permissible 3D locations.

Summary and differences from previous work: Viewpoint coding differs from prior work in two important ways. As compared to structured light methods, it does not rely on spatial or temporal coherence, making it suitable for a wider class of scenes, including those with moving objects. As compared to multi-baseline stereo, it allows for *guarantees* about the resulting surface reconstruction.

3. Viewpoint Coding

Our viewpoint coding approach argues that *multiple camera viewpoints* of a scene may substitute for temporal or spatial (active light) codes. To do so, we develop a camera configuration in which possible depths of points in the scene deterministically map to different disparities. With k camera positions, we are thus able to distinguish between 2^k depths, in a manner reminiscent of structured-light methods.

3.1. Framework for Viewpoints as Codes

Epipolar segment: One of the key insights, that allows camera viewpoints to act as codes, is that the epipolar line traditionally used to constrain stereo matching is far too general. The working volume constrains the range of possible depths within the epipolar line—this constraint is well known and has been widely used to enhance the efficiency of stereo matching. In our work, we focus only on the segment of the epipolar line corresponding to the working volume, which we call the *epipolar segment*. Unfortunately, it has remained common to visualize the entire epipolar line in diagrams, making reasoning about viewpoint coding difficult. Visualizing only the epipolar segment clarifies the actual constraints substantially.

Figure 1, top, shows a visualization of a multiview camera configuration. The epipolar segment of a single projector pixel is shown in red: it is the intersection between a ray from the projector P and the working volume. Figure 1, middle, shows the traditional visualization of the scene from camera C1. Note that the epipolar line extends across the entire image and there are several possible matches. Also shown is the same view from camera C1, together with only the epipolar segment. It is clear that the matching ambiguity has been substantially reduced. Figure 1, bottom, shows the views from each of the remaining cameras. Notice that the epipolar segment extends across a different number of stripes in each viewpoint.
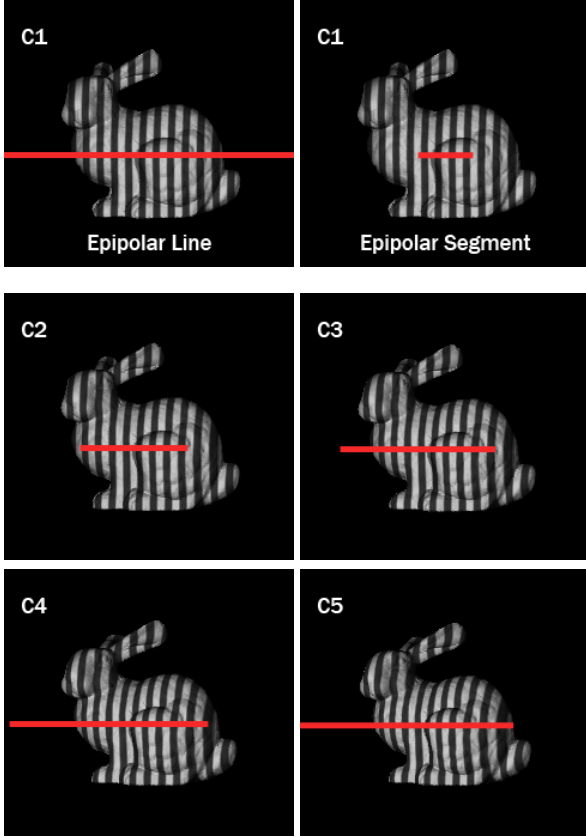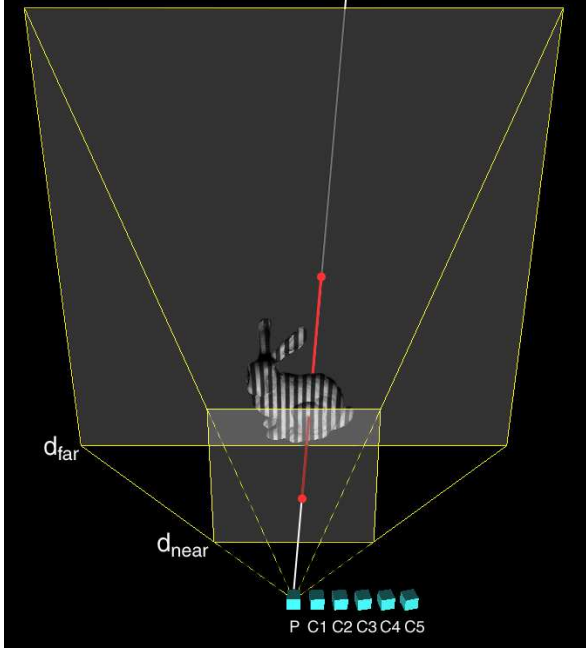
Figure 1: Top: *A 3D visualization of the* epipolar segment, *the portion of the epipolar line lying within the working volume.* **Middle:** *The scene as viewed from C1, with a comparison of visualizing the entire epipolar line and just the epipolar segment.* **Bottom:** *Views from the remaining cameras: note that the projected extent of the epipolar segment is variable.*

