Rapid Shape Acquisition Using Color Structured Light and Multi-pass Dynamic Programming

Li Zhang, Brian Curless, and Steven M. Seitz
Department of Computer Science and Engineering, University of Washington
Seattle WA 98195, USA

E-mail: {lizhang,curless,seitz}@cs.washington.edu

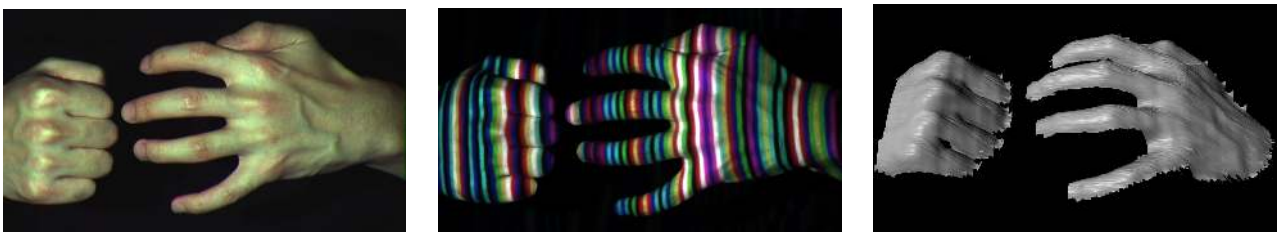


Figure 1. In this paper, we show how to reconstruct the shape of a scene, such as the two hands shown on the left, given a single photograph of the scene under color-striped illumination shown at center. A novel dynamic programming method leads to the geometric reconstruction on the right, shown as a shaded rendering from a new viewpoint.

Abstract

This paper presents a color structured light technique for recovering object shape from one or more images. The technique works by projecting a pattern of stripes of alternating colors and matching the projected color transitions with observed edges in the image. The correspondence problem is solved using a novel, multi-pass dynamic programming algorithm that eliminates global smoothness assumptions and strict ordering constraints present in previous formulations. The resulting approach is suitable for generating both high-speed scans of moving objects when projecting a single stripe pattern and high-resolution scans of static scenes using a short sequence of time-shifted stripe patterns. In the latter case, spacetime analysis is used at each sensor pixel to obtain inter-frame depth localization. Results are demonstrated for a variety of complex scenes.

1 Introduction

Reconstructing accurate shape from images is a long-standing and challenging problem in computer vision. Structured light techniques methods simplify the problem with the help of controlled illumination and can yield excellent results in practice. Recently, researchers have focused on speeding up the acquisition process by designing techniques that require only a small number of input images, in some cases even a single image. Indeed, being able to capture accurate shape from a single image opens up the possibility of scanning *moving* scenes by repeating the process at video rates.

Custom hardware solutions have been developed to solve the problem of rapid shape capture, but enabling the construction of rangefinders from more commonly available components makes them more accessible to other researchers. To this end, we focus on optical triangulation methods which can be developed with a video projector (in some cases, a slide projector) and a camera.

In designing optical triangulation systems, researchers seeking to minimize the number of images required for shape capture typically face a set of trade-offs. For instance, a gray ramp and a solid white pattern projected onto a surface can be used to encode position; after taking the ratio of the two images, the brightness at each pixel determines the corresponding point on the ramp. The drawback of such an approach is sensitivity to noise, as errors in brightness measurement can translate into substantial triangulation errors. Rather than a smooth pattern, we could instead project a high frequency pattern such as a square wave. While the imaged edges can be quite precisely localized and triangulated, an ambiguity problem arises: the correspondence between observed edges and projected image is not directly measurable.

In this paper, we seek to develop an accurate triangulation system, and thus follow the approach of projecting high frequency patterns. To simplify the correspondence problem, we project *color* patterns, which essentially encode more bits of information at each edge, at the expense of some limitations on surface reflectance (e.g., the surface must reflect

light in all channels). Nonetheless, due to the finite number of distinct edge transitions available in three color channels, ambiguity remains. Indeed, the edges themselves may be noisy and have non-zero likelihood of being associated with more than one different color transition. Our goal then is to find a surface that is the most likely among all possible hypothesized correspondences.

We address this multiple hypothesis correspondence problem with dynamic programming. Dynamic programming has long been used in stereo vision, but has a number of limitations that have made it less desirable than other methods of stereo reconstruction. In the context of color structured light rangefinding, however, we show that it can be quite powerful. We describe a dynamic programming method that constructs piecewise-continuous surfaces. One limitation of dynamic programming in this setting is the requirement that the surface be monotonic with respect to the projector and camera. To overcome this limitation, we develop a multi-pass version of dynamic programming that recovers surfaces that violate monotonicity. A second problem with dynamic programming as applied to traditional stereo correspondence is that incorporating inter-scanline constraints is problematic, resulting in abrupt disparity discontinuities between adjacent scanlines. With the aid of color structured light, however, we have observed that the correspondence is sufficiently robust that inter-scanline constraints are *simply not necessary*. This conclusion has been borne out in many experiments over a wide range of scanned objects.

Based on this dynamic programming technique, we demonstrate a system capable of reconstructing accurate shape from a single image. When more than one image can be obtained, notably for the case of a static scene, we show that the additional images can be incorporated to yield denser and more accurate reconstructions. In particular, using a small number of images acquired while the projector pattern shifts across the object, we can match against the time evolution of the reflected pattern observed at each pixel. This temporal matching is shown to have greater immunity to shape and shading variations.

The rest of the paper is structured as follows. Section 2 overviews some of the structured light scanning literature and proposes the architecture of our scanner. In Section 3 we formulate the edge correspondence problem as a multi-hypothesis code matching problem and present a multi-pass dynamic programming solution. Next, in Section 4, we describe the design of a color-coded projection pattern and its use in reconstructing shape from a single image. For static scenes, we then develop a spacetime method that can be used to attain high resolution results (Section 5). In Section 6, we describe our implementation and show results for both the single image and multiple image approaches. Finally, in Section 7, we summarize the work and suggest avenues for future research.

2 Related work

Optical triangulation has been an active area of research for decades. The techniques that have been developed range from those that require many images to reconstruct a surface to those that require only a single image. Here we discuss several, but by no means all, papers along that continuum.

Among the scanners that acquire many images, the swept stripe scanner is among the most common (e.g., [13, 21]): a plane of light sweeps across a surface while a CCD array images the stripe reflection and triangulates to the light plane, scanline by scanline. Rioux *et al.* [27] employ a flying spot and linear sensor array. Kanade *et al.* [23] sweep a light plane but record the time at which a peak is observed at each sensor pixel. This time is then used to triangulate the sensor line of sight back to the position of the stripe at that time. Curless and Levoy [12] generalize this temporal analysis (calling it "spacetime analysis") to other scanner configurations and observe that it substantially increases immunity to shape and reflectance variations which affect purely spatial analyses. While the Kanade scanner (developed as a low resolution prototype) and a version of the Rioux scanner [3] can achieve high frame rates, both require highly customized hardware.

In the direction of using fewer images, Sato and Inokuchi [29] describe a set of hierarchical stripe patterns to give range images with $\log N$ images, where N is the number of resolvable stripes. In particular, the images contain Gray codes, each camera pixel observes a bit code over time, and, at the finest resolution, each pixel is associated with the interior of a thin stripe. Caspi *et al.* [8] reduce the number of images further by using a color generalization of Gray codes. Finally, Hattori and Sato [19] refine the original hierarchical stripe technique by introducing sub-pixel offsets to the final stripe pattern to get finer resolution. Their approach is similar to spacetime analysis and uses $\log N$ stripes plus m shifted versions of the finest stripes.

Still fewer images are used by Carrihill and Hummel [7] who triangulate using two images: a ramp and a constant brightness projected image. As noted in Section 1, this technique is highly susceptible to sensor noise. Chazan and Kiryati [9] combine this method with hierarchical stripes to reduce noise susceptibility, and later Horn and Kiryati [20] developed novel piecewise linear patterns that required only a few more images than the Carrihill and Hummel approach.

