Measuring Depth: Spatial Light Coding

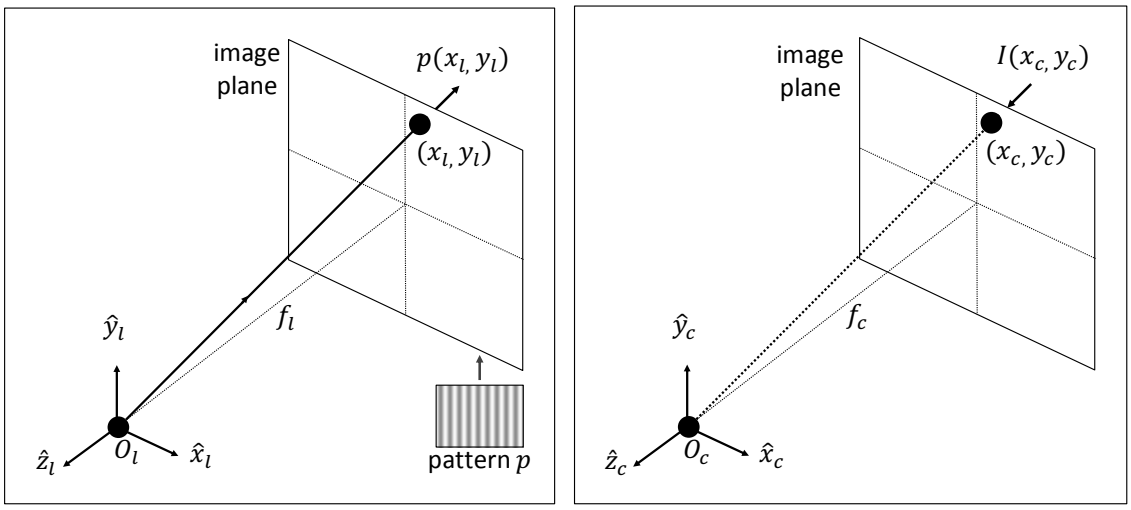
Quote.

Author

In this chapter, we discuss techniques that use spatially coded active illumination for measuring scene depths. The light source is modeled as a point source, with a 2D array of pixels in front of it. Each pixel is defined by its intensity and color transmittance. The intensity and color of a light ray emitted from the source passing through a pixel is given by the pixel's transmittance. The 2D array of pixel transmittance values is called the projected pattern or image. The transmittance of every pixel can be individually controlled [1], so that the source can be modeled as a projector that emits or *projects* a 2D spatial intensity pattern onto the scene. This is illustrated in Figure 1 (a). The point source is placed at the 3D location O_l, called the source's center of projection.

The camera is modeled as a pinhole (perspective projection model) with a 2D image plane. Camera's center of projection is placed at the 3D location O_c. This is illustrated in Figure 1 (b). Note that the camera and the light source have similar image formation geometry. The camera forms the image of the 3D world onto a 2D image plane, whereas the light source projects the image on its 2D image plane onto the 3D world. Thus, a spatially coded light source can be considered to be an *inverse camera*.

[1] The pixel array is typically implemented with 2D spatial intensity modulators, such as liquid crystal displays (LCDs), digital micro-mirror devices (DMDs) or a liquid crystal on silicon devices (LCOS). Color is modulated by using color filters, color wheels or dispersive/diffractive optics. For some techniques, a point laser source that is mechanically scanned in different directions can also be used as a spatially coded light source.



(a) Light source model (b) Camera model

Fig. 1. **Light source and camera models for spatial light coding.** Both the light source and the camera are modeled as pinhole devices, having an optical center and a 2D image plane. The light source projects a 2D intensity pattern onto the 3D world, whereas the camera maps the 3D world onto a 2D image plane.

1. DEPTH FROM ACTIVE TRIANGULATION

Most techniques discussed in this chapter are based on the principle of depth from active triangulation. Triangulation is used in binocular stereo, the primary process used by humans for estimating scene depths. In stereo, two cameras (eyes in humans) capture two images of a scene from different view points. Then, corresponding pixels (pixels on which the same scene point is imaged) are identified between the two images. The rays joining the corresponding pixels with the respective camera centers are then intersected to estimate the 3D location of the scene point. This process is called passive triangulation as both cameras image the scene passively.

In depth from active triangulation, one camera is replaced with an active spatially coded light source. Depth is computed by establishing correspondence between a camera pixel and the corresponding light source pixel (or a column of pixels). Once a correspondence is established, depth is computed by geometric triangulation, similar to the passive approach. These techniques are also known as structured light methods, and currently form the workhorse in many robotics applications where high resolution 3D imaging is required, including industrial assembly and inspection. Several consumer 3D imaging devices such as Microsoft Kinect (first generation) and hand-held laser scanners are structured light devices based on active triangulation.

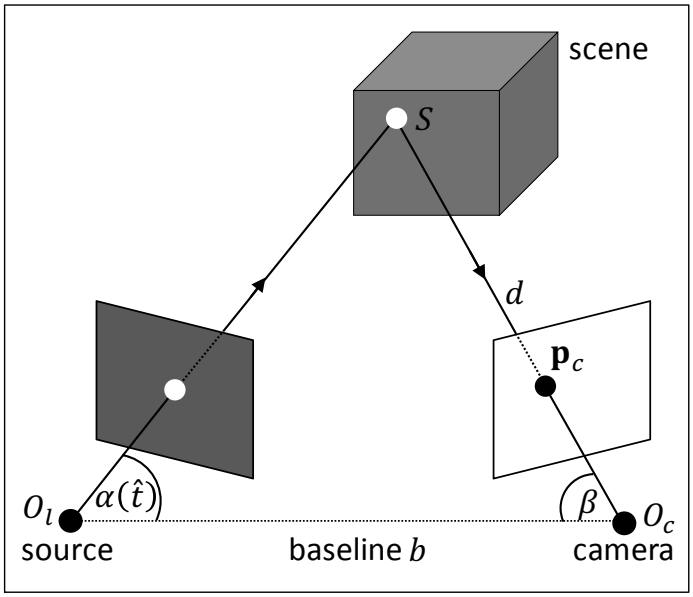


Fig. 2. **Point scanning:** The light source emits a single light beam and illuminates a single scene point. The camera captures the image of the scene. The illuminated scene point is imaged at a camera pixel called the illuminated pixel. The 3D location of the scene point is estimated by triangulating, i.e., by intersecting the light beam and the camera ray joining the illuminated pixel and the camera's optical center.

2. POINT SCANNING

The first active triangulation approach is point (or spot) scanning [Forsen 1968]. The light source emits a single light beam, thus illuminating a single scene point [2]. Let the scene point be S. The camera captures an image $I(x_c, y_c)$ of the scene. Let S be captured at pixel location $\hat{\mathbf{p}}_\mathbf{c} = (\hat{x_c}, \hat{y_c})$ in the camera image. This is illustrated in Figure 2.

Let the 3D locations of the camera and the projector centers be at O_c and O_l, respectively. The line joining the camera and the projector centers is called the baseline. Let $\vec{p_c}$ be the unit vector along the camera ray from O_c to pixel $\mathbf{p_c}$. Let α and β be the angle between the light beam and the camera ray $\vec{p_c}$, respectively. Since S lies at the intersection of the light beam and the camera ray, its 3D location can be found by triangulation. In particular, using trigonometry, the distance d between S and the camera center O_c is given by:

[2] For simplicity of exposition, we assume that the diameter of the beam is infinitesimally small.

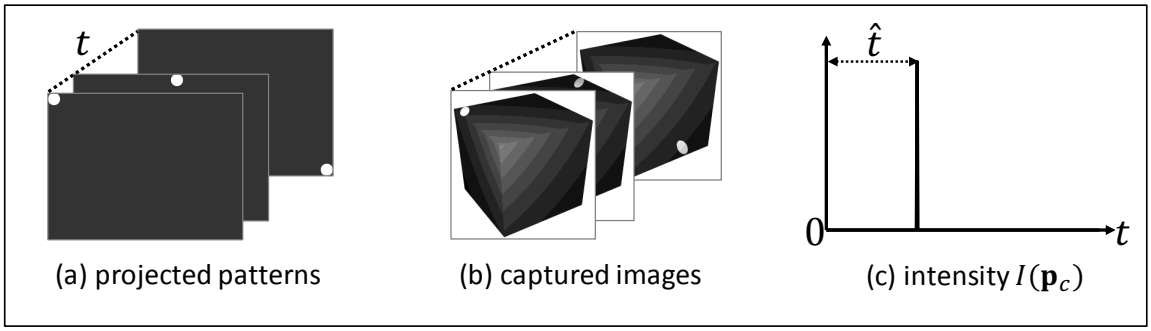


Fig. 3. **Point scanning of a scene.** (a) The light beam is scanned so that the entire scene is illuminated sequentially, one point at a time. This is similar to a projector projecting patterns with a single on pixel. The location of the on pixel is scanned over the projector image plane. (b) For each light beam orientation, the camera captures an image, creating an image stack. (c) The time instant when the scanning light beam illuminates the scene point imaged at a camera pixel can be estimated by identifying the peak in the pixel's temporal intensity profile.

$$d = \frac{b \tan\alpha \tan\beta}{\sin\beta\,(\tan\alpha + \tan\beta)}\,, \tag{1}$$

where $b = |O_c O_l|$ is the length of the baseline. The baseline length b and the angles α and β are estimated by measuring the intrinsic geometric parameters of the light source and the camera, and their relative geometric pose. This process is called geometric calibration. For details, see [Lanman and Taubin 2009]. Note that for a given light source-camera configuration, the calibration needs to be performed only one-time. The 3D co-ordinates of the point S are then given as:

$$S = O_c + d\vec{p_c}\,. \tag{2}$$

3D Scanning The Entire Scene. The above procedure recovers the 3D location of one scene point. In order to measure the shape of the entire scene, the angle $\alpha(t)$ between the emitted light beam and the baseline is changed over time so that the entire scene is illuminated sequentially, one point at a time. For each light beam orientation, the camera captures an image, creating an image stack $I\,(x_c, y_c, t)$. This is shown in Figure 3 (a-b). Figure 3 (c) shows an example temporal intensity profile $I\,(x_c, y_c, t)$.

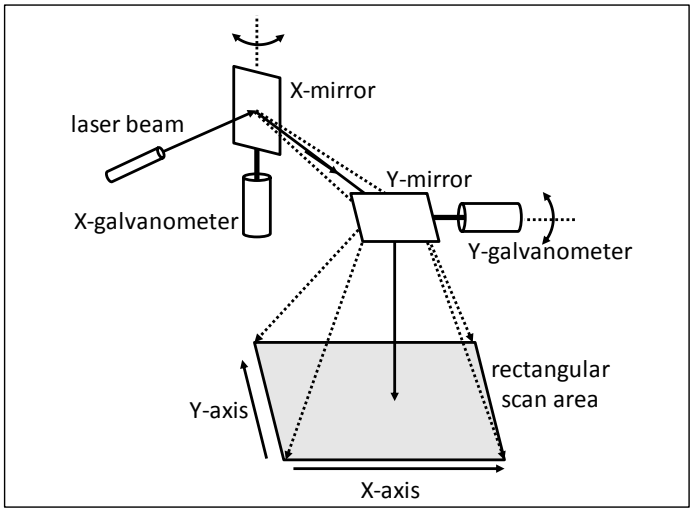


Fig. 4. **Point scanning: Hardware implementation.** A point scanning system is typically implemented with a laser light source which emits a collimated laser beam. The beam is then reflected twice, once each by two mirrors. The mirrors are attached to galvanometers which rotate along orthogonal (X and Y) axes in order to deflect the beam along different directions.

For a given camera pixel (x_c, y_c), let $\hat{t}$ be the time instant when the scanning light beam illuminates the corresponding scene point S. $\hat{t}$ can be estimated by identifying the peak in the pixel's temporal intensity profile:

$$\hat{t} = \arg\max_{t} I\left(x_c, y_c, t\right) \ . \tag{3}$$

Then, using Eq. 4, the depth of the scene point imaged at pixel (x_c, y_c) is given as:

$$d\left(x_c, y_c\right) = \frac{b \tan\left(\alpha(\hat{t})\right) \tan\left(\beta\left(x_c, y_c\right)\right)}{\sin\left(\beta\left(x_c, y_c\right)\right)\left(\tan\left(\alpha(\hat{t})\right) + \tan\left(\beta\left(x_c, y_c\right)\right)\right)} \ , \tag{4}$$

where $\alpha(\hat{t})$ is the angle made by the light beam with the baseline at time $\hat{t}$ and $\beta\left(x_c, y_c\right)$ is the angle between the baseline and the ray joining camera center and pixel (x_c, y_c). Note that both the angles $\alpha(\hat{t})$ and $\beta\left(x_c, y_c\right)$ are known. This is because the light beam's trajectory is assumed to be known a priori and the light source and camera are calibrated.