Viewpoints as codes: The changing projection size of the epipolar segment allows different cameras to represent different codes. Figure 2 shows the pixel intensities along the epipolar segment in each camera view stacked together into a single image. Each column is one possible surface depth, and each row can be thought of as the code provided by a particular camera. The codes due to cameras with smaller epipolar segments (in this case, C1 and C2 for example), are mostly low-frequency, since the small number of stripes in that segment covers the full depth volume. The codes for further cameras (C3, C4 and C5) are higher-frequency, since there are many more stripes within the (longer) epipolar segment. This is similar to the different spatial frequencies typically used for temporal binary codes in structured light.

Note that the codes due to this simple camera arrangement do not provide sufficient uniqueness to unambiguously recover depth, and we see that three different depths match the pixel color of 'white' in all camera views. In this paper, we show that by carefully choosing camera positions, the codes can be adjusted so that they uniquely code depth, in a manner analogous to binary time coded stripe patterns (see Fig. 3, which now has no ambiguities).
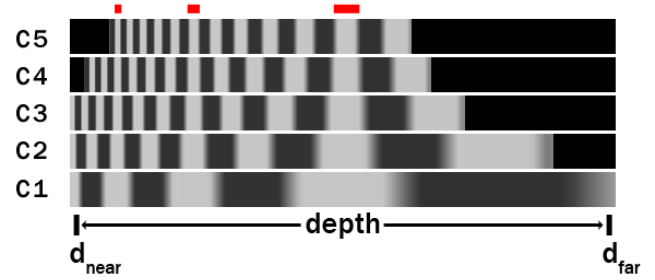

Figure 2: *The pixel intensities along the epipolar segment as observed in each camera view are stacked together. Each row can be thought of as a particular code. Note that in this case, the correct depth is ambiguous because the codes do not uniquely partition space.*
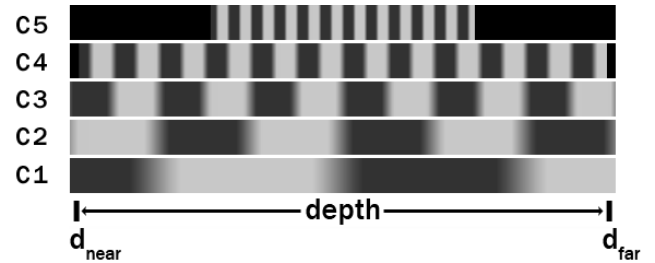