The last set of triangulation techniques we describe are those suitable for capturing moving scenes, each under some kind of constraint. Hall-Holt and Rusinkiewicz [18] describe a method that consists of projected stripe patterns that vary over time. By finding nearest stripe patterns over time, a unique code can be determined for any stripe at any time. The constraint in this case is that the object move slowly to avoid erroneous temporal correlations. Proesmans *et al.* [26, 25] demonstrate a scanner which projects

a grid pattern onto the scene and matches the observed grid to the projected pattern by a global 2D grid optimization algorithm. In this case, the constraint is that the visible portion of the object consist of a single connected component. Boyer and Kak [6] project a color stripe pattern in which each window of spatially adjacent stripes has a unique color intensity configuration. Davies and Nixon [14] propose a color dot pattern with a similar spatial neighborhood property. In both cases, local neighborhoods must exhibit enough spatial coherence to preserve the windows. Finally, Chen *et al.* [10] describe a stereo vision method that also uses projected color stripes, but solves correspondences between edges in the two cameras through dynamic programming, thus obtaining a global optimum. The constraint in this case is surface monotonicity between the cameras. Each of the color pattern methods above has the additional constraint that the surface does not change the reflected color too much, e.g., doesn't cause a color channel to drop out completely.

In this paper, we describe a technique that admits to both a single image and a multi-image analysis. The projected color stripe pattern does impose the reflectance restriction noted above, however, we have found that many objects and subjects of interest (e.g., human skin) work well. The single image method resolves correspondences with dynamic programming, as does the method of Chen *et al.* [10], but we develop a multi-pass technique to overcome the the monotonicity constraint, and our method requires only a single camera. Further, we demonstrate a method that combines spacetime analysis and dynamic programming to derive accurate shape using a small number of images.

3 Multi-hypothesis code matching

The basic principle of optical triangulation is illustrated in Figure 2(a): an illumination pattern is projected onto an object and the reflected light is captured by a camera. The relative distance between a point in the illumination pattern and its position in the captured image is inversely related to depth, allowing the 3D position of the point to be reconstructed, assuming the camera and projector parameters are known.

The primary challenge in optical triangulation is obtaining correspondence between points in the projected pattern and pixels in the image. This correspondence problem is mitigated by two observations. First, as shown in Figure 2(a), the 2D correspondence problem reduces to determining correspondences between each row of the projected pattern and a row of the *rectified* camera image [15]. Second, we can choose whatever pattern we wish to project; therefore, we should choose a pattern that simplifies the correspondence. Laser triangulation scanners take this strategy to the extreme by projecting a narrow beam or plane of light that is easily identified in the image as a point or contour on the surface. Since very little information is available in each image, reconstruction from laser scanners typically requires a very large number of images. Multi-stripe techniques, on the other hand, often use a detailed pattern to obtain as much information about the scene as possible from a small number of images. In this paper, we treat the problem of obtaining correspondence using multi-stripe techniques, in particular from color stripe patterns of the form shown in Figure 2(b) that yield camera images like the one shown in Figure 2(c).

While the concepts from this paper are applicable to a wide range of patterns, we begin by considering specifically the case of patterns consisting of equal-width color stripes to be used for capturing shape with a single image. We can enumerate these stripes by a string of colors $P = (p_0, p_1, \ldots, p_N)$. The information we will use for triangulation is encoded in the transitions between colors. This sequence, call it $Q = (q_0, q_1, \ldots, q_{N-1})$, is comprised of elements $q_j = (q_j^r, q_j^g, q_j^b)$ where each color channel takes on a value of -1, 0, or 1 corresponding to a falling, constant, or rising transition, respectively. For example, $(0, -1, 1)$ indicates no change in the red channel, a fall from 1 to 0 in the green channel, and a rise from 0 to 1 in the blue channel. For convenience, we adopt a notation in which, q_j refers to the j-th projector edge, q refers to any (generic) projector edge, and q^c refers to the transition in color channel $c \in \{r, g, b\}$ of projector edge q.

The reflection of the projected pattern from the scene is detected in the image as a sequence of color edges, $E = (e_0, e_1, \ldots, e_{M-1})$ for each rectified sensor scanline, where (using the simplified notation mentioned above) a color edge $e = (e^r, e^g, e^b)$ is described by its 1D intensity gradients in each of the three color channels. Our objective is to solve the correspondence between the transition sequence Q and the edge sequence E.

Correctly identifying the correspondence between projected and imaged stripes, shown in Figure 2(b) and (c) respectively, brings out two key difficulties:

- **Mislabeling**: In addition to the projected pattern, the color of image pixels depends on factors such as surface reflectance and shading, viewing direction, color cross-talk between projector spectra and sensor filters, and sensor noise. Consequently, obtaining reliable estimates of color directly from pixel values is not at all straightforward, and misclassifications can result.

- **Occlusions**: Real scenes often have occlusions, shadows, and surface discontinuities. It is therefore not realistic to assume that every projected transition is visible in the image.

We address the mislabeling problem by introducing a technique that allows for *multiple hypotheses*. Rather than assigning a unique label to every stripe in the image, every label assignment is represented, along with its probability of matching. A final labeling is then obtained by applying a

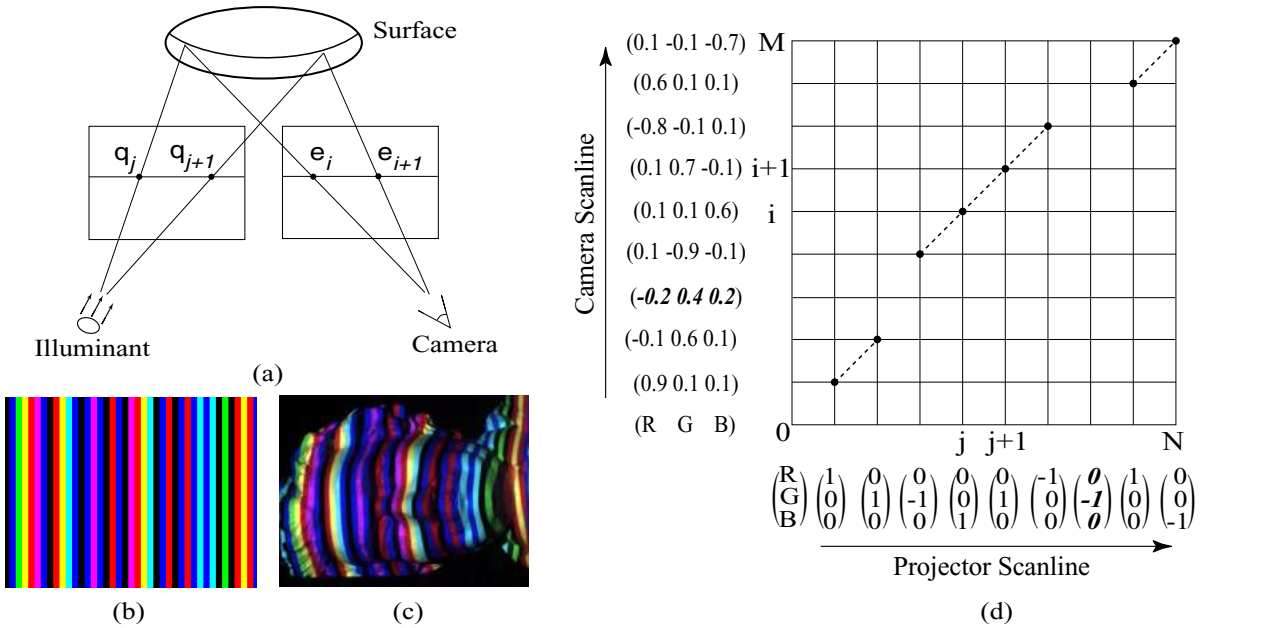


Figure 2. Summary of the one-shot method. (a) In optical triangulation, an illumination pattern is projected onto an object and the reflected light is captured by a camera. The 3D point is reconstructed from the relative displacement of a point in the pattern and image. If the image planes are rectified as shown, the displacement is purely horizontal (one-dimensional). (b) An example of the projected stripe pattern and (c) an image captured by the camera. (d) The grid used for multi-hypothesis code matching. The horizontal axis represents the projected color transition sequence and the vertical axis represents the detected edge sequence, both taken for one projector and rectified camera scanline pair. A match represents a path from left to right in the grid. Each vertex (j, i) has a score, measuring the consistency of the correspondence between e_i, the color gradient vectors shown by the vertical axis, and q_j, the color transition vectors shown below the horizontal axis. The score for the entire match is the summation of scores along its path. We use dynamic programming to find the optimal path. In the illustration, the camera edge in bold italics corresponds to a false detection, and the projector edge in bold italics is missed due to, e.g., occlusion.

global optimization technique that accounts for occlusions, shadows, and discontinuities.