Hardware Implementation. A point scanning system is typically implemented with a laser light source which emits a collimated laser beam. The beam is then reflected twice, once each by two mirrors. The mirrors are attached to galvanometers which rotate along orthogonal (X and Y) axes

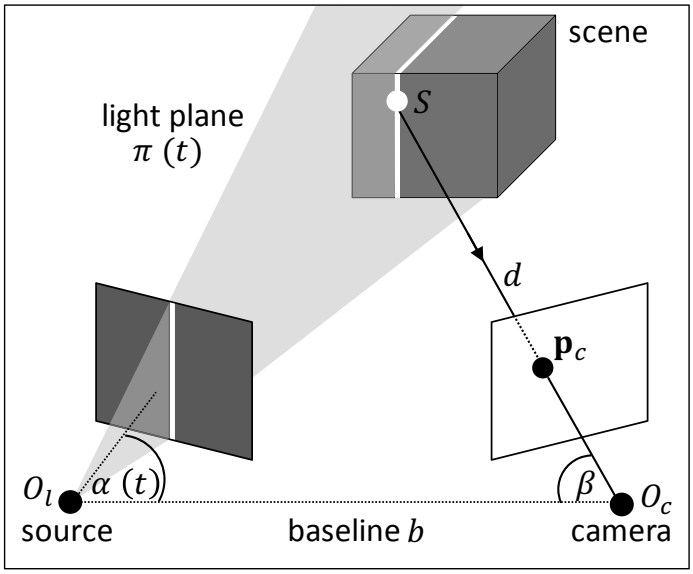


Fig. 5. **Stripe scanning.** The light source emits a sheet of light. The intersection of the sheet and the object forms a 1D curve, called a stripe. Consider a scene point S that lies on the illuminated stripe, and gets imaged at camera pixel (x_c, y_c). The 3D location of S can be determined by finding the intersection of the the light sheet (known) with the camera ray joining the pixel (x_c, y_c) with the camera's optical center.

in order to deflect the beam along different directions. This is illustrated in Figure 4. The amount of scene area covered is determined by the range of deflection beam angles. Point scanning can also be implemented with a digital projector where a single projector pixel is switched on at a time.

Number Of Images. Since point scanning technique recovers the depth of one scene point in one captured image, the total number of captured images is equal to the number of required point depth samples. Another way to count the number of images is to model the light source as a projector, as shown in Figure 1 (a). Let the projector image plane have M rows and N columns, and thus, a total of $M \times N$ pixels. The projected patterns are such that in each pattern, a single pixel is switched on, and the remaining pixels are off. Since one pattern is projected (and one image captured) for every projector pixel, the total number of images captured is $M \times N$. For instance, for a projector with a resolution of 640×480 pixels, point scanning technique would require capturing more than $300,000$ images. If the camera captures images at 30 frames-per-second, the total capture time would be more than 3 hours.

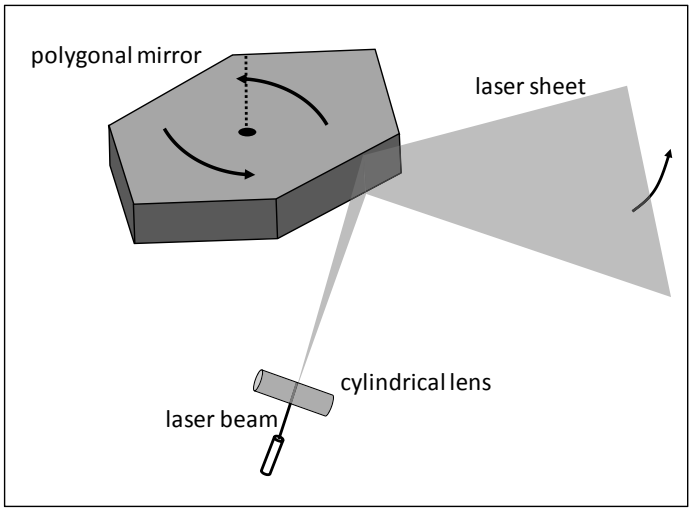


Fig. 6. **Stripe scanning: Hardware implementation.** Stripe scanners are typically implemented by spreading out a laser beam along one dimension into a laser sheet by using a cylindrical lens. The light sheet is then rotated along an axis (within the plane of the light sheet) by using a polygonal mirror.

3. STRIPE SCANNING

In stripe scanning [Shirai and Suwa 1971; Agin and Binford 1976; Curless and Levoy 1995], the light source emits a plane or a sheet of light. The intersection of the sheet and the object forms a 1D curve, called a stripe. This is illustrated in Figure 5. Intuitively, the shape of the stripe in the image captured by the camera is an indicator of the 3D shape of the scene. More formally, let $\pi = [\pi_x \pi_y \pi_z]$ be the representation of the plane of emitted light, so that $\pi_x x + \pi_y y + \pi_z z + 1 = 0$. As in point scanning, π is assumed to be known a priori from the knowledge of the projector's internal calibration parameters. Consider a scene point S that lies on the illuminated stripe, as shown in Figure 5. Let S be imaged at camera pixel (x_c, y_c), and $\vec{p_c}$ be the unit vector along the camera ray from O_c to pixel (x_c, y_c). The depth d of S is then determined by finding the intersection of the the plane π with the camera ray:

$$d = \frac{-\pi.O_c - 1}{\pi.\vec{p_c}}\,, \tag{5}$$

where $\pi.O_c$ and $\pi.\vec{p_c}$ are dot-products.

3D Scanning The Entire Scene. The above procedure recovers the 3D locations of scene points lying on a single stripe. In order to scan the entire scene, the light sheet is rotated (with respect to

the baseline) so that the entire scene is illuminated sequentially, one stripe at a time. Let $\alpha(t)$ be the time-varying angle between the baseline and the normal to the light sheet. For each $\alpha(t)$, the camera captures an image, creating an image stack. Similar to point scanning, for each camera pixel, the time instant when the corresponding scene point is illuminated is estimated by identifying the peak intensity in the pixel's temporal intensity profile.

Hardware Implementation. Stripe scanners are typically implemented by first spreading out a laser beam along one dimension into a laser sheet. This can be achieved with either a cylindrical lens, or a diffraction grating. The light sheet is then rotated along an axis (within the plane of the light sheet) by using either a mirror attached to a galvanometer, or by using a polygonal mirror, as shown in Figure 6. The latter implementation based on a polygonal mirror is one of the most popular in commercial scanners. Stripe scanning can also be implemented with a projector where one column of pixels is switched on at a time. Stripe scanning systems are often used as hand-held devices in several industrial settings, and have also been used for scanning large scale cultural artifacts [Levoy et al. 2000].

Number Of Images. Stripe scanning recovers the depth of one stripe of scene points in one captured image. The number of images is determined by the desired angular resolution [3] and the total scan angle of the laser sheet. For instance, if the sheet needs to move through an angle of 30° to cover the entire scene, and a resolution of 0.01° is desired, the number of acquired images will be 3000. Another way to count the number of images is to model the light source as a projector. The projected patterns are such that in each pattern, a single column (or row) is switched on, and the remaining pixels are off. If the number of columns is N, the total number of images captured is N. For instance, for a projector with a resolution of 640×480 pixels, stripe scanning technique would require capturing 480 images.

3.1. High Speed Scanning Using Asynchronous Sensors

Although stripe scanning requires capturing significantly fewer images as compared to point scanning, if the sensors have a limited frame rate and bandwidth, the acquisition times can still be prohibitively high.

Asynchronous stripe scanning is a method that can achieve significant speed-ups - up to two orders of magnitude higher than conventional stripe scanning. The main observation is that in

[3] The angular resolution is directly proportional to the desired depth resolution.

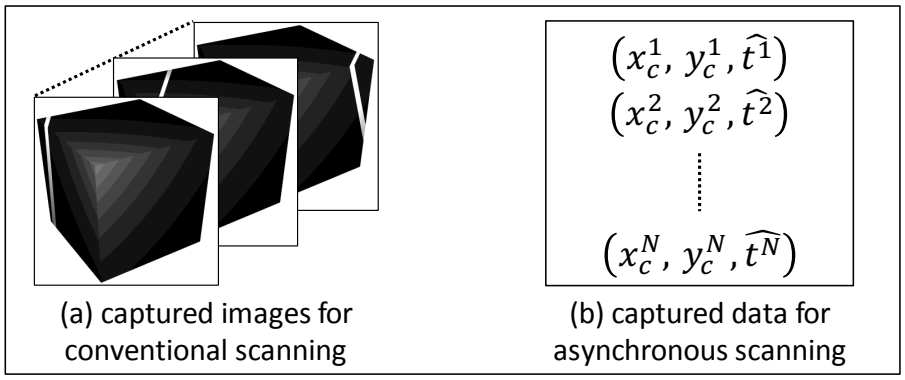


Fig. 7. **High speed scanning using asynchronous sensors.** In asynchronous stripe scanning, a sensor with an array of photo-receptors is used such that each of them acts independently and asynchronously. Each photo-receptor, instead of capturing brightness values continuously, only returns the 3-tuple $[x_c, y_c, \hat{t}]$, where x_c and y_c are the location of the photo-receptor on the sensor image plane, and $\hat{t}$ is the time instant when corresponding scene point is illuminated by the laser sheet. From this information, the 3D location of the scene point is estimated by performing plane-ray intersection.

conventional scanning systems, most of the sensor bandwidth is not utilized. This is because each captured image has only a 1D subset of pixels receiving light. The key idea behind asynchronous scanning is that instead of using sensors with conventional pixel arrays where every pixel captures light synchronously, a sensor with an array of photo-receptors is used such that each of them acts independently and asynchronously. Each photo-receptor (x_c, y_c), instead of capturing brightness values continuously, only returns the 3-tuple $[x_c, y_c, \hat{t}]$, where x_c and y_c are the location of the photo-receptor on the sensor image plane, and $\hat{t}$ is the time instant when the received intensity exceeds a pre-defined threshold. This is illustrated in Figure 7. From this information, the depth value can be estimated by performing plane-ray intersection as in the previous section.

Such sensors can be implemented either as arrays of photodiodes [Araki et al. 1987; Kanade et al. 1991], or by using position sensitive sensors [Oike et al. 2003a; 2003b; 2004] [4]. Since the amount of data captured per pixel is very small (2 location coordinates and a time stamp) and independent of the number of stripes, significant speed-ups (up to 1000 3D frames per second) have been achieved.

[4]Position sensitive detectors are commercially available at low cost. See, for example, http://www.thorlabs.com/newgrouppage9.cfm?objectgroup_id=4400.

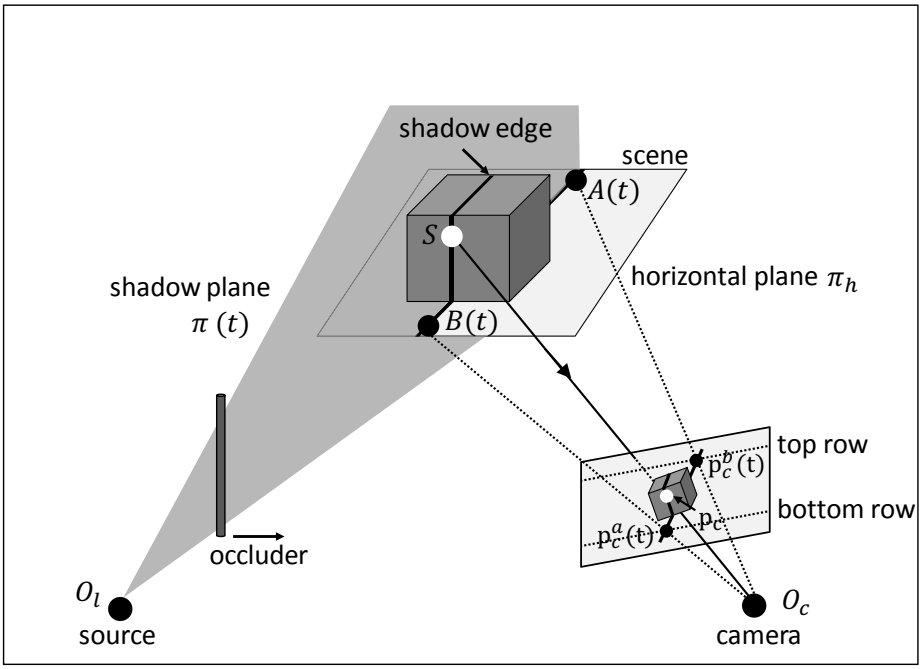


Fig. 8. **Low cost scanning using shadows.** In shadow scanning, the stripe is generated by moving a line occluder (e.g., a thin stick) in front of a point light source. Assuming that the occluder is sufficiently thin, a planar shadow is created which on intersection with the scene creates a dark stripe. Since it does not require precise light sources, shadow scanning is a relatively inexpensive and high accuracy method for 3D scanning.