Figure 3: *Changing the location of the cameras allows the observed codes to be changed. By choosing the correct camera locations, a unique coding resembling binary time codes is possible.*
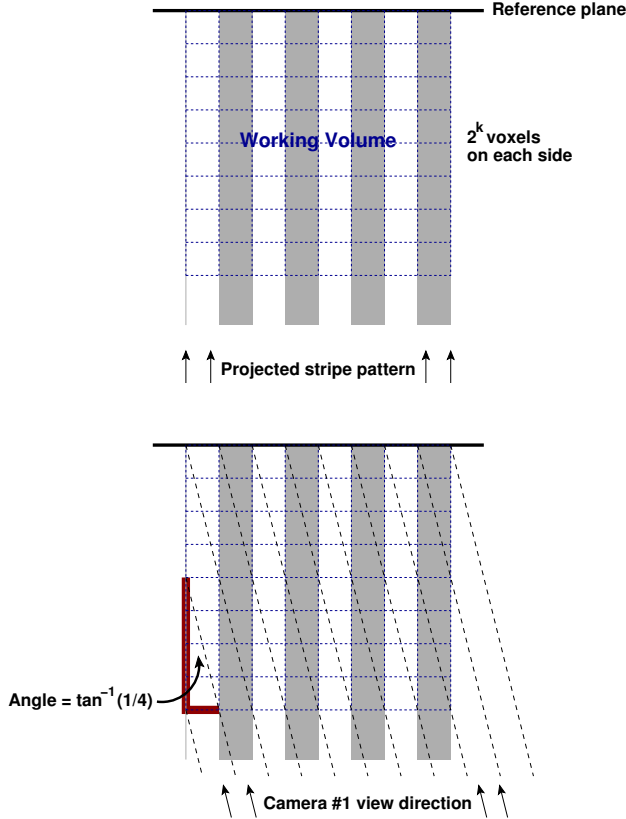
Figure 4: Top: *Simplified (parallel projection) configuration for deriving our viewpoint coding.* **Bottom:** *The first camera of our viewpoint coding scheme is oriented at an angle of* $\tan^{-1}(1/2^{k-1})$. *Additional cameras have angles of* $\tan^{-1}(1/2)^{k-i}$, *for* $i = 2\ldots k$.

3.2. Deriving Camera Placement

Orthographic cameras: In order to derive the required camera positions, we begin with a simplified arrangement with cameras that use parallel projection and a scene that is completely visible from all cameras. We will see later how our scheme may be extended to relax these assumptions, accommodating perspective projection and occlusion. We assume a cubical working volume of $N = 2^k$ voxels on each side, and consider a single epipolar slice, as shown in Figure 4, top. A light pattern of alternating black and white stripes of width 1 is projected onto the scene.

A camera with the same view direction as the projected stripes will, of course, see exactly the projected pattern. However, a rotated camera will see a distorted view of the stripe pattern, with the amount of deviation, or disparity, proportional to the object's height above the reference plane. Consider one such camera, with resolution equal to that of the stripe projector but its view rays rotated by an angle of $\tan^{-1}(1/4)$, as shown in Figure 4, bottom.

With this orientation, the disparity of any point within the working volume is constrained to be between 0 (for an object at the reference plane) and 2 (for an object at the

front of the working volume). Since the spacing of identical (hence ambiguous) stripes is 2, there is never an ambiguity in matching the pattern visible from camera 1 to the known projected pattern. By observing the disparity of a given projected stripe, we therefore constrain the possible depth within that column to half the working volume.

In order to further refine the depth within each column, we add more cameras that allow for greater disparities. In our example with 8 stripes, we need a second camera at an angle of $\tan^{-1}(1/2)$ and a third with an angle of $\tan^{-1}(1/1)$. These result in maximum disparities of 4 and 8, respectively, hence by themselves would allow for ambiguous correspondences. However, the restrictions introduced by camera 1 on which part of the working volume may be occupied make the match from camera 2 unambiguous, and the restrictions of cameras 1 and 2 constrain the match from the third camera. Each additional camera, therefore, adds one additional bit to the accuracy with which depths are recovered, while introducing no additional ambiguity.

Perspective cameras: The above construction is valid under any transformation of 3D space that maps straight lines to straight lines (which, in turn, re-maps the rays associated with each camera and projector). The set of mappings with this property are perspective transformations: mappings expressible through multiplication by 4×4 matrices in homogeneous coordinates. Applying a projective mapping will therefore turn the parallel (orthographic) projections into perspective projections, making the configuration realizable with standard cameras.