Specifically, a global match hypothesis ϕ between the projected transition sequence Q and observed edge sequence E is defined as a sequence of integer pairs

$$\phi = \left\{ \begin{pmatrix} j_1 \\ i_1 \end{pmatrix}, \begin{pmatrix} j_2 \\ i_2 \end{pmatrix}, \ldots, \begin{pmatrix} j_\Phi \\ i_\Phi \end{pmatrix} \right\} \tag{1}$$

where Φ is the number of integer pairs of ϕ, $j_1 < j_2 < \ldots < j_\Phi$, and $i_1, i_2, \ldots, i_\Phi$ are distinct from each other. Each integer pair $(j_k, i_k)^{\mathrm{T}}$ indicates that edge e_{i_k} corresponds to the transition q_{j_k}.

The match ϕ is equivalent to a path in a 2D grid, as illustrated in Figure 2(d). The horizontal axis represents the projected transition sequence Q and the vertical axis represents the observed edge sequence E. The match ϕ represents a path from left to right in the grid, intersecting each row and each column no more than once.

The quality of a match is computed by assigning each vertex (j, i) a score, $\texttt{score}(q_j, e_i)$, measuring the consistency of the correspondence between edge e_i and transition q_j. The specific definition of $\texttt{score}(q_j, e_i)$ is described in Section 4.

The score of the entire match ϕ is the summation of scores of all the vertices in ϕ, defined as

$$\sigma(\phi) = \sum_{k=1}^{\Phi} \texttt{score}(q_{j_k}, e_{i_k}) \tag{2}$$

The optimal match is therefore defined as

$$\phi^* = \arg\max_{\phi} \{\sigma(\phi)\} \tag{3}$$

In general, the possible paths represented by the ϕ's may go up and down and have disconnected components due to occlusion, texture edges, etc. Therefore, the space of all possible matches is enormous, of size $O(M^N)$. A common technique for making this optimization problem tractable is to introduce an assumption of depth-ordering, or *monotonicity* [1]

$$i_1 < i_2 < \ldots < i_\Phi. \tag{4}$$

With this assumption, Eq. (3) may be solved efficiently using dynamic programming [1, 24, 17, 11, 2, 10, 22, 4]. The monotonicity assumption should be used with care, however, since it is violated in the presence of occlusions, and can

produce artifacts. In practice, we have observed that violations of the monotonicity assumption result in dropouts, i.e., portions of the scene that are not reconstructed. We show how this problem can be addressed by use of a multi-pass dynamic programming technique. First, however, we review the basics of dynamic programming in the following subsection.

3.1 Correspondence by dynamic programming

We adopt a notation similar to that of Cox *et al.* [11]. Let $G_{j,i}$ be the sub-grid defined by $[0, j] \times [0, i]$, and $\phi^*_{j,i}$ be the optimal path in $G_{j,i}$. Three possible configurations of $\phi^*_{j,i}$ exist: (1) it consists of vertex (j, i) and the optimal path $\phi^*_{j-1,i-1}$ in $G_{j-1,i-1}$, (2) it is entirely in the sub-grid $G_{j-1,i}$, or (3) it is entirely in $G_{j,i-1}$. In the latter two cases, $\phi^*_{j,i} = \phi^*_{j-1,i}$ or $\phi^*_{j,i} = \phi^*_{j,i-1}$, respectively. Consequently, $\sigma(\phi^*_{j,i})$ may be recursively computed as

$$\sigma(\phi^*_{j,i}) = \begin{cases} 0, & \texttt{if j = 0 or i = 0;} \\ \max \begin{cases} \sigma(\phi^*_{j-1,i-1}) + \texttt{score}(q_j, e_i), \\ \sigma(\phi^*_{j-1,i}), \\ \sigma(\phi^*_{j,i-1}) \end{cases} & \texttt{otherwise} \end{cases}, \tag{5}$$

The cost of the optimal solution ϕ^* to Eq. (3) is given by $\sigma(\phi^*_{N,M})$, and evaluating every $\sigma(\phi^*_{j,i})$ takes $O(MN)$ space and time. $\phi^*_{N,M}$ is computed by tracing back through the cost matrix [11], which takes $O(M + N)$ time. A common technique to reduce the complexity is to restrict the depth range to a user-defined value (e.g., 10% of the maximum possible depth range).

3.2 Multi-pass dynamic programming

A fundamental limitation of matching algorithms based on dynamic programming (DP, for short) is the assumption of monotonicity, which is violated in the presence of occlusions. An example of such a violation is shown in Figure 3(a). In the figure, a thin foreground element lies in front of a background plane. Due to the occlusion, the order of projected transitions and detected edges is not the same, resulting in a non-monotonic path in the grid, shown in Figure 3(b).

The DP algorithm therefore fails to find the optimal path; instead, it will identify the optimal monotonic solution. While this solution could potentially be quite different than the optimal path, in practice we have seen that it corresponds to a monotonic component of the optimal solution. In the case of Figure 3(b), DP identifies the sub-path consisting of (A, B, C, D, E). The rest of the optimal solution, sub-path (F, G), is itself monotonic and can be identified by applying DP on the sub-grid obtained by removing columns $(1, 2, 4, 5, 6, 9)$, and rows $(1, 2, 5, 6, 7, 8)$ from the original grid. The same procedure may be repeated until all rows and columns are exhausted. This procedure, which we call `MultiPassDP`, is summarized as follows

```
procedure  MultiPassDP(grid)
   set path to be empty;
   while(path1 := DP(grid) is not empty)
      path := path⋃path1;
      remove columns and rows in path1 from grid;
   return path;
end  MultiPassDP
```

`MultiPassDP` computes the monotonic components of the optimal path in multiple passes, enabling solution of correspondence problems with occlusions that are not possible with traditional DP. Instead of exhausting the positive monotonic components, $path1$, in the grid, the number DP passes can also be specified by a user, based on prior knowledge of how many "layers" of structure the scene contains.

4 One-shot patterns and scoring functions

The previous section describes the machinery needed to compute an optimal surface given a projected pattern and observed image. Capturing the shape of a scene from a single image is sometimes called "one-shot" scanning. In this section, we discuss design decisions in choosing a pattern to project and a scoring function to be used in computing the optimal one-shot surface.

4.1 De Bruijn illumination patterns

Choosing a good pattern to project is of critical importance for achieving accurate correspondence with optical triangulation techniques, particularly one-shot methods. Assuming the patterns and images are rectified, the pattern may be designed for a single scanline and replicated to produce a 2D vertical pattern. One design choice is whether to project a smooth or a piecewise-constant pattern. For a one shot method, encoding information in smooth intensity variations is difficult to do, because surface shading can affect these variations substantially. Thus, we choose to encode information on edges and resort to a piecewise-constant illumination pattern.

In addition, a good pattern has the property that correspondence between projected edges and observed edges is easy to determine. As noted earlier, each edge element of Q is comprised of three color edge transition labels that can each take values -1, 0, or 1. This value assignment implies an "edge alphabet" of 27 unique edges; however, since the edge $(0, 0, 0)$ is really no edge at all, 26 edges are actually available to us.