3.2. Low Cost Scanning Using Shadows

An interesting variation of stripe scanning, called shadow scanning, was proposed in [Bouguet and Perona 1998; 1999]. Shadow scanning does not require a projector or a laser stripe source. Instead, the stripe is generated by moving a line occluder (e.g., a thin stick) in front of a point light source. Assuming that the occluder is sufficiently thin, a planar shadow is created which on intersection with the scene creates a dark stripe (in contrast, conventional stripe scanning creates a bright stripe). This is illustrated in Figure 8. Since it does not require precise light sources, shadow scanning is a relatively inexpensive and high accuracy method for 3D scanning.

Method Details. The camera captures images as the occluder is moved across the scene. Consider a scene point S getting imaged at camera pixel (x_c, y_c). Let $\hat{t}$ be the time instant that the shadow edge passes through (x_c, y_c). $\hat{t}$ can be estimated by finding the location of a negative peak in the temporal intensity profile at (x_c, y_c):

$$\hat{t} = \arg\min_{t} I\left(x_c, y_c, t\right) \ . \tag{6}$$

Let the plane containing the shadow at time $\hat{t}$ be represented by $\pi(\hat{t})$. The 3D coordinates of S can be determined by intersecting the plane $\pi(\hat{t})$ with the ray from the camera center to the pixel, as discussed previously.

Determining the plane $\pi(\hat{t})$. In conventional stripe scanning, the equation of the light plane is determined by calibrating the light source. In shadow scanning, the location of the occluder is not precisely known, especially if the occluder is moved manually. In order to determine the plane $\pi(\hat{t})$, the following procedure is used. Let the object sits on a horizontal plane π_h in 3D space. π_h can be estimated a priori since the camera is calibrated. It is assumed that the shadow stripe is always visible along two rows in the image, as shown in Figure 8. At time instant $\hat{t}$, first, the shadow stripe locations on the two image rows are identified. We call them $\mathbf{p}_c^a(\hat{t})$ and $\mathbf{p}_c^b(\hat{t})$. Then, the corresponding 3D locations on the horizontal plane π_h are estimated by back-projecting the rays from O_c to $\mathbf{p}_c^a(\hat{t})$ and $\mathbf{p}_c^b(\hat{t})$. Let the 3D locations be $A(\hat{t})$ and $B(\hat{t})$. Note that both $A(\hat{t})$ and $B(\hat{t})$ lie on the shadow plane. $\pi(\hat{t})$ is then determined by fitting a plane to points $A(\hat{t})$ and $B(\hat{t})$, and the point light source position O_l (assumed to be known).

4. BINARY CODING

So far, we have considered methods that illuminate only a small fraction of the scene (a single point or a 1D stripe) at a time. These methods need to scan the projected light beam or sheet over the scene, resulting in large acquisition times. Next, we discuss techniques that achieve significantly higher acquisition speeds by illuminating the entire scene simultaneously. These techniques, called structured light methods, emit multiple light sheets simultaneously. As discussed in the previous section, for a given camera pixel, its corresponding light sheet must be estimated in order to compute depth by triangulation. In stripe scanning, the correspondence estimation is relatively straight-forward since only one light sheet is emitted at a time.

However, in structured light methods, since multiple light sheets are emitted simultaneously, the correspondence is computed by assigning a unique temporal code to each sheet so that the sheet's intensity varies temporally according to its code. This is achieved by using a projector (analog or digital) as the light source. Each column on the projector image plane generates a light sheet.

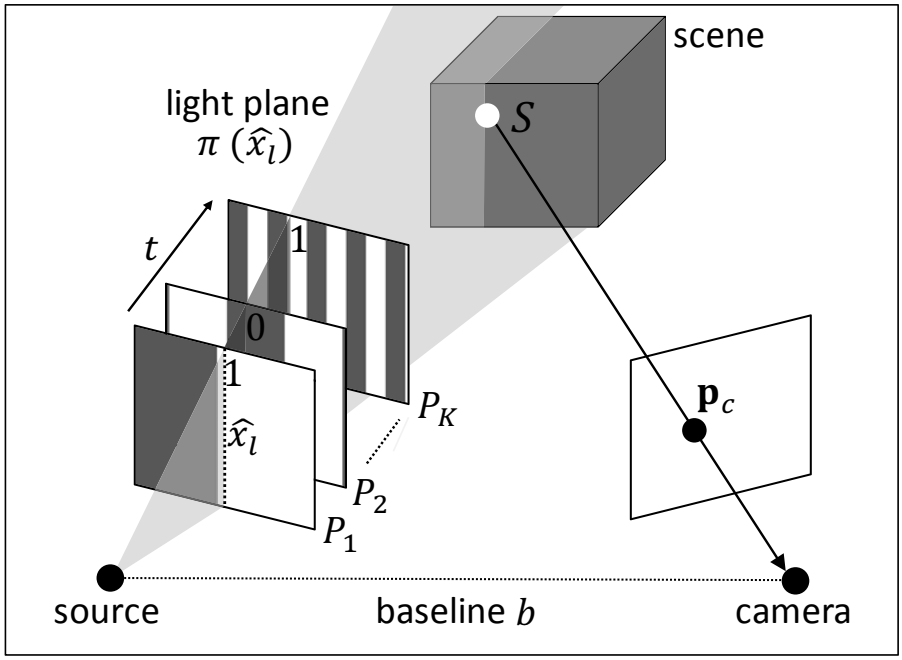


Fig. 9. **Binary coding.** In binary coding, the light source emits multiple light sheets simultaneously. Each sheet is assigned a unique temporal binary code so that the sheet's intensity varies temporally according to its code. This is achieved by using a projector as the light source. Each column on the projector image plane generates a light sheet. By projecting 2D binary coded illumination patterns where each column has a unique temporal binary code, correspondence is established between camera pixels and projector columns (light sheets).

Thus, by projecting 2D coded illumination patterns where each column has a unique temporal intensity code, correspondence is established between camera pixels and projector columns.

The binary coding method [Posdamer and Altschuler 1982] is one of the most popular methods in practice due to its simplicity and ease of implementation. In binary coded structured light, each projector column (and hence, the light sheet that it generates) is assigned a unique binary code. The length of the code is $\lceil \log_2 N \rceil$ bits, where N is the number of projector columns. The projector projects one binary intensity pattern (consisting of 1 and 0 values) for every bit. As a result, each column generates a light sheet whose intensity varies temporally according to its binary code. This is illustrated in Figure 9. The camera captures an image for every projected pattern. This is illustrated in Figure 10.

Let the projected patterns be P_i for $1 \leq i \leq \log_2 N$, and the captured images be I_i for $1 \leq i \leq \log_2 N$. For a pattern P_i, pixels in column number $x_l, 1 \leq x_l \leq N$ are on if $P_i(x_l) = 1$. In this case, the column generates a light sheet. If $P_i(x_l) = 0$, no light is projected.

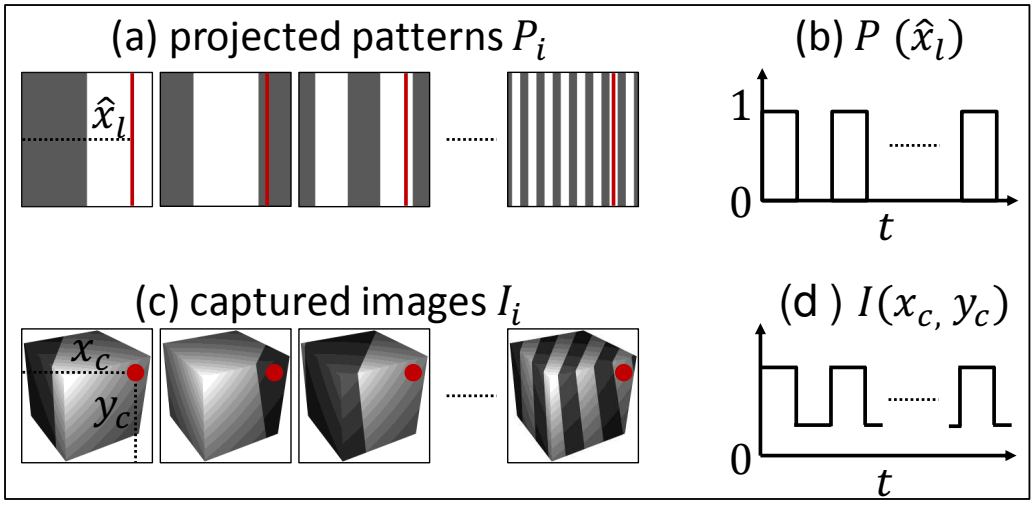


Fig. 10. **Binary coding: Projected patterns and captured images.** (a) The projector projects a series of binary coded patterns on the scene. (b) The set of intensities emitted by a projector column forms its unique binary code. (c) The camera captures an image for every projected pattern. (d) The set of brightness values captured at a camera pixel (after thresholding) are used to compute the corresponding projector column.

Consider a scene point S. Suppose it is illuminated by projector column number $\hat{x}_l$. Let S be imaged at the camera pixel $\mathbf{p}_c$. The image brightness values received at $\mathbf{p}_c$ are given by:

$$I_i(\mathbf{p}_c) = \alpha(\mathbf{p}_c, \hat{x}_l) P_i(\hat{x}_l) + A(\mathbf{p}_c)\,, \tag{7}$$

where $\alpha(\mathbf{p}_c, \hat{x}_l)$ is a proportionality constant. It is the image brightness received at $\mathbf{p}_c$ if the projector column $\hat{x}_l$ emits unit intensity [5]. $A(\mathbf{p}_c)$ is the constant ambient illumination term. It is the image brightness at $\mathbf{p}_c$ due to light sources other than the projector. Note that both $\alpha(\mathbf{p}_c, \hat{x}_l)$ and $A(\mathbf{p}_c)$ are unknown. The goal is to recover the bit sequence $P_i(\mathbf{p}_c)$ from the captured intensities $I_i(\mathbf{p}_c)$, and thus the corresponding projector column. This process is called decoding.