Consider the i-th orthographic camera, as derived above. Its j-th ray intersects the xy plane at some location $(x_j, y_j, 0)$, but all the rays have the same direction $(-1/2^{k-i}, 0, -1)$. Here we assume that the x direction points towards the right of the page, z points towards the bottom of the page, and y points out of the page. In homogeneous coordinates, therefore, the camera rays may be written as

$$\begin{pmatrix} x_j \\ y_j \\ 0 \\ 1 \end{pmatrix} + t \begin{pmatrix} -1/2^{k-i} \\ 0 \\ -1 \\ 0 \end{pmatrix}. \tag{1}$$

We now search for a perspective mapping that will map each bundle of parallel rays into a set of rays that intersect at a single point. Equivalently, we are taking rays that intersect at a "point at infinity," and moving the intersection point to be finite. As we will show, the following matrix M accomplishes the desired transformation:

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{2}$$

To see this, we multiply by the ray equations to obtain:

$$M \left[\begin{pmatrix} x_j \\ y_j \\ 0 \\ 1 \end{pmatrix} - t \begin{pmatrix} 1/2^{k-i} \\ 0 \\ 1 \\ 0 \end{pmatrix} \right] = \begin{pmatrix} x_j \\ y_j \\ 1 \\ 0 \end{pmatrix} - t \begin{pmatrix} 1/2^{k-i} \\ 0 \\ 0 \\ 1 \end{pmatrix} \quad (3)$$

$$= \begin{pmatrix} x_j - t \cdot 1/2^{k-i} \\ y_j \\ 1 \\ -t \end{pmatrix}. \quad (4)$$

Finally, we perform the homogeneous divide, yielding

$$\begin{pmatrix} -x_j/t + 1/2^{k-i} \\ -y_j/t \\ -1/t \end{pmatrix} = \begin{pmatrix} 1/2^{k-i} \\ 0 \\ 0 \end{pmatrix} - \frac{1}{t} \begin{pmatrix} x_j \\ y_j \\ 1 \end{pmatrix}. \quad (5)$$

So, we see that the rays for camera i, which used to have a common direction but different origins, have been transformed to have a common origin but different directions. Considering the set of all cameras, we see that their centers of projection are all located on the x axis, spaced in a geometric series. The projector, which corresponded to a ray direction of $(0, 0, -1)$ in the orthographic case, is similarly remapped to a standard perspective projector located at the origin. This configuration is shown in Figure 5. Note that the cameras may be rotated arbitrarily to ensure that the working volume is visible.
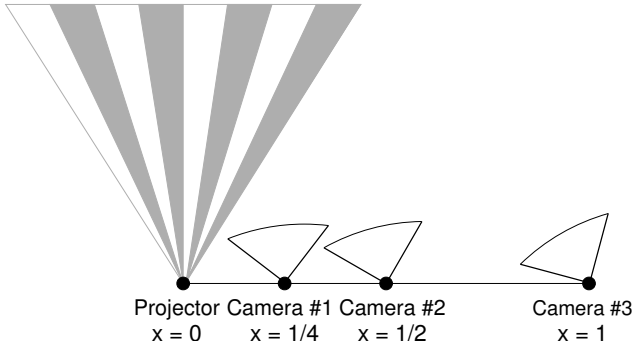

Figure 5: *Schematic of our viewpoint coding configuration with perspective cameras.*

3.3. Occlusion

For general scenes, we must account for the possibility of occlusion. In this case, we replace the simple correlation-based multiview correspondence algorithm with a voxel-carving [17, 29] approach that considers the scene from front to back. For each voxel, we find its projections into all the cameras and evaluate whether the data are consistent with its being occupied. Note that unlike with voxel coloring, in which consistency is evaluated by similarity of colors from all the camera positions, the consistency check in our case simply considers whether the projections are all "white" or all "black." Because voxels on the front-most layers are not occluded, their presence or absence can

Figure 6: *Screen capture from our simulator. For the selected point, the simulator displays epipolar line segments for all camera views (right), and shows the colors seen by each camera for all hypothesized depths along the ray (bottom). This capture shows an equidistant camera configuration, resulting in an ambiguous match for this ray (acceptable matches are marked in red).*

be unambiguously determined. For deeper layers of voxels, we evaluate whether, given the scene reconstruction so far, the voxel might have been occluded from any camera view. If so, we mark the voxel as "shadowed" and do not attempt to either carve it or mark it as occupied. In this way, we are guaranteed to obtain a conservative reconstruction: any voxels we mark as occupied are guaranteed to contain geometry. This is in contrast to standard multi-view voxel carving methods, which are only probabilistically correct (i.e., false correspondences may result in spurious reconstructed geometry). The effect of performing the space carving is shown in the following section.