Projecting exactly 26 edges would lead to well-determined correspondence, since each edge is unique, but this would yield very sparse reconstructions. Alternatively,

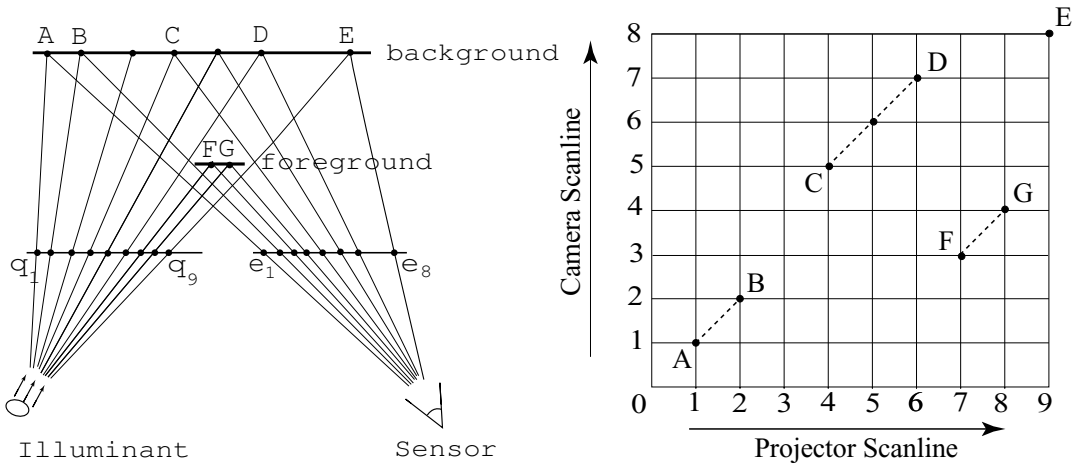


Figure 3. Violation of the monotonicity assumption. (a) 9 transitions, $q_1, q_2, \ldots, q_9$, are projected onto a scene comprised of a thin foreground surface against a background surface, and 8 edges, $e_1, e_2, \ldots, e_8$, are detected. Projected edge q_3 is occluded by the foreground element and not detected. (b) The resulting match grid shows that camera edge indices do not increase monotonically with corresponding projector edge indices. A multi-pass DP algorithm can recover the (A,B,C,D,E) path in the first pass, and the (F,G) path in the second pass.

we could lay out the same sequence of edges and repeat them as many times as needed to fill out the desired total number of projected edges. Consider, though, the case of an object that is 26 projected stripes in width. The dynamic programming algorithm will prefer to find a sequence of matches that corresponds to a single connected surface (rather than a set of scattered matches), but due to the repetition in the pattern, a set of equivalent answers will arise. In fact, such a repetive pattern can result in ambiguity for even larger objects. To minimize such ambiguity, we instead we seek to find a sequence that has a good *windowed uniqueness* property, i.e., a sequence that has the desired number of transitions with a small window size n such that each sub-sequence of n consecutive stripes is unique within the entire sequence.

Our goal now is to devise a color sequence P that yields an edge sequence Q with a good windowed uniqueness property. First, we consider the fact that given a color p_j, the next color p_{j+1} must be different in at least one channel for an edge transition to occur. If we think of these colors as being mapped to base-2 numbers $\{000, 001, \ldots, 111\}$ (i.e., black, blue, ..., white), then a legal edge is produced by performing a bitwise XOR (exclusive or) of a given color with a number in the range $\{001, \ldots, 111\}$. For example, the color index corresponding to green, 010, could change to 110 (yellow) by flipping the red channel, i.e., by XOR'ing with 100. We could then attain local and global non-periodicity if we could choose a sequence of XOR patterns (of which we have 7 to choose from) that are unique when taken in groups of n at a time.

De Bruijn sequences [16] are ideally suited to this problem. In particular, a k-ary de Bruijn sequence of order n is a circular sequence $d_0, d_1, \ldots, d_{k^n-1}$, where each element d_j is taken from a set of k values and for which any subsequence of length n appears exactly once. In our case, we can construct up to a 7-ary sequence with each element $d_l \in \{001, \ldots, 111\}$. Then, we can generate a color index sequence $(p_0, p_1, \ldots p_{7^n})$ by choosing an initial color $p_0 \in \{000, \ldots, 111\}$ and following the iteration:

$$p_{j+1} = p_j \ \texttt{XOR} \ d_j \tag{6}$$

In practice, we only needed 125 stripes and thus worked with $k = 5, n = 3$. We eliminated 110 and 111 from the de Bruijn sequence, as we found that simultaneous red and green transitions suffered the most from residual crosstalk errors after approximate decoupling (see Section 4.2). Figure 4 shows a complete color stripe pattern generated in this manner. Note that while de Bruijn sequences are not straightforward to derive, they have been previously tabulated by researchers in combinatorics. We use generators available online [28] to obtain de Bruijn cycles.

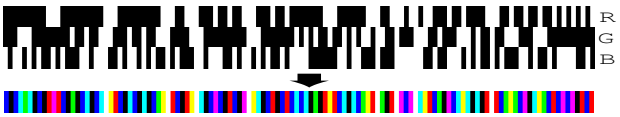


Figure 4. Using a de Bruijn sequence, we can generate binary R, G, and B patterns that combine to make a sequence for which each three consecutive color transitions are unique. This example is a complete sequence for $k = 5, n = 3$, which is used to generate the results in this paper.

4.2 Color edge detection

After the illumination pattern from the previous section is projected onto an object, a camera records an image of the reflected light. In an ideal world, the light from each projector color channel reaches only its correspondingly colored sensor pixel (e.g., red light is seen only by red pixels). In practice, however, each projector color channel has some influence on all three sensor channels, a phenomenon known as color crosstalk. This coupling is complicated by the intervention of a surface that may modify the projector spectrum in unknown ways before it is observed at the sensor. One solution would be to assume that the surface is spectrally uniform, so that we need only measure a projector-camera color crosstalk matrix that indicates how much each projector channel influences each camera channel. Caspi *et al.* [8], however, demonstrate that a less severe restriction can work fairly well. In particular, they relate observed camera color s to projector color p as:

$$\underbrace{\begin{pmatrix} s^r \\ s^g \\ s^b \end{pmatrix}}_{s} = \underbrace{\begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{bmatrix}}_{X} \underbrace{\begin{bmatrix} \rho^r & 0 & 0 \\ 0 & \rho^g & 0 \\ 0 & 0 & \rho^b \end{bmatrix}}_{F} \underbrace{\begin{pmatrix} p^r \\ p^g \\ p^b \end{pmatrix}}_{p} + \underbrace{\begin{pmatrix} o^r \\ o^g \\ o^b \end{pmatrix}}_{o} \tag{7}$$

where X is the projector-camera color channel crosstalk matrix, F is the scene albedo matrix at a point on the surface, and o is the ambient light observed by the camera for the same surface point. By pre-multiplying each camera color by X^{-1}, we obtain new camera colors:

$$\tilde{s} = X^{-1}s = \begin{pmatrix} \rho^r p^r + \tilde{o}^r \\ \rho^g p^g + \tilde{o}^g \\ \rho^b p^b + \tilde{o}^b \end{pmatrix} \tag{8}$$

where $\tilde{o} = X^{-1}o$. Using this model, crosstalk has largely been factored out so that each color channel can be analyzed independently and correlated more closely to projected colors.

Given a color-corrected camera scanline, we can now localize color edges by looking for local extrema in gradients (1D derivatives) in each of the color channels. In practice, however, this will lead to distinct localizations in each color channel. Instead, we compute a combined gradient function along a scanline that is comprised of the sum of the squares of the gradients in each of the color channels. The edges are then determined to be local maxima of this function, and their strengths are the color gradient values at the localized edges.

4.3 Edge-based scoring functions

We now consider the problem of evaluating a match between a projected color transition q and an observed edge e in an image by defining a score function $\texttt{score}(q, e)$. Let $e = (e^r, e^g, e^b)$, where $e^c \in [-1, 1]$ is the 1D intensity gradient of e in channel c, and let transition $q = (q^r, q^g, q^b)$, with $q^c \in \{-1, 0, 1\}$. e and q are consistent only if they match in all three channels. Accordingly, we use the following definition of `score`:

$$\texttt{score}(q, e) = \min_{c \in \{r,g,b\}} \{\texttt{consistency}(q^c, e^c)\} \tag{9}$$

where $\texttt{consistency}(q^c, e^c) \in [-1, 1]$ measures the agreement between corresponding color channels of q and e. For example, when $q^c = 1$, $\texttt{consistency}(1, e^c)$ should be 1 if e^c is sufficiently large, 0 if $|e^c|$ is sufficient small, and negative if e^c is negative. More formally, $\texttt{consistency}(q^c, e^c)$ is defined by the following equation, Eq. 10(a), and illustrated in Figure 5(a). For the cases of $q^c = 0$ and -1, $\texttt{consistency}(0, e^c)$ and $\texttt{consistency}(-1, e^c)$ are similarly defined in Eq. 10(b,c) and illustrated in Figure 5(b,c) respectively.