Decoding Process. Decoding can be performed by using a simple thresholding approach. An intensity threshold $\tau(\mathbf{p}_c)$ is established for each camera pixel by capturing two additional images, one by projecting an all bright pattern (all projector columns are on), and the other by projecting an all dark pattern (all columns are off). Let the two images be I_{bright} and I_{dark}:

[5] $\alpha(\mathbf{p}_c, \hat{x}_l)$ encapsulates the scene point's reflectance properties and orientation, camera's gain, light source's brightness, and intensity fall-off (reduction).

$$I_{bright}(\mathbf{p}_c) \;=\; \alpha(\mathbf{p}_c, \hat{x_l}) + A(\mathbf{p}_c) \tag{8}$$

$$I_{dark}(\mathbf{p}_c) \;=\; A(\mathbf{p}_c)\,. \tag{9}$$

The threshold is the average of the bright and dark images, $\tau(\mathbf{p}_c) = \frac{I_{bright}(\mathbf{p}_c)+I_{dark}(\mathbf{p}_c)}{2}$. The projected sequence is estimated by comparing the captured intensities with the threshold:

$$\hat{P}_i(\mathbf{p}_c)) = \begin{cases} 1, & \text{if } I_i(\mathbf{p}_c) > \tau(\mathbf{p}_c) \\ 0, & \text{otherwise} \end{cases}, \tag{10}$$

where $\hat{P}_i(\mathbf{p}_c))$ is the estimated projected values. This decoding method requires capturing two additional images. Another, more robust, decoding method is to project an additional inverse pattern $1 - P_i$ for every original pattern, and capturing an image. The captured image $\bar{I}_i$ for the inverse pattern is given by:

$$\bar{I}_i(\mathbf{p}_c) = \alpha(\mathbf{p}_c, \hat{x_l})\,(1 - P_i(\hat{x_l})) + A(\mathbf{p}_c)\,. \tag{11}$$

Decoding is then performed by comparing the original images and the inverse images:

$$\hat{P}_i(\mathbf{p}_c)) = \begin{cases} 1, & \text{if } I_i(\mathbf{p}_c) > \bar{I}_i(\mathbf{p}_c) \\ 0, & \text{otherwise} \end{cases}. \tag{12}$$

Hardware Implementation And Acquisition Speed. Binary coding can be implemented by a digital camera and a digital projector. The projector and camera need to be calibrated for their external and internal parameters. For a tutorial on projector-camera calibration, the reader is referred to [Lanman and Taubin 2009]. Binary coding method requires capturing approximately $10 - 20$ images for capturing a single 3D scan, depending on the projector resolution. Typical digital projectors can project patterns at the rate of $60 - 120$ frame per second. Thus, with binary coding, approximately $5 - 10$ 3D frames can be captured per second. It is possible to achieve significantly higher speeds (up to 500 frames per second) by using high speed DLP projectors which can project binary patterns at the rate of more than $10,000$ frames per second [Narasimhan et al. 2008; Koppal et al. 2012].

4.1. Gray Coding

Given a set of N projector columns, there are several possible binary coding schemes which assign unique binary codes of length $\lceil \log_2 N \rceil$ to different columns. In practice, the most widely used binary coding method is the Gray coding scheme [Inokuchi et al. 1984; Sato and Inokuchi 1985]. Gray codes are binary codes such that the codes for adjacent columns are different in only one bit, i.e., the Hamming distance between codes for adjacent columns is 1. In contrast, in conventional binary codes, the Hamming distance between adjacent codes may be up to $\log_2(N)$. Gray codes, by minimizing the Hamming distance between adjacent codes, achieve higher robustness during decoding, especially if there is a significant mismatch in the projector and camera sensor resolution, or in the presence of common imaging degradations such as defocus blur [Gupta et al. 2013].

5. K-ARY AND COLOR CODING

K-ary coding techniques are generalizations of binary coding methods where the projected patterns use K different intensity levels, instead of only 2 as in binary methods. Suppose the projector has N columns. In order to assign a unique code to each column, a K-ary coding scheme will require a code of $\lceil \log_K N \rceil$ symbols. Each symbol can take one of K different values. For instance, if a projector with $K = 8$ intensity levels and $N = 512$ columns is used, a code of length 3 would be sufficient to assign a unique code to every projector column. Similar to binary coding, one image is projected and captured for every symbol in the code. In the above mentioned example, since the code length is 3, the number of images that need to be projected and captured is 3.

Decoding and Speed-Robustness Tradeoff. In K-ary coding, the decoding process involves estimating the intensity level (one out of K) emitted from the projector column corresponding to each camera pixel. Let the intensity levels emitted by the projector be uniformly distributed in the normalized range $[0, 1]$, i.e., $\left[0, \frac{1}{K-1}, \frac{2}{K-1}, \ldots, K\right]$. As K increases, the gap between consecutive intensity levels decreases, and the sensor needs to distinguish between closely spaced projected intensity levels for correct decoding. As a result, due to noise and limited dynamic range of the sensor, the decoding process becomes increasingly error prone. On the other hand, for a fixed number N of projector columns, as K increases, the required number of images, $\lceil \log_K N \rceil$, decreases. This presents a trade-off between robustness and speed. Larger values of K result in high speed, whereas a small K results in high robustness, albeit at the cost of a large acquisition time.

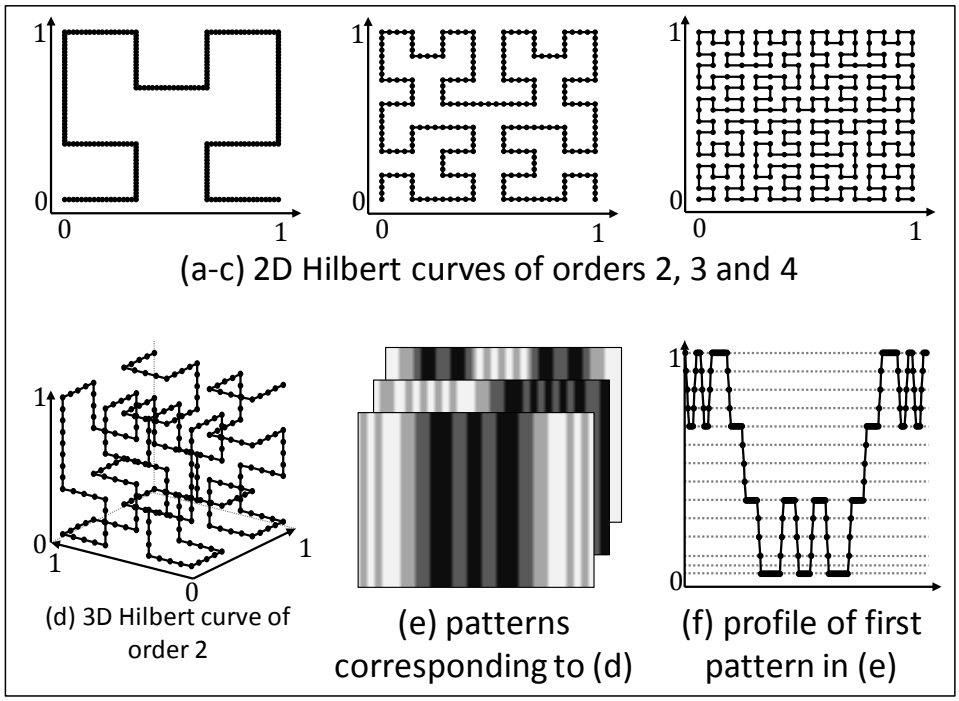


Fig. 11. **K-ary coding.**

The optimal K depends on scene characteristics (scene brightness values), sensor characteristics (noise and dynamic range) and the amount of ambient illumination. For low brightness scenes and low quality (strong noise and low dynamic range) sensors, a lower value of K is required to ensure correct decoding, albeit at the cost of a large acquisition time. On the other hand, a larger value of K can be chosen for high albedo scenes, brighter light sources and high quality sensors. In practice, the optimal K can be estimated in a scene-adaptive manner by measuring scene and sensor characteristics as a pre-processing step (by capturing a few extra images), and then designing the optimal codes according to the characteristics [Caspi et al. 1998].

Design Of Projected Patterns. Once K, the number of intensity levels, is decided, the next question is how to design the codes or the projected patterns, i.e., how to assign unique intensity codes to projector columns. Let $L = \lceil \log_K N \rceil$ be the number of projected patterns. The code for each of the N columns can be represented as an L-dimensional vector. The code design problem can be thought of as placing N points in the L-dimensional space. Ideally, in order to achieve the maximum robustness against image noise, the points should be placed so that the inter-point distance is maximized. Horn *et al.* [Horn and Kiryati 1997] showed that while finding the optimal solution to this

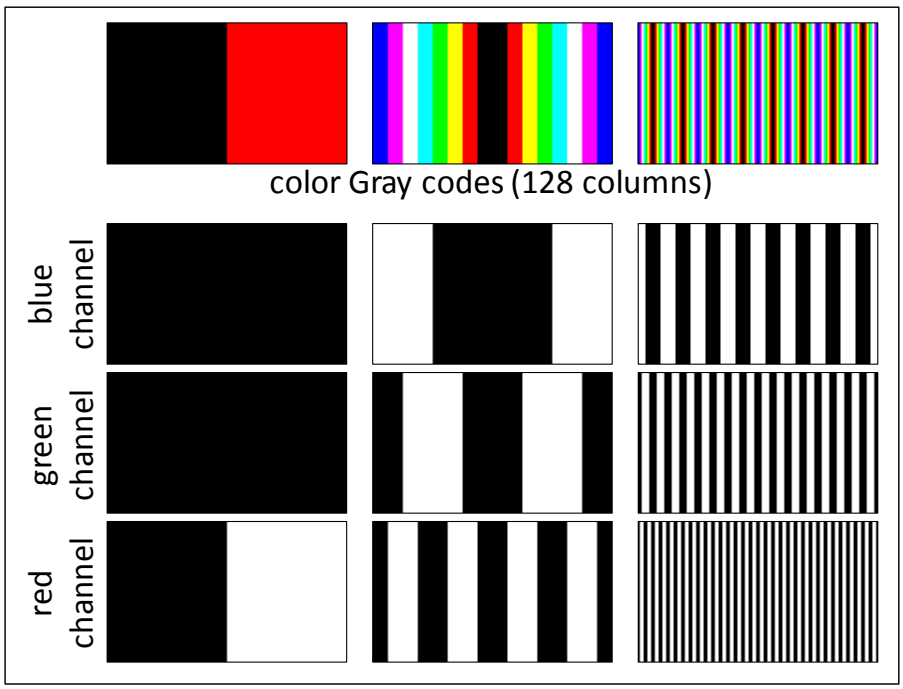


Fig. 12. **Color coding.** It is possible to lower the acquisition time by embedding several monochromatic projected patterns into a smaller number of color patterns. (Top) An example set of color coded projected patterns, using three different color channels. (Bottom) Corresponding monochromatic patterns.

problem is intractable, placing the points along L-dimensional space filling Hilbert curves produces high quality results. Example 2D and 3D Hilbert curves are shown in Figure 11 (a-d). Projected patterns corresponding to the curve in Figure 11 (d) are shown in Figure 11 (e).

Color Coding. Most digital projectors are designed to emit light in three distinct color channels (red, green and blue). By using color variations in addition to intensity variations, it is possible to achieve additional flexibility in code design. Boyer and Kak [1987] proposed a coding scheme which required projecting and capturing only a *single* color image, thus making it suitable for capturing dynamic scenes. However, this method assumed that the reflectance of the scene is neutral or grey.

Caspi *et al.* [Caspi et al. 1998] provided a more general framework for designing color coded patterns. In general, if K_r, K_g and K_b intensity levels are used for the red, green, and blue channel, respectively, the total number of available intensity levels is $K = K_r \times K_g \times K_b$. They provided a method to choose the number of levels K_r, K_g and K_b according to scene and sensor characteristics. Figure 12 shows an example set of projected patterns for $K_r = K_g = K_b = 2$. The projector has $N = 128$ columns, thus requiring only $\lceil \log_8 128 \rceil = 3$ images.

Radiometric Calibration Of The Projector. While binary coding methods require the projector to emit only two light levels (0 and 1), K-ary and color coding methods require emitting several different intensities and colors. For instance, in a K-ary coding scheme, the K levels could be uniformly distributed values between 0 and 1: $[0, \frac{1}{K-1}, \frac{2}{K-1}, \ldots, 1]$. In practice, if a digital projector is instructed to project a brightness value v, it emits a transformed value $R(v)$. The function $R(.)$ is called the radiometric response of the projector, and is often non-linear. In order to compensate for the non-linear response, the radiometric response of the projector should be measured, and the inverse response R^{-1} should be applied to the patterns before projecting. The process of measuring and compensating for the radiometric response is called projector radiometric calibration. The reader is referred to [Nayar et al. 2003; Grossberg et al. 2004] for more details on projector radiometric compensation.

6. CODING USING CONTINUOUS FUNCTIONS

Binary and K-ary coding belong to the class of *discrete* coding techniques. These methods assume that the light source has a finite spatial resolution (e.g., digital projectors), and thus can emit a discrete set of intensity coded light planes. The depth resolution achieved by discrete coding is limited by the number of emitted light planes. This is because during depth recovery from triangulation, the number of possible 3D locations (and hence, depth values) for a scene point is equal to the number of light planes.