4. Results

Simulator: We have implemented a software simulator for the proposed viewpoint coding configuration, and used it to verify that it is possible to obtain high-quality reconstructions. A screen shot is shown in Figure 6.

We used our simulator to investigate the effects of camera placement on ambiguity. Figure 7 shows the results. In this experiment, we were comparing the proposed configuration to one in which the same number of cameras was used, but they were equally spaced (this has traditionally been the most typical camera placement for multibaseline stereo). In the visualization, green pixels denote unambiguous matches while magenta pixels indicate ambiguity. The equidistant configuration results in many ambiguous matches, while the proposed method produces almost no
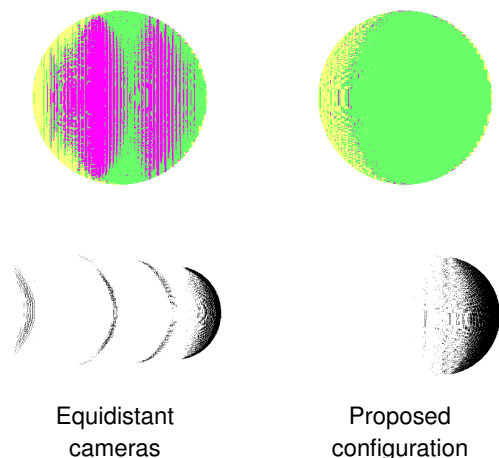
Equidistant
cameras

Proposed
configuration

Figure 7: *A comparison of equidistant camera placement and our proposed method.* ***Top:*** *Green pixels indicate unique matches, while magenta denotes ambiguity. Yellow pixels indicate no match is found (primarily because not all cameras see the point).* ***Bottom:*** *The ambiguous depths in the equidistant configuration effectively lead to multiple copies of the geometry at different depths. By contrast, our proposed configuration produces an accurate reconstruction. Note that all reconstructions are obtained by local matches for each pixel separately, and do not use spatial windows or global continuity.*

ambiguity. The ambiguity in the equidistant configuration is manifested as effectively multiple copies of the geometry at distinct depths, as opposed to the accurate reconstruction with our proposed configuration.

A simulator also allows us to analyze the behavior of our occlusion handling method, since the scene geometry is known. Figure 8 shows a simple two-plane scene with a discontinuity. The visualization shows correct matches in green, incorrect matches in red, and occluded regions in blue. Without occlusion handling some camera codes are drawn from the front surface and some codes from the back surface, resulting in incorrect depths. Occlusion handling correctly detects the shadow regions (blue) near discontinuities so that they can be excluded from processing.

Real-world data: In addition to the software simulator, we have experimented with real data, captured by moving a digital camera to many positions using a translational stage (Figure 9).

Figure 10 shows an example scene captured and then reconstructed using this device. We captured images both using our proposed method and using equidistant cameras. In each case we show the viewpoint coding. Note that just as with the synthetic data, equidistant cameras result in ambiguity; in contrast, our method yields unique codes.

Figure 11 shows a sample mesh reconstructed using viewpoint coding, as a re-lit rendering. Note the relatively low noise and high detail present in the resulting 3D model.
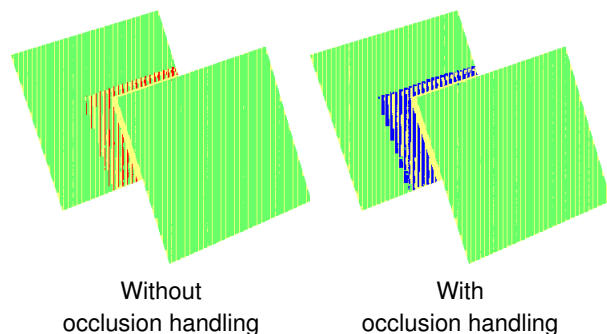