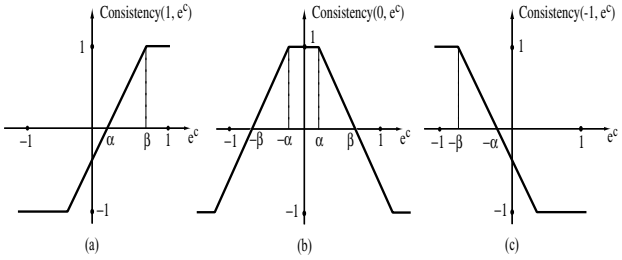


Figure 5. Consistency measure $\texttt{consistency}(q^c, e^c)$ between projector transition q^c and e^c. (a) $q^c = 1$, (b) $q^c = 0$, and (c) $q^c = -1$.

$$\begin{aligned} &\texttt{consistency}(1, e^c) = \texttt{CLAMP}(\tfrac{e^c - \alpha}{\beta - \alpha}; -1, 1) && (a) \\ &\texttt{consistency}(0, e^c) = \texttt{CLAMP}(1 - \tfrac{|e^c| - \alpha}{\beta - \alpha}; -1, 1) && (b) \\ &\texttt{consistency}(-1, e^c) = \texttt{consistency}(1, -e^c) && (c) \end{aligned} \tag{10}$$

where

$$\texttt{CLAMP}(x; x_0, x_1) = \begin{cases} x_0 & \texttt{if } x < x_0; \\ x & \texttt{if } x_0 < x \leq x_1; \\ x_1 & \texttt{if } x_1 < x. \end{cases}$$

and $0 \leq \alpha < \beta \leq 1$ are *soft thresholds* that are chosen based on the uncertainty of edge measurement. In particular, gradients in the range of $[\alpha, \beta]$ can be classified with fractional values that reflect their uncertainty, whereas gradients with absolute values that are sufficiently large or small are assigned either -1, 0, or 1. The decision on how to label each edge is deferred to the global optimization stage. In the degenerated case when $\alpha = \beta$, the gradients are classified with *hard thresholds*, as in [6]. The larger the value of $\beta - \alpha$, the more uncertain `consistency`. Less certain consistency measures are useful when there are significant differences in the intensity of projected and reflected patterns, due for instance to noise, shading, or surface texture.

Note that edge pair (q, e) will get matched by DP only if $\texttt{score}(q, e)$ is positive and it will not get matched if any

of its channel consistency measures is negative. As a result, clamping of negative consistency values is not necessary in theory, but in practice avoids possible numerical problems of large negative numbers when α and β and nearly equal.

5 Color-coded spacetime analysis

One-shot patterns are particularly useful in cases where all measurements must be captured at the same time, for instance in the case of reconstructing the instantaneous shape of a moving object. Static scenes, on the other hand, provide the opportunity to capture multiple measurements. Instead of projecting a single pattern, it is possible to project several patterns to improve accuracy or completeness of the resulting reconstruction.

One approach to increasing the resolution is to take a set of images in which the one-shot projector pattern is shifted a pixel to the right between photographs, followed by combining the one-shot result for each image into a single range map. This approach, while feasible, does not make the most of the opportunity to take multiple shots. One-shot scanning techniques are sensitive to errors as a result of surface discontinuities and texture, both of which can bias the calculated location of edges or introduce false edges. Curless and Levoy [12] describe this phenomenon in detail (for the case of determining the center of a Gaussian, rather than location of an edge) and demonstrate how these problems are addressed through the use of *spacetime analysis*. In their case, a laser stripe is projected onto an object and is swept slowly over its surface. The reflected light is captured by a camera to produce a sequence of images during the sweep. Tracking the intensities recorded for a single line of sight from the sensor gives a temporal profile of light reflected from a single point on the surface. The peak of this profile corresponds to the time at which the stripe passes over that pixel and can be estimated to sub-pixel accuracy. The advantage of the spacetime approach is that it is far less sensitive to discontinuities and texture, and has been shown to produce superior reconstructions [12].

A disadvantage of previous applications of spacetime analysis is that a very large number of images are required to ensure that the stripe passes over every pixel in the image. We show that spacetime analysis may be adapted and incorporated into our multi-hypothesis code matching framework to generate high-quality range data using a much smaller number of images with the use of a projector instead of a laser scanner. As in the one-shot case, we first need to choose both the illumination patterns and the scoring function.

5.1 Smoothed, shifted de Bruijn patterns

Figure 6(a) illustrates how the spacetime method works in the context of shifting color stripe patterns. The shift in the pattern over time defines a temporal profile for each projected ray and each pixel in the image. Thus, by matching sensor profiles to projector profiles, we can reconstruct the surface. In general, each sensor profile will triangulate to a projector line of sight that is *between* projector pixels. To do sub-pixel interpolated matching, we require that the projection patterns be smooth relative to the rate of shifting the pixels. In addition, we seek to project as few images as possible while allowing reliable correspondence to be determined between projector and sensor profiles. For these reasons, we employ a smoothed de Bruijn color pattern; i.e., we take the one-shot color pattern, smooth it with a Gaussian filter, and project shifted copies over time.

Each camera pixel profile has a number of color transitions, depending on the number of patterns projected and the rate v at which the pattern shifts. If the profile is long enough to contain at least n transitions, where n is the order of the de Bruijn sequence, the correspondence between image and pattern may be uniquely determined in principle, based on the windowed uniqueness property of de Bruijn sequences. However, we can use fewer frames and allow DP to resolve the ambiguity. In fact, by defining a `score` function for spacetime patterns, we can again employ the multi-hypothesis code matching technique to derive an *optimal match*, that better accounts for noise and other sources of measurement error, as shown in Figure 6(b).

5.2 Spacetime scoring function

In spacetime analysis, instead of comparing projected edges to observed edges, we compare the temporal pattern at each projector pixel to the temporal pattern recorded at each camera pixel. We can still describe the matching problem in terms of a scoring function $\texttt{score}(q, e)$, but now the elements q^c and e^c ($c \in \{r, g, b\}$) are each vectors in a T-dimensional space, where T is the number of frames in the sequence. In principle, after color calibration, we should be able to measure how close the projector and camera patterns are by computing the difference between e^c and the estimated reflection $\rho^c \cdot q^c + \tilde{o}^c$ (see Eq. 8). Since ρ^c and $\tilde{o}^c$ are unknown, we can estimate the best values that minimize differences between measurement and prediction. A shortcoming of this approach arises when e^c is on an oblique surface, for example, close to the target's contour tangent to the camera's viewing direction, where e^c is usually very small compared to the pixel colors on a frontal parallel suface. In this case, the difference between q^c and e^c is no larger than $\|e^c\|^2$ by setting $\rho^c = 0$ and $\tilde{o}^c = 0$. The result is that bad matches are not penalized signficantly for low intensity e^c's. To counter this problem, we have designed a simple symmetric per channel cost ("inconsistency") function:

$$\begin{aligned} \texttt{cost}^c(q^c, e^c) &= \min_{a,b} \| \, a \cdot q^c + b \cdot \vec{1} - e^c \, \|^2 \\ &+ \min_{a,b} \| \, a \cdot e^c + b \cdot \vec{1} - q^c \, \|^2 \end{aligned} \tag{11}$$