Continuous coding techniques are applicable to scenarios where the light source can emit a continuous set of individually coded light planes (e.g., analog slide projectors or digital projectors with defocus). In continuous coding schemes, the intensity profile for each projected pattern is a continuous function, for example, a sinusoid. Continuous coding methods can theoretically achieve infinite depth resolution because there are an infinite number of light planes. However, the finite resolution of the camera, finite numerical precision, and image noise place practical limits on the achievable resolution. In general, for a given sensor, continuous techniques are capable of achieving higher depth resolution as compared to discrete methods.

6.1. Intensity Ratio Method

The first example of a continuous coding method is the intensity ratio method proposed by [Carrihill and Hummel 1985]. This method involves projecting two images on the scene. The first pattern

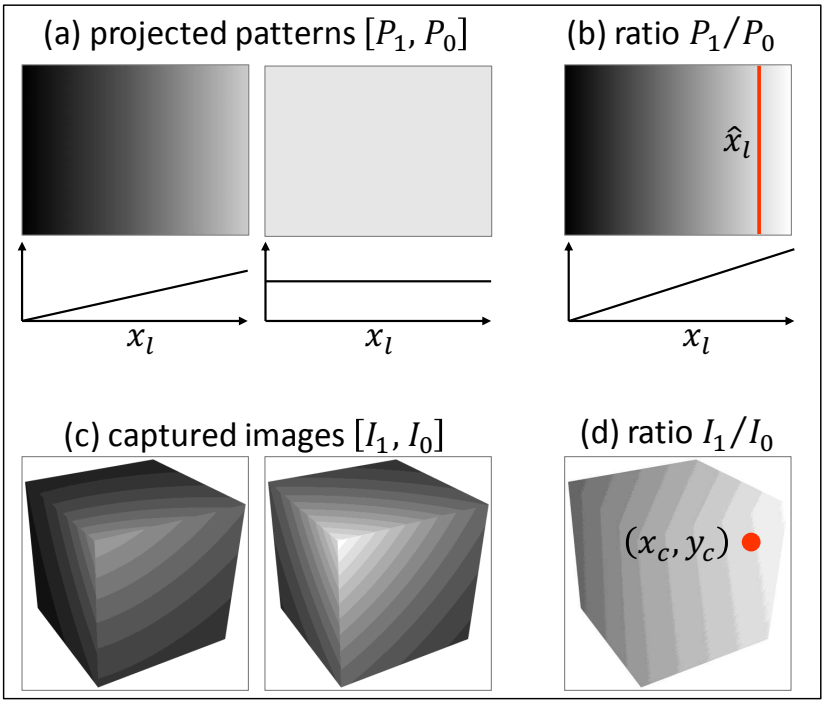


Fig. 13. **Intensity ratio method.** (a) Two patterns are projected on the scene. First is a constant brightness pattern. Second is a 'ramp' pattern with brightness increasing linearly from 0 to 1 across columns. (b) Pixel-wise ratio of the projected patterns. (c) Example captured images. (d) Pixel-wise ratio of the two captured images.While the captured images show intensity variations due to surface shading, the ratio image does not have intensity variations due to surface shading. For each camera pixel, the ratio image depends only on the corresponding projector column value.

P_0 is a constant brightness pattern, $P_0(x_l) = 1$. The second pattern P_1 is a 'ramp' function with projected brightness increasing linearly from 0 to 1 across columns:

$$P_1(x_l) = (x_l - 1) \times \Psi \,, \tag{13}$$

where $\Psi = \frac{1}{N-1}$ is the slope of the ramp and N is the number of total projector columns. This is shown in Figure 13. The two patterns are projected sequentially, and two images are captured, one for each pattern. Similar to Eq. 7, the two captured intensities I_0 and I_1 at a camera pixel $\mathbf{p}_c$ are given as:

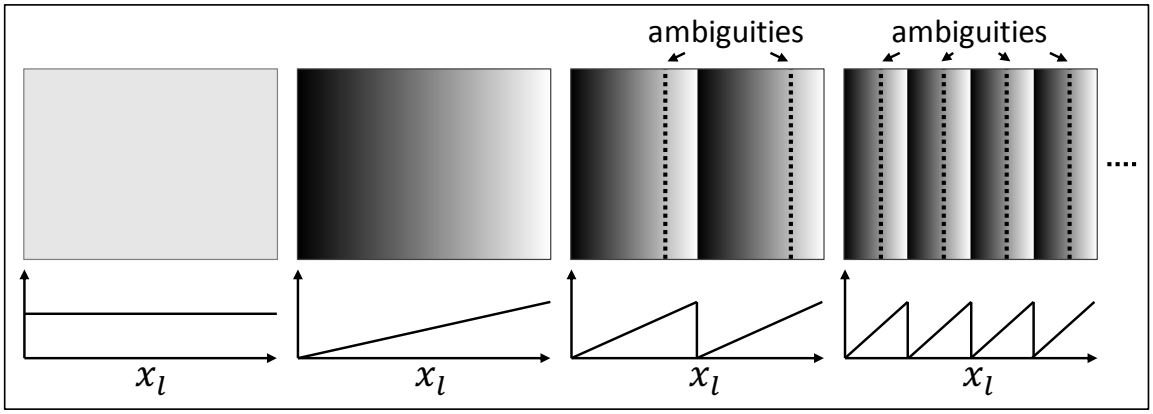


Fig. 14. **Sawtooth coding.** In addition to the constant pattern and the single ramp pattern, sawtooth coding uses patterns with progressively increasing number of ramps. The horizontal intensity profiles of the projected patterns have a saw-tooth like appearance.

$$I_0(\mathbf{p}_c) \;=\; \alpha(\mathbf{p}_c, \hat{x_l}) + A(\mathbf{p}_c) \tag{14}$$

$$I_1(\mathbf{p}_c) \;=\; \alpha(\mathbf{p}_c, \hat{x_l})(\hat{x_l} - 1)\Psi + A(\mathbf{p}_c)\,, \tag{15}$$

where $\hat{x}_l$ is the projector column corresponding to camera pixel $\mathbf{p}_c$. $\alpha(\mathbf{p}_c, \hat{x_l})$ is the scene-dependent proportionality constant, and A is the amount of ambient illumination. If the ambient illumination A is assumed to be zero, the scene dependent constant $\alpha(\mathbf{p}_c, \hat{x_l})$ can be eliminated by computing the ratio image I_{ratio}:

$$I_{ratio}(\mathbf{p}_c) \;=\; \frac{I_1(\mathbf{p}_c)}{I_0(\mathbf{p}_c)} = (x_l - 1)\Psi\,. \tag{16}$$

A pair of example captured images and the corresponding ratio image are shown in Figure 13. Note that while the captured images show intensity variations due to surface shading, the ratio image does not have intensity variations due to surface shading. The ratio image depends only on the corresponding column $\hat{x}_l$, which can be recovered as:

$$\hat{x_l} = \frac{I_{ratio}(\mathbf{p}_c)}{\Psi} + 1\,. \tag{17}$$

6.2. Sawtooth Coding

The intensity ratio method as described above is highly sensitive to image noise. Let $\triangle I_{ratio}(\mathbf{p}_c)$ be the error in the ratio image due to noise in the captured images I_1 and I_2. The resulting error $\triangle \hat{x}_l$ in the estimated correspondence is given by:

$$\triangle \hat{x}_l = \frac{\triangle I_{ratio}(\mathbf{p}_c)}{\Psi} . \tag{18}$$

The above equation states that the error in the estimated correspondence is inversely proportional to the slope of the intensity ramp. In conventional intensity ratio scheme, the slope of the ramp is small, thus resulting in large errors. Chazan and Kiryati [Chazan and Kiryati 1995] developed a hierarchical (pyramidal) intensity ratio method. In addition to the constant pattern and the single ramp pattern, they used patterns with progressively increasing number of ramps, as shown in Figure 14. The coding scheme is also called saw-tooth coding due to the saw-tooth like appearance of the profile of the projected patterns. As the number of ramps increase, the slope of the ramps also increase, thereby reducing the error in the estimated correspondence. In particular, for a pattern P_i with n_i ramps, the slope of each ramp is $n_i\Psi$, where Ψ is the slope for the single ramp pattern. Following Eq. 19, the error in the correspondence estimated using P_i is reduced by a factor of n_i:

$$\triangle \hat{x}_l = \frac{\triangle I_{ratio}(\mathbf{p}_c)}{n_i\Psi} . \tag{19}$$

However, if a pattern with multiple ramps is used, there are ambiguities in the estimated correspondence as multiple projector columns have the same intensity, as shown in Figure 14. The ambiguities are resolved in a hierarchical manner. Starting with the pattern with the highest number of ramps (most number of potential correspondences), the list of possible correspondences is pruned by using the approximate, but less ambiguous results obtained for the previous pattern. The process is repeated until a single correspondence is achieved [Chazan and Kiryati 1995].

6.3. Rainbow Range Finder

A method related to the intensity ratio method is the rainbow range finder (RRF) proposed by [Tajima and Iwakawa 1990]. Instead of using an intensity ramp in the projected pattern, the RRF uses a *wavelength ramp*, i.e., each projected light plane has a different wavelength of light. The wavelengths increase linearly across the planes. This is illustrated in Figure 15. Each camera pixel

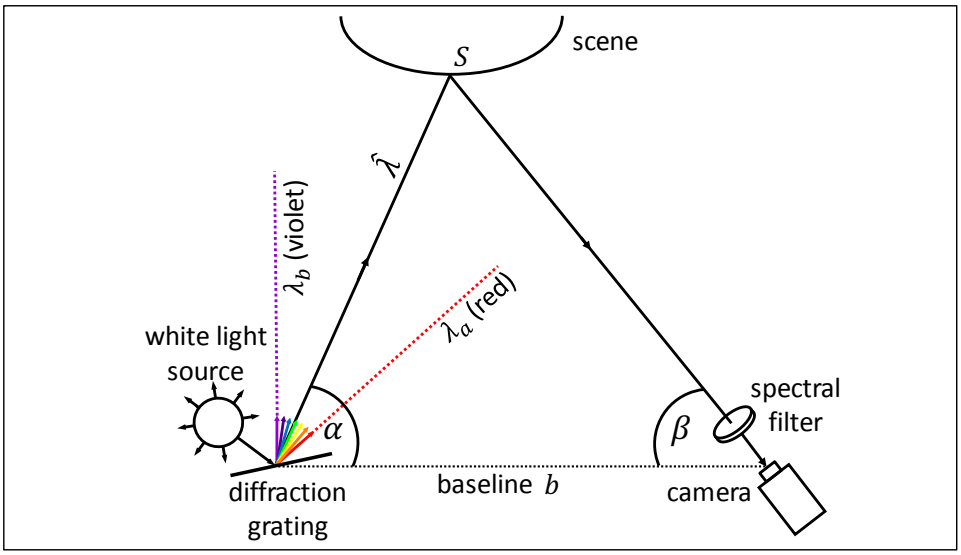


Fig. 15. **Rainbow range finder.** The rainbow range finder uses a wavelength ramp, i.e., each projected light plane has a different wavelength of light. The wavelengths increase linearly across the planes. Each camera pixel receives light of a single wavelength. The received wavelength is the same as that of the light plane illuminating the corresponding scene point. Correspondence can be computed by estimating the wavelength of the received light. This is achieved by capturing two images with different spectral filters in front of the sensor, and taking their ratio.

receives light of a single wavelength [6]. The received wavelength is the same as that of the light plane illuminating the corresponding scene point. Thus, correspondence can be computed by estimating the wavelength of the received light. This is achieved by capturing two images with different spectral filters in front of the sensor, and taking their ratio. For details, the reader is referred to [Tajima and Iwakawa 1990].

The advantage of the RRF method is that it can potentially be used for computing depth in a single image. This can be achieved by using an array of spectral filters placed on top of the sensor array, similar to the Bayer pattern used in most consumer sensors. The individual spectral measurements can be extracted and demosaiced in a way similar to the demosaicing step performed for computing color image from a Bayer pattern image. The disadvantage of the RRF sensor is that it is only applicable on scenes with relatively neutral (grey) reflectance properties. Since scene points are illuminated with only a single wavelength of light, they will reflect light towards the sensor only if their reflectance spectrum contains the incident wavelength.