Without
occlusion handling

With
occlusion handling

Figure 8: *Occlusion handling is important for correct reconstruction. Green pixels indicate correct matches, red pixels incorrect matches, and blue pixels regions detected to be in shadow. The yellow pixels represent points for which a depth value was not found, and are mostly (as expected) at stripe boundaries. Note that there are many incorrect depths (red points) located in the shadow region when occlusion handling is not used. With occlusion handling the relevant regions are detected as occluded (blue) and excluded from depth recovery.*

5. Conclusion and Future Work

Viewpoint coding adds a new class of methods to the known theories of structured light. Previously, unambiguous reconstruction codes could only be encoded spatially or temporally. Viewpoint coding, in contrast, allows for depth reconstruction without making any spatial or temporal continuity assumptions about the scene. Our construction provides for decoding of N unique depths with $\log(N)$ camera positions.

One consequence of our analysis is that multiple cameras will allow for "one shot" depth reconstruction. To explore this, we are investigating the construction of a device with 10 synchronized video cameras and a static pattern projector. Based on the analysis in this paper, with data captured using a linear translation stage, we anticipate that such a rig will permit reconstruction of complex moving objects. We believe that the lack of temporal and spatial continuity assumptions will yield high quality for objects that are either too fast-moving or too complex to be reconstructed by current methods that use temporal or spatial coding.

Our examples so far have used exclusively viewpoint coding. However, it is possible to combine viewpoint coding with other coding strategies. For example, using stripes of c different colors it is possible to encode N unique depths with $\log_c N$ camera positions. Other combined strategies, including those combining viewpoint and temporal coding, are future work.

6. Acknowledgements

Figure 9: *Capture rig used for viewpoint coding experiments. A digital camera is moved to the different required positions using a translational stage, while structured illumination is provided using a projector.*

References

[1] G. Agin and T. Binford. Computer description of curved objects. *IEEE Transactions on Computers*, C-25(4):439–449, 1976.

[2] J. Batlle, E. Mouaddib, and J. Salvi. Recent progress in coded structured light as a technique to solve the correspondence problem: A survey. *Pattern Recognition*, 31(7):963–982, 1998.

[3] P. Besl. *Active Optical Range Imaging Sensors, in Advances in Machine Vision, chapter 1*, pages 1–63. 1989.

[4] K. L. Boyer and A. C. Kak. Color-encoded structured light for rapid active ranging. *Trans. PAMI*, 9(1), 1987.

[5] B. Carrihill and R. A. Hummel. Experiments with the intensity ratio depth sensor. *Computer Vision, Graphics, and Image Processing*, 32(3):337–358, 1985.

[6] D. Caspi, N. Kiryati, and J. Shamir. Range imaging with adaptive color structured light. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(5):470–480, 1998.

[7] F. Chen, G. M. Brown, and M. Song. Overview of three-dimensional shape measurement using optical methods. *Optical Engineering*, 39(1):10–22, 2000.

[8] B. Curless and M. Levoy. Better optical triangulation through spacetime analysis. In *ICCV*, pages 987–994, 1995.

[9] J. Davis, R. Ramamoorthi, and S. Rusinkiewicz. Spacetime stereo: A unifying framework for depth from triangulation. In *CVPR*, pages 359–366, 2003.

[10] P. M. Griffin, L. S. Narasimhan, and S. R. Yee. Generation of uniquely encoded light patterns for range data acquisition. *Pattern Recognition*, 25(6):609–616, 1992.

[11] R. Gvili, A. Kaplan, E. Ofek, and G. Yahav. Depth keying. In *SPIE Electronic Imaging*, 2003.

[12] P. S. Huang, C. Zhang, and F.-P. Chiang. High-speed 3-D shape measurement based on digital fringe projection. *Optical Engineering*, 42(1):163–168, 2003.