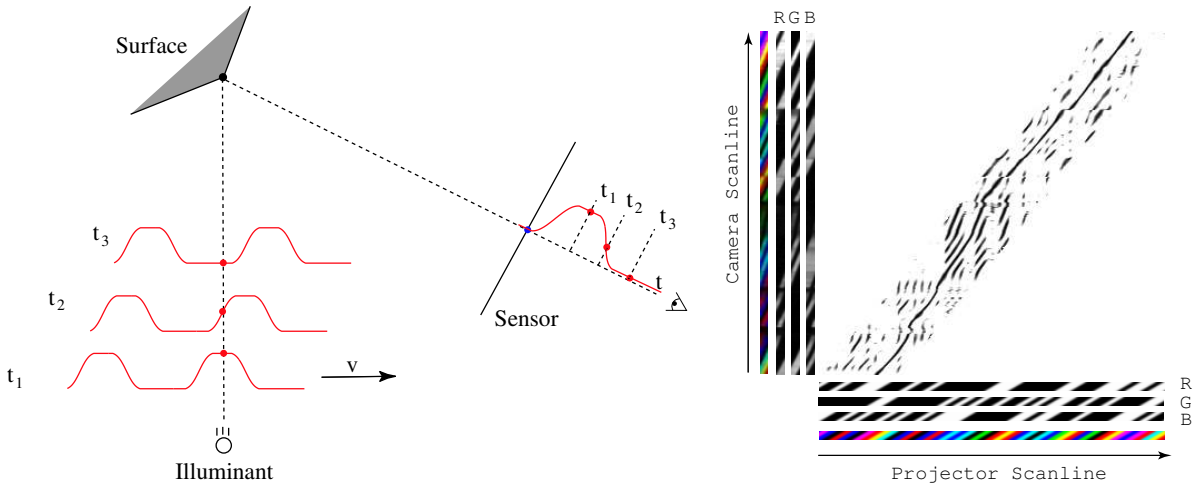


Figure 6. Color-coded spacetime analysis. (a) The curves represent projected and reflected intensity profiles in the red channel. (The green and blue channel profiles are not shown to simplify the figure.) As the illumination pattern shifts to the right in time, we can track a single line of sight from the projector to a point on the surface. The light reflects from that surface point to the camera along a sensor line of sight. We can see that, in the continuous case, the projection pattern will be reproduced *over time* at the sensor. In the discrete case, the sensor records discrete samples at times $t_1, \ldots, t_T$, which can later be matched to the projection pattern. (b) The temporal profiles, instead of spatial edges, are matched using dynamic programming. The horizontal axis shows the projected r, g, b channels separately and as a combined color pattern. The vertical axis shows the recorded color patterns. The gray-scale image illustrates a score grid through which DP finds a globally optimal path. (The score is in proportion to the darkness.) In this example, we apply the depth range constraint, which implies that grid vertices outside of a prescribed diagonal band are ruled out by assigning their scores to be 0.

where a, b are scalar coefficients independently optimized in each of the addends, and $\vec{1}$ is a T-dimensional vector of 1's. The total cost, $\texttt{cost}(q, e)$ is then the sum over all three channels. Note that the smaller $\texttt{cost}(q, e)$ is, the more consistent q and e are. Since the matching problem, Eq. 3, is formulated as a maximization over positive numbers, the $\texttt{score}(q, e)$ is defined as

$$\texttt{score}(q, e) = C_0 - \texttt{cost}(q, e) \tag{12}$$

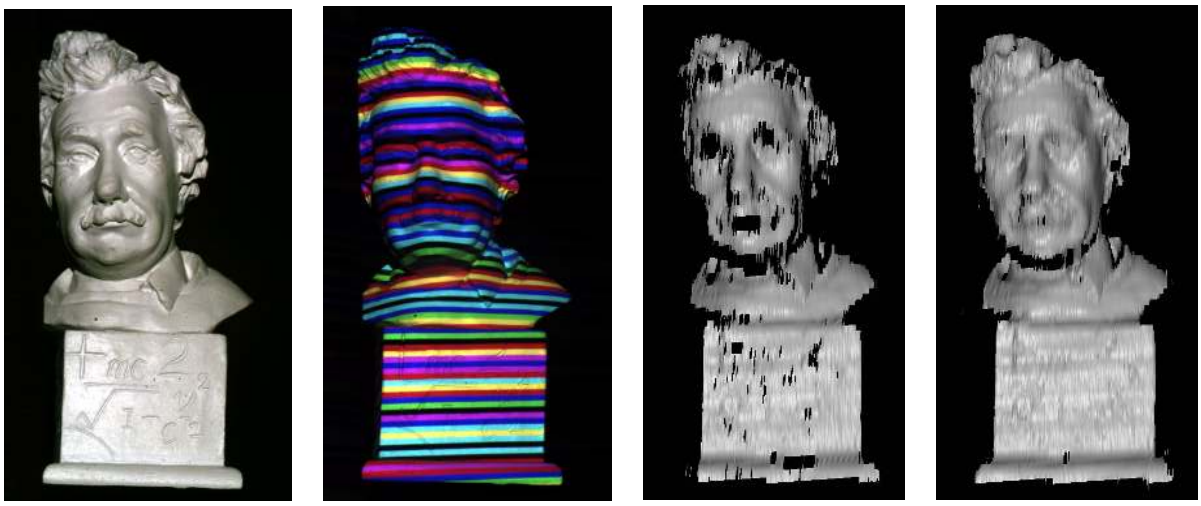
where C_0 is global constant between $\underline{s} = \min_{q,e}\{\texttt{score}(q, e)\}$ and $\bar{s} = \max_{q,e}\{\texttt{score}(q, e)\}$. Recall that DP will not match (q, e) pairs with negative scores. If $C_0 < \underline{s}$, $\texttt{score}(q, e)$ is negative for every (q, e) pair and the optimal match between projected and observed edge sequences is simply empty. If $C_0 > \bar{s}$, $\texttt{score}(q, e)$ is positive for every (q, e) pair and DP will try to match every possible edge pair without violating monotonicity constriant. In short, too small C_0 will result in false dropouts and too large C_0 may introduce false matches. In practice, we choose $C_0 = \underline{s} + 0.2(\bar{s} - \underline{s})$, which works well for our experiment setup.

5.3 Sub-pixel matching

The dynamic programming technique only gives camera and projector correspondence up to pixel resolution. Sub-pixel correspondence can be obtained using a post-processing step so that each camera pixel can be matched *between* projector pixels. Specifically, for each corresponding pair of camera pixel e_i and projector pixel q_j generated by DP, if $\texttt{score}(q_j, e_i)$ is larger than both $\texttt{score}(q_{j-1}, e_i)$ and $\texttt{score}(q_{j+1}, e_i)$, a parabola is fit to the three points $(j-1, \texttt{score}(q_{j-1}, e_i))$, $(j, \texttt{score}(q_j, e_i))$, and $(j+1, \texttt{score}(q_{j+1}, e_i))$ and the optimal matching position is obtained by computing the peak of the parabola. If $\texttt{score}(q_j, e_i)$ is not larger than its neighbors, both $\texttt{score}(q_{j-1}, e_i)$ and $\texttt{score}(q_{j+1}, e_i)$ are checked to see whether they are local maxima. If again no peak is found, the procedure repeats once more, expanding in both directions. If still no peak is found, the integer solution generated by DP is retained.

6 Results

We have developed an experimental system for testing our one-shot and spacetime shape capture methods. The hardware consists of a Kodak DCS520 digital still camera and a



(a) (b) (c) (d)

Figure 7. Comparison between DP and CFG [6]. (a) Photo of original Einstein bust. (b) Stripe image used for one-shot reconstruction. The bust was photographed on its side, but the image shown here is rotated by 90 degrees for convenient visualization. (c) Shaded rendering of the CFG reconstruction. (d) Shaded rendering of the DP reconstruction. In areas of high curvature, false edges are more prevalent, and result in dropouts in the CFG reconstruction, whereas DP is free to ignore such edges in pursuit of a global optimum.

Compaq MP1800 digital projector. To simulate resolutions more comparable to a video camera (the sensor type we ultimately plan to use for realtime capture), we downsampled the Kodak images by a factor of 2X in each dimension, yielding 864x576 images. The projector operates at 1024x768 resolution. For geometric calibration, we image a checkerboard textured plane in a variety of poses. For each pose, we also *project* a distinct checkerboard pattern onto the plane and take an additional image. The images taken without the projected pattern are used to estimate the camera intrinsics and plane poses (see Bouguet [5]). For the remaining images, we can compute the 3D coordinates of the the projected pattern features (corresponding to projector rays) and thus calibrate the projector. We employ a linear projective model for both the camera and projector. In addition, to improve color channel alignment in the digital camera, we image the checkerboard textured plane an additional time and compute separate 2D homographies for the red and blue channels relative to the green.