[6] In practice, due to a finite pixel size, the received light has a narrow range of wavelengths

7. PHASE SHIFTING USING SINUSOIDS

Phase-shifting is a continuous coding method where the intensity of the projected patterns varies sinusoidally across columns [Srinivasan et al. 1985]. It is one of the most popular active triangulation based shape recovery techniques, and is widely used in several commercial systems due to its high speed and high accuracy.

Method Details. Consider a projection pattern $P(x_l)$ whose brightness varies sinusoidally across columns:

$$P(x_l) = o_p + a_p \cos(\phi_p) \ , \tag{20}$$

where $\phi_p = \frac{2\pi x_l}{N}$ is the phase of the sinusoid that encodes the column number x_l. N is the total number of columns of the projector. o_p and a_p are the offset and the amplitude of the sinusoid, respectively. o_p and a_p are chosen so that the projected pattern is non-negative. Typical values are $o_p = a_p = 0.5$.

Let $I(\mathbf{p}_c)$ be the image intensity captured at camera pixel $\mathbf{p}_c$ when P is projected. If the projector column corresponding to $\mathbf{p}_c$ is $\hat{x}_l$, $I(\mathbf{p}_c)$ is given as:

$$I(\mathbf{p}_c) = \alpha(\mathbf{p}_c, \hat{x_l}) P(\hat{x_l}) + A(\mathbf{p}_c) \, , \tag{21}$$

where $\alpha(\mathbf{p}_c, \hat{x_l})$ is a scene-dependent proportionality constant and $A(\mathbf{p}_c)$ is the constant ambient illumination term as defined previously after Eq. 7. Substituting the value of $P(x_l)$ from Eq. 34 into Eq. 21 and simplifying, we get:

$$I(\mathbf{p}_c) = o_c(\mathbf{p}_c) + a_c(\mathbf{p}_c) \cos\left(\hat{\phi}_p\right) \ , \tag{22}$$

where $o_c = \alpha(\mathbf{p}_c, \hat{x_l}) o_p + A(\mathbf{p}_c)$ and $a_c = \alpha(\mathbf{p}_c, \hat{x_l}) a_p$ are unknown parameters that depend on the scene and sensor characteristics, and the ambient illumination. The correspondence information $\hat{x}_l$ is encoded in the phase term $\hat{\phi}_p = \frac{2\pi \hat{x}_l}{N}$.

Note that in the above equation, the expression for intensity $I(\mathbf{p}_c)$ is a sinusoid with three unknowns - the offset $o_c(\mathbf{p}_c)$, the amplitude $a_c(\mathbf{p}_c)$, and the phase $\hat{\phi}_p$. Since there are three unknowns, phase-shifting requires capturing three images $I_i(\mathbf{p}_c)$, $i = [1, 2, 3]$, one each for projected patterns $P_i(x_l)$, $i = [1, 2, 3]$. The patterns $P_i(x_l)$, $i = [1, 2, 3]$ have different (typically evenly spaced) phases:

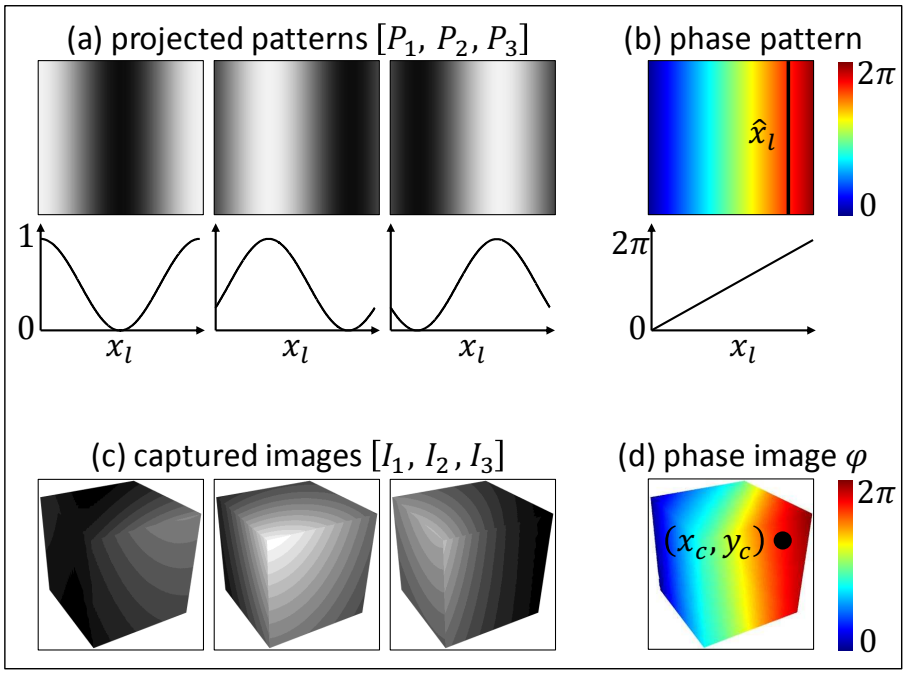


Fig. 16. **Phase shifting.** (a) In phase shifting, the intensity of the projected patterns varies sinusoidally across columns. Three patterns are projected, where each pattern is a *horizontally shifted* version of the other patterns. (b) A projector column is encoded by the sinusoid's phase at that column. (c) Three captured images for an example object. (d) The intensities captured at a camera pixel are used to compute the phase of its corresponding column.

$$P_i(x_l) = o_p + a_p \cos\left(\phi_p + \frac{2\pi(i-1)}{3}\right), \quad 1 \leq i \leq 3. \tag{23}$$

The patterns and the corresponding intensity profiles are shown in Figure 16. Since each projected pattern is a *horizontally shifted* version of the other patterns, the method is called phase shifting. The captured images are given as:

$$I_i(\mathbf{p}_c) = o_c(\mathbf{p}_c) + a_c(\mathbf{p}_c) \cos\left(\hat{\phi}_p + \frac{2\pi(i-1)}{3}\right), \quad 1 \leq i \leq 3. \tag{24}$$

Example captured images are shown in Figure 16. The above equations can be written compactly as a linear system of three equations:

$$\mathbf{I} = \mathbf{M}\mathbf{X}, \tag{25}$$

where

$$\mathbf{I} = \begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix}, \quad \mathbf{M} = \begin{bmatrix} 1 & \cos(0) & -\sin(0) \\ 1 & \cos\left(\frac{2\pi}{3}\right) & -\sin\left(\frac{2\pi}{3}\right) \\ 1 & \cos\left(\frac{4\pi}{3}\right) & -\sin\left(\frac{4\pi}{3}\right) \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} o_c \\ a_c \cos\left(\hat{\phi}_p\right) \\ a_c \sin\left(\hat{\phi}_p\right) \end{bmatrix}. \tag{26}$$

For brevity, we have dropped the argument $\mathbf{p}_c$. $\mathbf{I}$ is the 3×1 vector of the measured intensities at pixel $\mathbf{p}_c$. $\mathbf{M}$ is the 3×3 measurement matrix. $\mathbf{X}$ is the 3×1 unknown vector. $\mathbf{X}$ can be estimated by simple linear inversion: $\mathbf{X} = \mathbf{M}^{-1}\mathbf{I}$. Note that this linear inversion is performed for each camera pixel individually. Once $\mathbf{X}$ is recovered, the phase $\hat{\phi}_p$ is computed as:

$$\hat{\phi}_p = \arccos\left(\frac{\mathbf{X}(2)}{\sqrt{\mathbf{X}(2)^2 + \mathbf{X}(3)^2}}\right), \tag{27}$$

where $\mathbf{X}(j)$ is the j^{th} ($j = [1, 2, 3]$) element of the estimated vector $\mathbf{X}$. $\arccos(.)$ is the inverse cosine function.

Resolving The Ambiguity Due To The $\arccos(.)$ **Function.** The function $\arccos(.)$ returns a phase value $\hat{\phi}_l$ in the range $[0, \pi]$. The true phase value lies in the range $[0, 2\pi]$, and thus, could be either $\hat{\phi}_l$ or $2\pi - \hat{\phi}_l$. This is because $\cos(\theta) = \cos(2\pi - \theta)$. We resolve this two-way ambiguity by computing $\hat{\phi}_l$ as follows:

$$\hat{\phi}_p = \begin{cases} \arccos\left(\frac{\mathbf{X}(2)}{\sqrt{\mathbf{X}(2)^2 + \mathbf{X}(3)^2}}\right), & \text{if } \mathbf{X}_3 \geq 0 \\ 2\pi - \arccos\left(\frac{\mathbf{X}(2)}{\sqrt{\mathbf{X}(2)^2 + \mathbf{X}(3)^2}}\right), & \text{otherwise} \end{cases}. \tag{28}$$

An example recovered phase map is given in Figure 16. The correspondence $\hat{x}_l$ can be recovered from the estimated phase as:

$$\hat{x}_l = \hat{\phi}_l \frac{N}{2\pi}. \tag{29}$$

Using More Images For Increasing Accuracy. While three images are theoretically sufficient for estimating the correspondence, more measurements may be taken for increasing robustness in the presence of strong image noise. In general, we can project $L \geq 3$ patterns P_i given as:

$$P_i(x_l) = o_p + a_p \cos\left(\phi_p + \frac{2\pi(i-1)}{L}\right), \quad 1 \leq i \leq L. \tag{30}$$

All the patterns are sinusoids with different, evenly spaced phases. Similar to Eq. 24, the captured images I_i are given as:

$$I_i(\mathbf{p}_c) = o_c(\mathbf{p}_c) + a_c(\mathbf{p}_c)\cos\left(\hat{\phi}_p + \frac{2\pi(i-1)}{L}\right),\ \ 1 \leq i \leq L\,. \tag{31}$$

These equations can also be written as a linear system similar to Eq. 25, where

$$\mathbf{I} = \begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_L \end{bmatrix},\quad \mathbf{M} = \begin{bmatrix} 1 & \cos(0) & -\sin(0) \\ 1 & \cos\left(\frac{2\pi}{L}\right) & -\sin\left(\frac{2\pi}{L}\right) \\ & \vdots & \\ 1 & \cos\left(\frac{2\pi(L-1)}{L}\right) & -\sin\left(\frac{2\pi(L-1)}{L}\right) \end{bmatrix},\quad \mathbf{X} = \begin{bmatrix} o_c \\ a_c\cos\left(\hat{\phi}_p\right) \\ a_c\sin\left(\hat{\phi}_p\right) \end{bmatrix}. \tag{32}$$

The above is an over-determined linear system of equations which can be solved by using linear least squares: $\mathbf{X} = \mathbf{M}^{\dagger}\mathbf{I}$, where † is the pseudo-inverse operator for a matrix.