[13] S. Inokuchi, K. Sato, and F. Matsuda. Range-imaging for 3D object recognition. In *ICPR*, pages 806–808, 1984.

[14] R. Jarvis. A perspective on range finding techniques for computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2):122–139, 1983.

[15] T. Kanade, A. Gruss, and L. Carley. A very fast VLSI rangefinder. In *IEEE International Conference on Robotics and Automation*, pages 1322–1329, 1991.

[16] M. Kawakita, K. Iizuka, T. Aida, H. Kikuchi, H. Fujikake, J. Yonai, and K. Takizawa. Axi-vision camera (real-time distance-mapping camera). *Applied Optics*, 39(22):3931–3939, Aug 2000.

[17] K. Kutulakos and S. Seitz. A theory of shape by space carving. In *Proc. ICCV*, 1999.

[18] R. Lange and P. Seitz. Solid-state time-of-flight range camera. *IEEE Journal of Quantum Electronics*, 37(3), Mar 2001.

[19] R. A. Morano, C. Ozturk, R. Conn, S. Dubin, S. Zietz, and J. Nissanov. Structured light using pseudorandom codes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(3):322–327, 1998.

[20] Y. Oike, M. Ikeda, and K. Asada. Design of real-time VGA 3-D image sensor using mixed-signal techniques. In *Asia South Pacific Design Automation Conference*, pages 523–524, 2004.

[21] J. Park, G. N. DeSouza, and A. C. Kak. Dual-beam structured-light scanning for 3-D object modeling. In *3D Digital Imaging and Modeling*, pages 65–72. IEEE Computer Society, 2001.

[22] J. L. Posdamer and M. D. Altschuler. Surface measurement by space-encoded projected beam systems. *Computer Graphics and Image Processing*, 18(1):1–17, 1982.

[23] D. Poussart and D. Laurendeau. *3-D Sensing for Industrial Computer Vision, in Advances in Machine Vision, chapter 3*, pages 122–159. 1989.

[24] M. Proesmans, L. V. Gool, and A. Oosterlinck. One-shot active 3D shape acquisition. In *Proc. International Conference on Pattern Recognition (ICPR) Vol. 7276*, 1996.
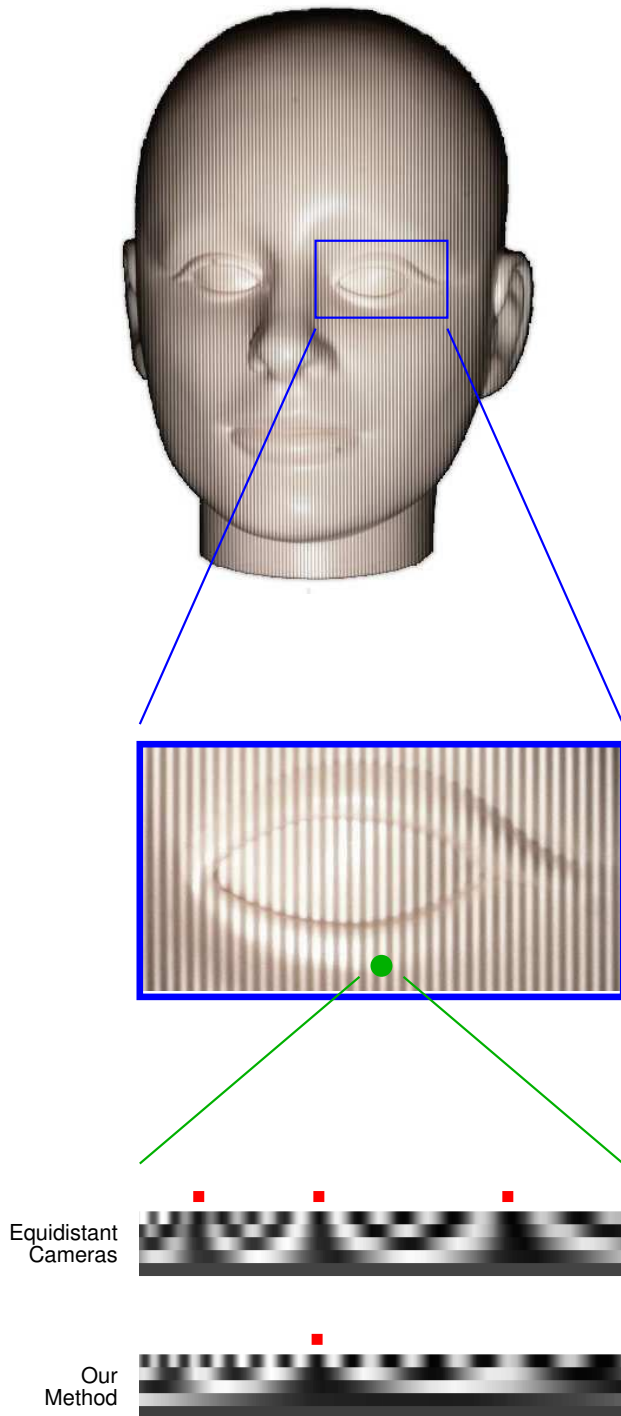
Figure 11: *Lit renderings of an example mesh reconstructed using our viewpoint coding. The mesh has relatively low noise yet preserves detail (e.g. around the lips and eyes).*