The X matrix in Eq. 7 is approximated by projecting solid red, green, and blue patterns to a fronto-parallel white board and capturing three images accordingly. The mean colors of the three captured images constitute the three columns of X.

We must note that, while we have taken some measures to reduce color misalignments and account for color crosstalk, we still observe some residual misalignments and non-linear crosstalk behaviors that have not been accounted for. As a result, some of the rendered reconstructions shown in this section exhibit coherent ridges at color transition boundaries. We are currently developing techniques for calibrating away these residual artifacts.

6.1 One-shot Scanning

We have tested the one-shot capture method on a variety of scenes. In each case, the projected pattern consists of de Bruijn generated sequences with 7 pixel wide stripes (roughly 150 stripes total).

First, we compare the dynamic programming approach to the *Crystal Fitting and Growing* (CFG) algorithm developed by Boyer and Kak [6], for which results are shown in Figure 7. CFG first labels each edge transition code by computing the signs of intensity transitions in the three channels, followed by matching the labeled edge sequence by iteratively finding the longest matching sub-sequences. In practice, we have found this technique to be fragile in the presence of erroneous edge labels, resulting in outlier points and holes in the reconstruction as shown in the figure. One-pass DP, on the other hand, is able to cope with these mis-labelings by searching for a globally optimal, monotonic solution. The result is fewer holes in the reconstruction.

The Einstein bust in the previous example is fairly "white," similar to the color crosstalk calibration target. To test sensitivity to fairly non-white surfaces, we took one-shot images of human hands (Figure 1) and of a painted porcelain cat (Figure 8). In both cases, the reconstructions are fairly accurate and complete, including, for instance, regions around the cat's red noise and over its orange body. Particularly dark areas result in holes in the reconstructions, since edges in these areas are not detected by the sensor.

(a) (b) (c)

Figure 8. Sensitivity to surface reflectance. (a) Photo of original porcelain cat model. (b) Stripe image used for one-shot reconstruction. (c) Shaded rendering of the DP reconstruction. Although this model is not "colorless" the reconstruction behaves reasonably well. Completely black regions, of course, lead to range dropouts.

Note that one noticeable artifact in Figure 1 is the occurrence of false edge extensions at the boundary of the object. This artifact arises, because DP is free to add points onto the boundary while still increasing the score. Thresholding out low intensity gradients minimizes the effect. Further, these points can be downweighted when used, e.g., to reconstruct surfaces [30]

Finally, to demonstrate the multi-pass DP method, we show a simple example that violates the monotonicity constraint: a finger in front a piece of cloth (Figure 9). Using a single DP pass, the finger is lost in favor of reconstructing the cloth background. The second pass, however, recovers much of the lost finger. Note that this example, together with Figure 1, also demonstrates that our scanning method is applicable to scenes with disconnected components, which can not be reconstructed by methods that rely on traversing edge graphs spatially within a single connected component (as in the case of, e.g., Proesmans *et al.* [26, 25]).

In the above experiments, each range map takes less than 1 minute to compute offline using a 900 MHz Pentium III PC. The exact time generally depends on the number of edges detected and the depth range of the scene. Reconstructions typically contain tens of thousands of range points, with dense vertical sampling along stripe edges and comparatively sparse sampling horizontally. For a triangulation angle of approximately 17 degrees, and an x-y field of view of about 40cm x 25cm, we have found plane-fit accuracy to have a standard deviation of 0.18mm.

6.2 Spacetime Analysis

To show the improvement possible using spacetime analysis for static scenes, we have done a more detailed study of the Einstein bust, as shown in Figure 10. In this case, we project a shifted sequence of 7 patterns onto the bust. For comparison with a non-spacetime method, we first choose the same pattern as in the one-shot method, shift the pattern by one pixel 7 times, and independently estimate a range map for each image. We then combine these range maps into a single high resolution range map as shown. For the spacetime method, we blur the same pattern with a Gaussian filter (σ = 1.5 pixels), shift it by two pixels at a time, and perform the reconstruction described in Section 5. As the figure shows, the spacetime method does a substantially better job of resolving fine detail. In particular, the edge detection method used in the one-shot technique is susceptible to the rapid shading changes in high curvature areas, whereas the spacetime technique is much less so. Further, while the plane-fit accuracy of the multiple one-shot method does not improve with more images, the spacetime method exhibits significantly improved accuracy, down to a standard deviation of 0.048mm, almost four times less noisy.

7 Conclusion and Future Work

This paper presents a general multi-pass dynamic programming algorithm to solve the multiple hypothesis code matching problem in structured light scanning. The algorithm is applied to two specific scanning methods: a one-shot scanning method suitable for measuring range data for moving objects, and a spacetime method which generates high-resolution range data for static scenes.

This work has potential for improvement and future research in several directions. In the short term, we hope to mitigate the effects of the occasional ridges induced by color edges, as noted in the previous section. We believe that a self-calibration in which the projected color patterns are first observed by the camera and characterized directly in cam-

era space could be employed. We also hope to implement a realtime capture (possibly offline processing) system using synchronized video and projection. In addition, we hope to experiment with using spacetime analysis to reduce the number of images required to reconstruct shapes. Finally, we hope to explore the reconstruction of 3D shape, reflectance, and motion models using the system described in this paper as a starting point.

Acknowledgments

The support of Intel Corporation, Microsoft Corporation, and the National Science Foundation under grants IIS-0049095 and DMS-9803226 is gratefully acknowledged. The authers would also like to thank Brett Allen for insightful discussions.

References

[1] H. H. Baker and T. O. Binford. Depth from edges and intensity based stereo. In *Int. Joint Conf. on Artificial Intelligence*, pages 631–636, 1981.

[2] P. N. Belhumeur. A bayesian approach to binocular stereopsis. *Int. J. on Computer Vision*, 19(3):237–262, 1996.

[3] J. Beraldin, M. Rioux, F. Blais, J. Domey, and L. Cournoyer. Registered range and intensity imaging at 10-mega samples per second. *Optical Engineering*, 31(1):88–94, 1992.

[4] A. F. Bobick and S. S. Intille. Large occlusion stereo. *Int. J. on Computer Vision*, 33(3):181–200, 1999.

[5] J.-Y. Bouguet. *Camera Calibration Toolbox for Matlab*. http://www.vision.caltech.edu/bouguetj/calib_doc/index.html, 2001.

[6] K. L. Boyer and A. C. Kak. Color-encoded structured light for rapid active ranging. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9(1), 1987.

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

[8] D. Caspi, N. Kiryati, and J. Shamir. Range imaging with adaptive color structured light. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(5), 1998.

[9] G. Chazan and N. Kiryati. Pyramidal intensity ratio depth sensor. Technical Report 121, Center for Communication and Information Technologies, Department of Electrical Engineering, Technion, Haifa, Israel, Oct 1995.

[10] C. Chen, Y. Hung, C. Chiang, and J. Wu. Range data acquisition using color strucred lighting and stereo vision. *Image and Vision Computing*, 15:445–456, 1997.

[11] I. J. Cox, S. L. Hingorani, S. B. Rao, and B. M. Maggs. A maximum likelihood stereo algorithm. *Computer Vision and Image Understanding*, 63(3):542–567, 1996.

[12] B. Curless and M. Levoy. Better optical triangulation through spacetime analysis. In *Int. Conf. on Computer Vision*, pages 987–994, June 1995.

[13] Cyberware Inc., http://www.cyberware.com/. *Model 15 scanner*, 2001.

[14] C. J. Davies and M. S. Nixon. A hough transformation for detecting the location and orientation of three-dimensional surfaces via color encoded spots. *IEEE Trans. on Systems, Man, and Cybernetics*, 28(1B), 1998.

[15] O. Faugera. *Three-Dimensional Computer Vision*. MIT Press, 1993.

[16] H. Fredricksen. The lexicographically least debruijn cycle. *Journal of Combinatorial Theory*, 9(1):509–510, 1970.

[17] D. Geiger, B. Ladendorf, and A. Yuille. Occlusions and binocular stereo. *Int. J. on Computer Vision*, 14(3):211–226, 1995.