7.1. High Frequency Phase Shifting

Phase-shifting as described so far uses patterns with sinusoids of unit frequency, i.e., the period of the sinusoid is equal to the width of the projected pattern. In the presence of image noise, this method is prone to large errors in the estimated correspondence values. This is because the error in recovered correspondences in a phase-shifting system is inversely proportional to ω, the spatial frequency of the pattern used [Wang et al. 2010]:

$$\triangle\hat{x}_l \propto \frac{1}{\omega}\,. \tag{33}$$

In practice, in order to achieve higher accuracy, high-frequency phase-shifting patterns are used. These patterns have sinusoids with higher spatial frequency (smaller periods), and thus contain multiple sinusoid periods within each pattern. Similar to unit frequency sinusoids, high frequency phase shifting is performed by projecting three patterns $P_i^{\omega}(x_l),\ 1 \leq i \leq 3$:

$$P_i^{\omega}(x_l) = o_p + a_p\cos\left(\phi_p^{\omega} + \frac{2\pi(i-1)}{3}\right), \tag{34}$$

where ω is the spatial frequency of the sinusoid used, and $\phi_p^{\omega} = \omega\frac{2\pi x_l}{N}$ is the phase of the sinusoid for frequency ω. Example high frequency patterns and for $\omega = 8$ are shown in Figure 17 (a). Corre-

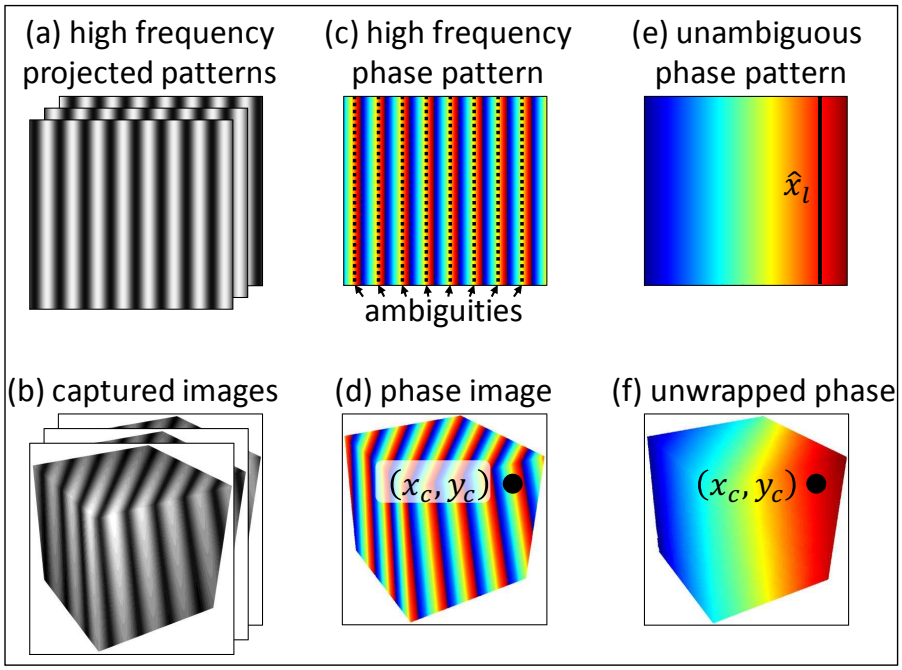


Fig. 17. **High frequency phase shifting.** In order to achieve high signal-to-noise ratio, patterns with high frequency sinusoids are projected. (a) These patterns contain multiple sinusoid periods within each pattern. (b) Example set of captured images for high frequency sinusoidal patterns. (c) Several projector columns have the same phase, resulting in ambiguities in the phase maps computed from the captured images (d). This is called the wrapped phase problem. (e-f) Multiple frequencies can be used for phase unwrapping and recovering unambiguous depth.

sponding camera captured images are shown in Figure 17 (b). The phase at each camera pixel is computed using the same algorithm as discussed above.

Phase Ambiguities And Phase Unwrapping. In high frequency phase shifting, there are multiple repeated sinusoids within the projected images. This creates ambiguities; the phase computed at a camera pixel may correspond to multiple projector columns, as illustrated in Figures 17 (c-d). This is called the *wrapped phase problem*. The process of disambiguation (recovering unambiguous phase) is called phase-unwrapping. Phase-unwrapping is frequently encountered in a variety of disciplines, including interferometry [Gushov and Solodkin 1991; Takeda et al. 1997], radar [Goldstein et al. 1988] and time-of-flight imaging [Jongenelen et al. 2010; Jongenelen et al. 2011]. There are several approaches to phase-shifting, including the path-following approach that assume the

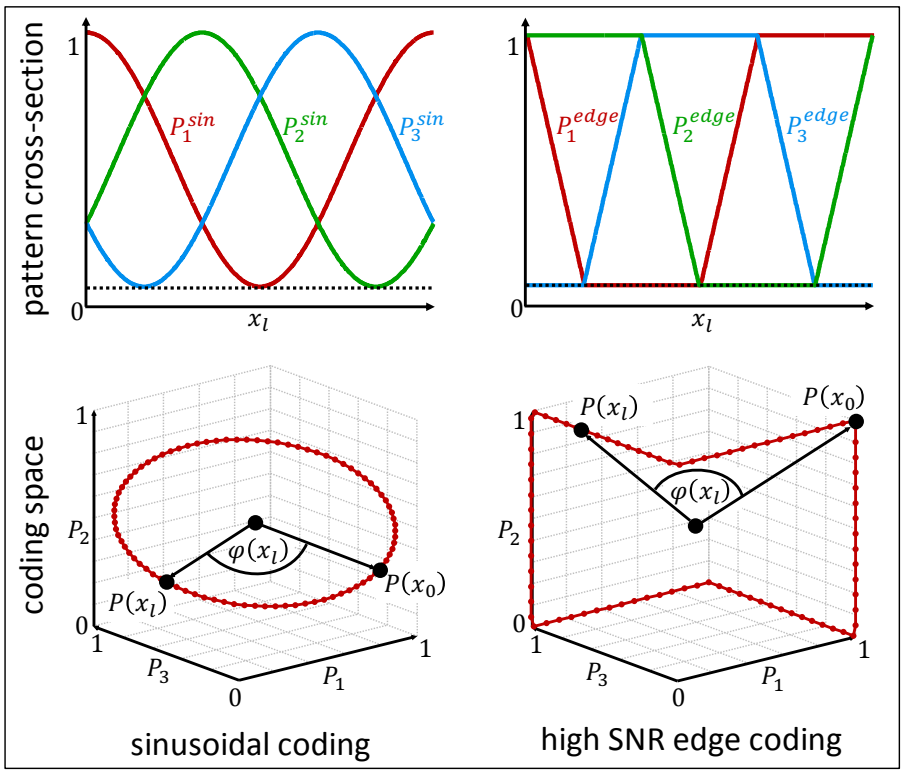


Fig. 18. **High SNR phase shifting**

phase to vary continuously in the captured image [Ghiglia and Pritt 1998]. While this approach is valid for smooth surfaces, it is not easily applicable to scenes with sharp depth discontinuities.

One popular approach is to use multiple-frequencies. For instance, one high frequency and one low (unit) frequency can be used. The phase computed using the higher frequency sinusoid provides high accuracy, but ambiguous, projector correspondence. The low accuracy but unambiguous phase computed using the unit frequency sinusoid is then used to resolve the ambiguities [Towers et al. 2005]. This is similar to the hierarchical approach used in saw-tooth coding (Section 6.2). The unwrapped phase for both the projected pattern and the captured images is shown in Figures 17 (e-f). If F different frequencies are used, the total number of captured images is $3F$. Another similar approach is to use a combination of high-frequency phase-shifting and coarse binary Gray coding patterns [Sansoni and Patrioli 2000].

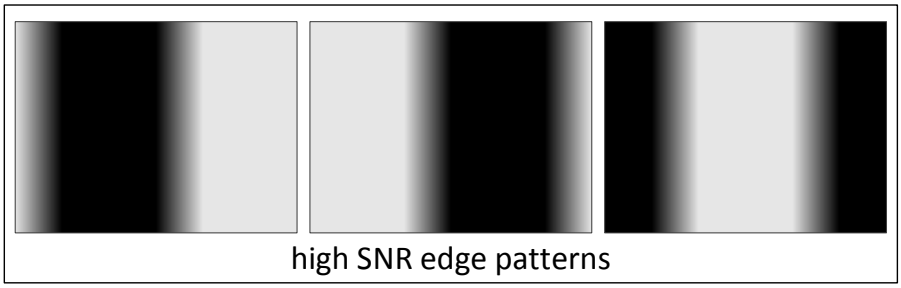


Fig. 19. **High SNR phase shifting: Projected patterns**

7.2. High SNR Phase Shifting

Recall from Section 5 that a structured light method that projects L images can be represented by a set of points, one for each projector column, in the L dimensional coding space. The curve joining these points is called the *coding curve* of the structured light method. It can be shown that the coding curve of the conventional three-image phase-shifting approach is a circle in 3D space [Wang et al. 2010]. Figure 18 (a) shows the coding curve for three-image sinusoidal phase shifting, for a projector with $N = 1024$ columns.

Since the accuracy of a structured light method is proportional to the inter-point distance in the coding space (as also discussed in Section 5), for a given number of points (equal to the number of projector columns), longer curves achieve higher accuracy. [Wang et al. 2010] proposed a novel phase-shifting approach where the coding curve follows the edges of the 3D cube, as shown in Figure 18 (b). The corresponding projected patterns, called the edge patterns, are shown in Figures 19.

The length of the coding curve for the edge patterns is approximately 1.24 times more than the coding curve for conventional phase-shifting. Thus, the edge patterns achieve a signal-to-noise-ratio improvement over conventional phase shifting by a factor of approximately 1.24, with the same number of captured images.

7.3. High Speed Phase Shifting

Phase-shifting requires projecting and capturing a minimum of three images sequentially. For correct depth recovery, it is assumed that the camera and the scene remain static during the time it takes to capture all the images. Thus, phase shifting is unsuitable for recovering shape of dynamic scenes. One method for handling dynamic scenes is to use a color projector and embed the three phase shifting patterns in three color channels of the projected light [Wust and Capson 1991; Huang

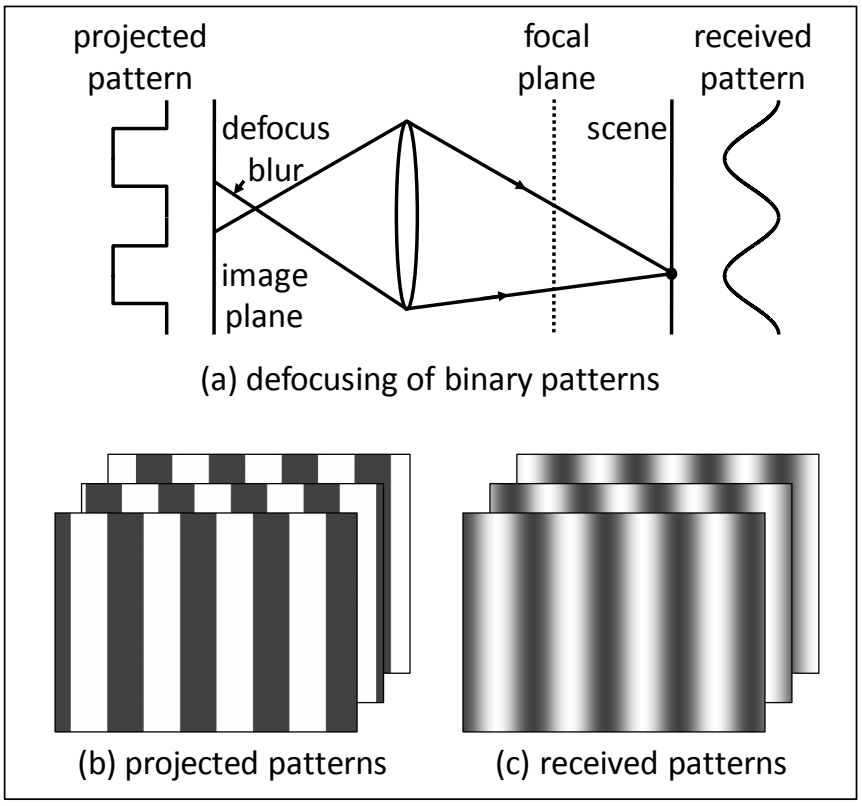


Fig. 20. **High speed phase shifting.** DLP projectors can be used to perform high speed phase shifting. (a) These projectors can project binary patterns at up to $10,000$ frames per second. (a) If a binary pattern with alternating black and white stripes is projected, due to projector defocus, the pattern received at the scene is approximately sinusoidal. The period of the sinusoid is equal to twice the stripe width of the projected binary pattern. (b) The projected patterns. (c) Patterns received at the scene.

et al. 1999; Pan et al. 2006]. By capturing an image with a three-color camera, and separating the color components of the captured image, it is possible to perform phase shifting with a single image. This method, however, assumes that the scene has neutral/grey reflectance, and thus may not be applicable in general.

Another technique for performing high speed phase shifting is to use high-speed DLP projectors [Zhang et al. 2010]. As discussed earlier in Section 4, these projectors can project *binary* patterns at up to $10,000$ frames per second. However, for phase-shifting, sinusoidal patterns need to be projected. How can one generate a continuous range of intensity levels that are required for phase shifting from only binary projected patterns? The key idea is to use defocus of projectors for converting the binary patterns into sinusoidal patterns. While so far we have assumed the pro-

jectors to be a pin-hole device, in practice, projectors have a large aperture and a lens in order to increase the light throughput. While the projected pattern is perfectly focused on the projector focal plane, the pattern gets blurred as one moves away from the focal plane. This is illustrated in Figure 20 (a).