[25] S. Rusinkiewicz, O. Hall-Holt, and M. Levoy. Real-time 3D model acquisition. *ACM Transactions on Graphics (SIG-GRAPH 02)*, 21(3), 2002.

[26] P. Saint-Marc, J. Jezouin, and G. Medioni. A versatile PC-based range finding system. *IEEE Transactions on Robotics and Automation*, 7(2):250–256, 1991.

[27] J. Salvi, J. Pagès, and J. Batlle. Pattern codification strategies in structured light systems. *Pattern Recognition*, 37(4):827–849, 2004.

[28] S. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Proc. CVPR*, 2006.

[29] S. Seitz and C. Dyer. Photorealistic scene reconstruction by voxel coloring. In *Proc. CVPR*, 1997.

[30] Y. Shirai. Recognition of polyhedrons with a range finder. *Pattern Recognition*, 4(3):243–244, 1972.

[31] T. C. Strand. Optical three-dimensional sensing for machine vision. *Optical Engineering*, 24(1):33–40, 1985.

[32] H. Takasaki. Moiré topography. *Applied Optics*, 9(6):1467–1472, 1970.

[33] P. Vuylsteke and A. Oosterlinck. Range image acquisition with a single binary-encoded light pattern. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(2):148–164, 1990.

[34] L. Zhang, B. Curless, and S. Seitz. Rapid shape acquisition using color structured light and multi-pass dynamic programming. In *IEEE 3D Data Processing Visualization and Transmission*, 2002.

[35] L. Zhang, B. Curless, and S. Seitz. Spacetime stereo: Shape recovery for dynamic scenes. In *CVPR*, 2003.

[36] L. Zhang, N. Snavely, B. Curless, and S. Seitz. Spacetime faces: High resolution capture for modeling and animation. *ACM Trans. Graphics (Proc. ACM SIGGRAPH)*, 23(3):548–558, 2004.

Figure 10: *Result of reconstruction using real-world data. At top is an image of the target object (mannequin head), together with a closeup showing the projected pattern. Below, we show the viewpoint codes for a single point using both equidistant and viewpoint coded cameras. Note that viewpoint coded cameras result in a uniq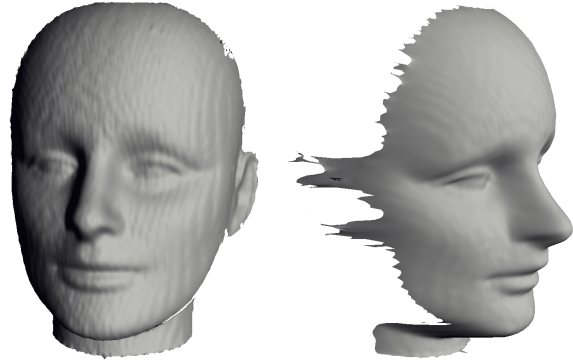ue match at each pixel, allowing correct reconstruction.*