[18] O. Hall-Holt and S. Rusinkiewicz. Stripe boundary codes for real-time structured-light range scanning of moving objects. In *Int. Conf. on Computer Vision*, pages 359–366, 2001.

[19] K. Hattori and Y. Sato. Accurate rangefinder with laser pattern shifting. In *Int. Conf. on Pattern Recognition*, pages 849–853, 1996.

[20] E. Horn and N. Kiryati. Toward optimal structured light patterns. *Image and Vision Computing*, 17, 1999.

[21] Impact Studio, http://www.laserscan3d.com/. *3D Laser Scanning Services*, 2001.

[22] H. Ishikawa and D. Geiger. Occlusions, discontinuities, and epipolar lines in stereo. In *Eur. Conf. on Computer Vision*, pages 232–248, 1998.

[23] T. Kanade, A. Gruss, and L. Carley. A very fast vlsi rangefinder. In *Int. Conf. on Robotics and Automation*, volume 39, pages 1322–1329, April 1991.

[24] Y. Ohta and T. Kanada. Stereo by intra- and inter-scanline searching using dynamic programming. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 7(2):139–154, 1985.

[25] M. Proesmans, L. V. Gool, and A. Oosterlinck. Active acquisition of 3d shape for moving objects. In *Int. Conf. on Image Processing*, pages 647–650, 1996.

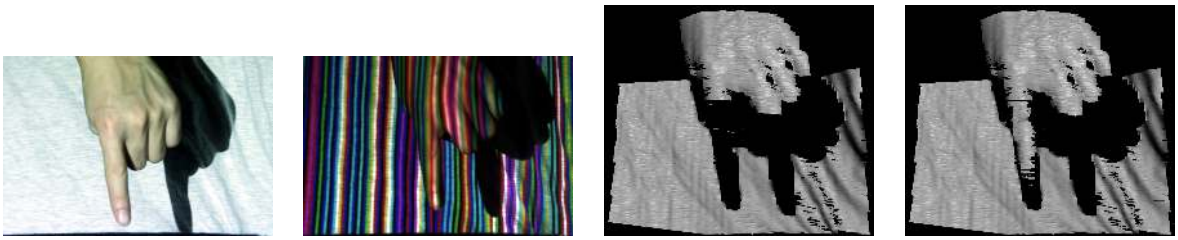
[26] M. Proesmans, L. V. Gool, and A. Oosterlinck. One-shot active 3d shape acquization. In *Int. Conf. on Pattern Recognition*, pages 336–340, 1996.

[27] M. Rioux, G. Bechthold, D. Taylor, and M. Duggan. Design of a large depth of view three-dimensional camera for robot vision. *Optical Engineering*, 26(12):1245–1250, Dec 1987.

[28] F. Ruskey. *The Combinatorial Object Server*. http://www.theory.csc.uvic.ca/~cos/, 2000.

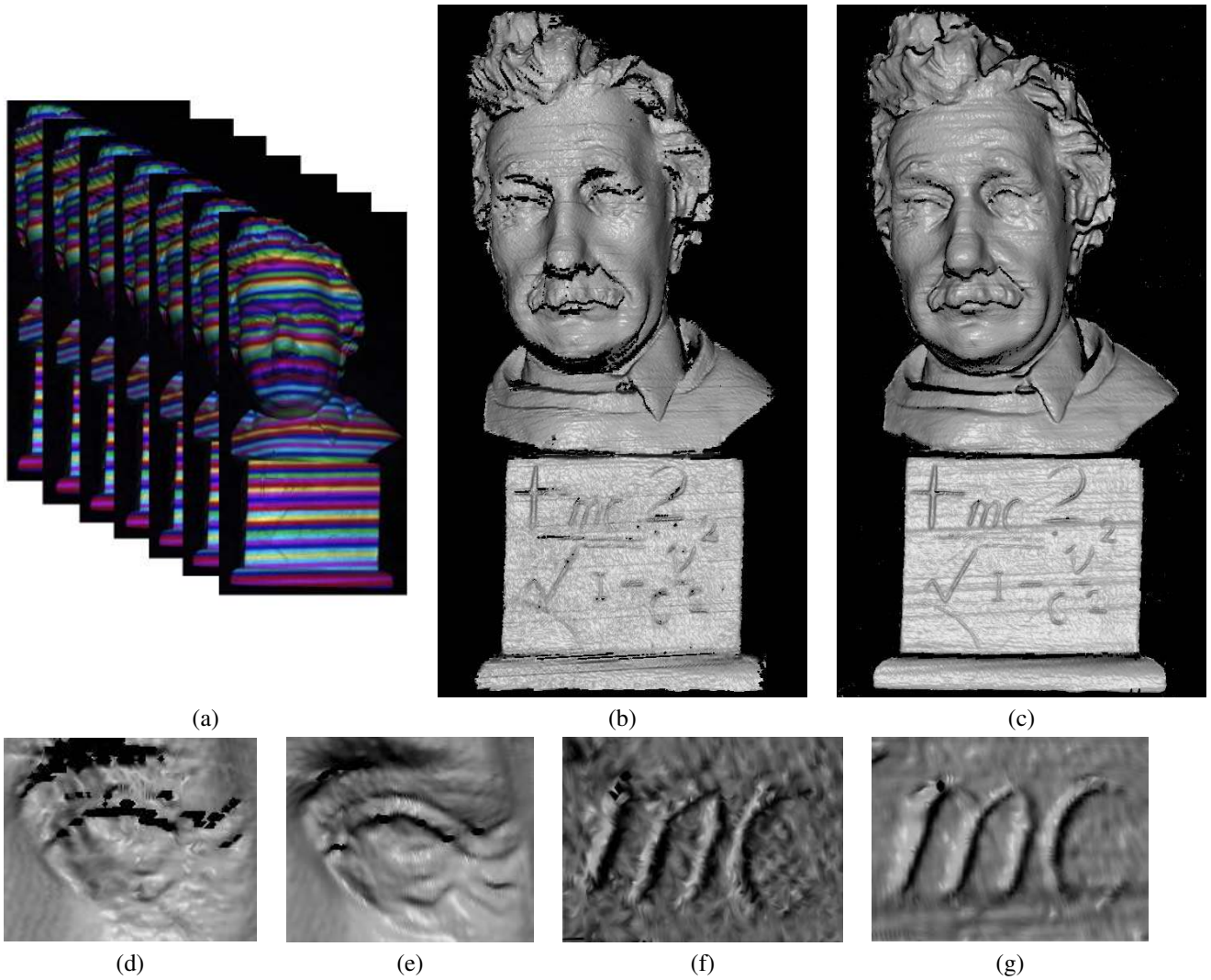
[29] K. Sato and S. Inokuchi. Three-dimensional surface measurement by space encoding range imaging. *Journal of Robotic System*, 2:27–39, 1985.

[30] G. Turk and M. Levoy. Zippered polygonal meshes from range images. In *SIGGRAPH*, pages 311–318, 1994.



(a) (b) (c) (d)

Figure 9. Comparison between one-pass DP and two-pass DP. (a) Photo of original scene of a hand and finger in front of a cloth background. (b) Stripe image used for one-shot reconstruction. (c) Shaded rendering of a one-pass DP reconstruction. (d) Shaded rendering of a two-pass DP reconstruction. The second pass recovers most of the finger that violated monotonicity and was not recoverable in a single pass. The "double-finger" hole in the background corresponds to projector and sensor visibility shadows.



(a) (b) (c)

(d) (e) (f) (g)

Figure 10. Comparison between multiple one-shots and spacetime analysis. (a) Stack of 7 stripe images taken of the Einstein bust for use with spacetime analysis. (b) Shaded rendering of reconstruction produced by combining 7 one-shot results (using shifted one-shot patterns). (c) Shaded rendering of reconstruction produced by spacetime analysis (using the patterns in (a)). (d) and (e) Renderings of the left eye (on right side of the (b) and (c) images) using multiple one-shots and spacetime, respectively. Notice the improved resolution in the wrinkles under spacetime. (f) and (g) Renderings of the letters "mc" on the base of the bust using multiple one-shots and spacetime, respectively. Notice the crisper, less noisy reconstruction under spacetime.