Let P be the projected pattern. If the scene is assumed to be approximately planar, the pattern received by the scene is given by $P_{defocus} = P * B_{defocus}$, where $B_{defocus}$ is the 2D defocus blur kernel. $B_{defocus}$ is a function of the scene depth, the focal plane location and the projector optics. It can be shown that if a binary pattern with alternating black and white stripes is projected, and the focal plane of the projector is placed sufficiently far from the scene, the received pattern is approximately sinusoidal. The period of the sinusoid is equal to twice the stripe width of the projected binary pattern. Example patterns are shown in Figure 20 (b-c). A high speed camera is used for capturing the images. This method can measure more than 500 3D frames per second, thus making it suitable even for dynamic scenes.

8. SINGLE-SHOT CODING

The techniques discussed so far require capturing multiple images in order to establish projector-camera correspondence. The scene and camera are assumed to remain static during image capture. Hence, these techniques are not suitable in scenarios where the scene or camera are moving.

In order to handle dynamic scenes, single-shot coding methods have been developed. These techniques require projecting and capturing only a single image. In single-shot methods, each projector column (or pixel) is encoded with a unique *spatial intensity code*, i.e., the projected pattern is designed so that each of its columns has a unique intensity distribution in its local spatial neighborhood. Intuitively, single-shot coding can be thought of as spatial counterpart to multi-shot techniques (such as binary coding and phase-shifting) where each projector column is encoded with a unique *temporal intensity code.*

There are several coding schemes for designing single-shot patterns. One of the most popular is called pseudo-random coding [Hugli and Maitre 1989]. This approach is based on *De Bruijn sequences*, a sequence often used in combinatorial mathematics. A De Bruijn sequence $S_{DB}^{[k,n]}$ is a sequence of symbols where each symbol can have k different values, and every possible subsequence of n consecutive symbols appears exactly once in the sequence. The length of $S_{DB}^{[k,n]}$ is k^n symbols. A special case is the binary sequence $S_{DB}^{[2,n]}$, called the M-sequence or the maximum length sequence.

Decoding Process. The decoding process in single-shot schemes is based on the following observation: the intensity distribution in a spatial window around a camera pixel is similar to the spatial intensity code of the corresponding projector column. In order to compute the correspondence for a camera pixel, intensities in its local neighborhood are compared against the spatial intensity codes of the projector columns. The column whose code is the best match is returned as the correspondence.

Specifically, suppose a binary projector pattern is designed where the intensities of columns follow a $S_{DB}^{[2,n]}$ sequence. In such a pattern, each projector column has a unique spatial intensity code of size $1 \times n$ (rows $\times$ columns). An example pattern designed with a $S_{DB}^{[2,10]}$ sequence is shown in Figure 21 (a). The corresponding captured image is shown in Figure 21 (b). Consider a scene point S getting illuminated by a projector column $\hat{x}_l$ and imaged at camera pixel $\mathbf{p}_c = (x_c, y_c)$. Let the spatial intensity code around column $\hat{x}_l$ be $PS(\hat{x}_l; n) = \left[P\left(\hat{x}_l - \frac{n}{2}\right), \ldots, P\left(\hat{x}_l + \frac{n}{2} - 1\right)\right]$. Let $I(\mathbf{x}_c, \mathbf{y}_c; n) = \left[I\left(\mathbf{x}_c - \frac{n}{2}, \mathbf{y}_c\right), \ldots, I\left(\mathbf{x}_c + \frac{n}{2} - 1, \mathbf{y}_c\right)\right]$ be the sequence of captured intensities in a 1D spatial neighborhood of the pixel $\mathbf{p}_c$. If the scene's reflectance is assumed to be locally constant, and the captured image is assumed to be rectified [7], $I(\mathbf{x}_c, \mathbf{y}_c; n)$ is approximately a scaled and offset version of the projected subsequence $PS(\hat{x}_l; n)$:

$$I(\mathbf{x}_c, \mathbf{y}_c; n) \approx \alpha PS(\hat{x}_l; n) + A\,, \tag{35}$$

where α is the scaling factor due to scene albedo and surface shading, and A is the offset due to constant ambient illumination. For each camera pixel, the correspondence can be found by first extracting and thresholding the intensities in its 1D spatial neighborhood to remove the effect of scene albedo, shading and ambient illumination. This is shown in Figure 21 (c). The thresholded intensities are then compared to the set of all projected spatial intensity codes by performing cross-correlation. The location of the best-match is returned as the correspondence $\hat{x}_l$ (Figure 21 (d)):

$$\hat{x}_l = \arg\max_{x_l} I_{th}(\mathbf{x}_c, \mathbf{y}_c; n) \star PS(x_l; n)\,, \tag{36}$$

where $\star$ is the cross-correlation operator, and $I_{th}(\mathbf{x}_c, \mathbf{y}_c; n)$ is the captured image subsequence after thresholding. There are several other correlation measures that could be used instead of

[7] The captured image is rectified by applying a geometric transformation to it so that the projector and the transformed camera image plane are co-planar. Each row in the rectified camera image corresponds to a row in the projector image plane. For details on image rectification, see [Hartley and Gupta 1993; Papadimitriou and Dennis 1996]

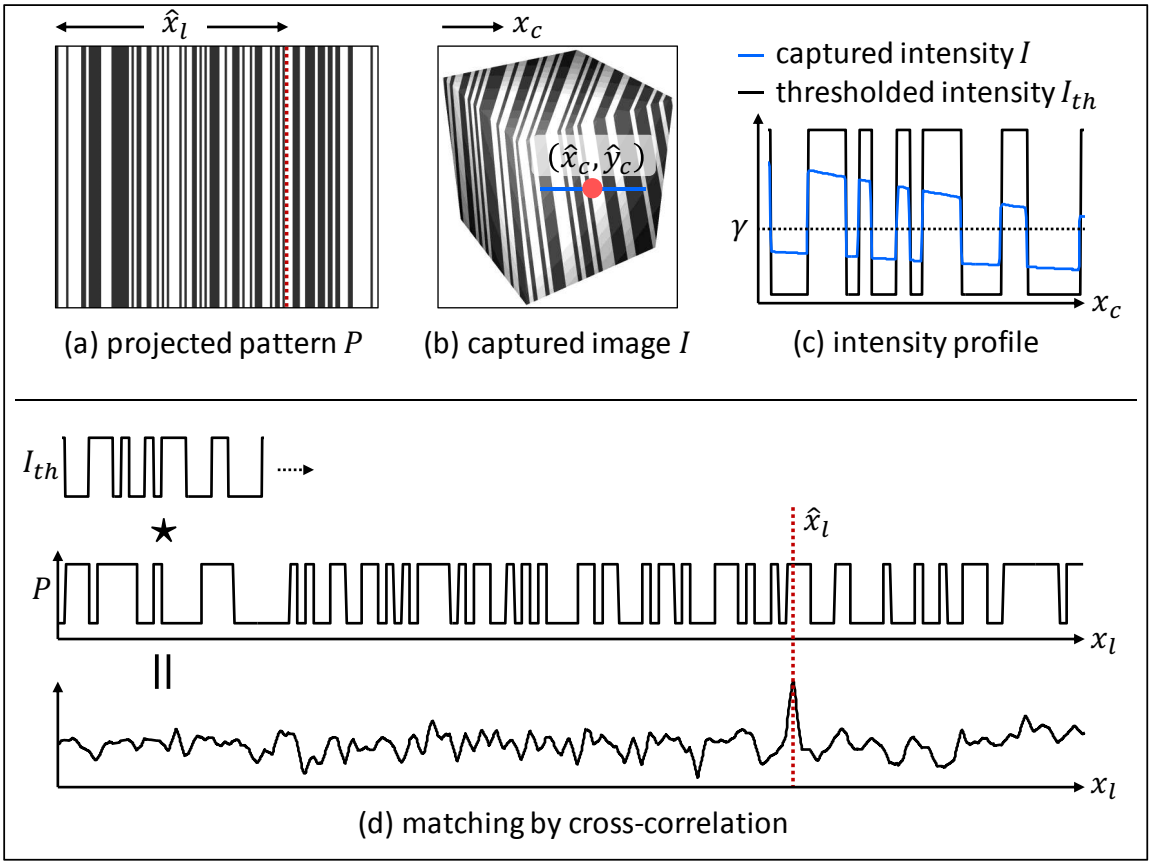


Fig. 21. **Single shot deBruijn coding.** Single shot coding methods require projecting and capturing only a single image. The projected pattern is designed so that each of its columns has a unique spatial intensity code, i.e., a unique intensity distribution in its local spatial neighborhood. This could be achieved by using De Bruijn sequences in which each sub-sequence (of a specified number) of consecutive symbols appears only once. (a) An example pattern designed using a binary De Bruijn sequence. (b) Captured image. (c) For each camera pixel, the correspondence can be found by first extracting and thresholding the intensities in its 1D spatial neighborhood to remove the effect of scene albedo, shading and ambient illumination. (d) The thresholded intensities are then compared to the set of all projected spatial intensity codes by performing cross-correlation. The location of the best-match is returned as the correspondence.

cross-correlation, such as the sum of squared differences (SSD), or the sum of absolute differences (SAD).

Dealing With Surface Discontinuities. The above matching algorithm implicitly assumes that the order of symbols in the projected sequence remains the same in the captured image. If the scene has strong depth discontinuities, the order of the projected sequence may change. In order to deal

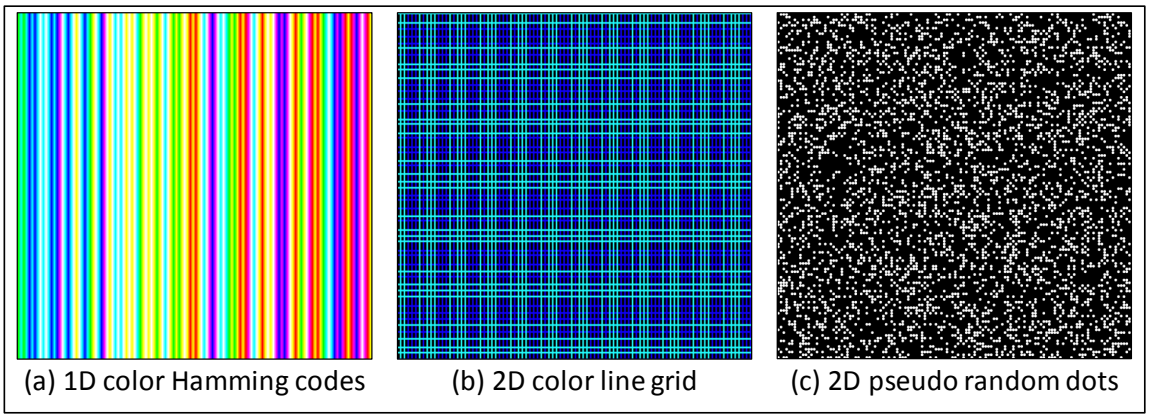


Fig. 22. **Example single shot patterns.**

with such scenes, more sophisticated matching algorithms are used. These algorithms are based on dynamic programming [Zhang et al. 2002; Yamazaki et al. 2011] and perform matching globally over the entire image, instead of matching only local image neighborhoods.

Other Single-Shot Patterns. There are several other single-shot patterns and corresponding matching algorithms, for example, 1D color De Bruijn codes [Zhang et al. 2002; Yamazaki et al. 2011], sparse set of 1D stripes with random cuts [Maruyama and Abe 1993], multiple sets of 1D stripes for all-round 3D scanning [Furukawa et al. 2010], sparse 2D grid of lines [Salvi et al. 1998; Proesmans et al. 1996b; 1996a], 2D color encoded grids [Boyer and Kak 1987; Sagawa et al. 2009], 2D pseudo-random binary code [Vuylsteke and Oosterlinck 1990], and 2D random dots (used in the first generation Microsoft Kinect depth sensing cameras). Some example patterns are shown in Figure 22. For methods that use 2D patterns, the matching is performed on 2D spatial neighborhoods. A single-shot method that adapts the projected pattern according to the scene content for achieving high accuracy and robustness was proposed by [Koninckx and Van Gool 2006]. Interested readers are referred to [Pages et al. 2005; Koninckx and Van Gool 2006] for an overview of single shot techniques.

A related approach is the stripe boundary codes [Rusinkiewicz et al. 2002; Hall-Holt and Rusinkiewicz 2001], where multiple patterns with 1D stripes are projected sequentially on the scene. Correspondences are computed for each captured image by matching spatial gradients of captured and projected images, in spatio-temporal image windows. This is shown in Figure 23. Although this is not a single-shot method, since matching is performed at every time instant, this technique can capture dynamic scenes.

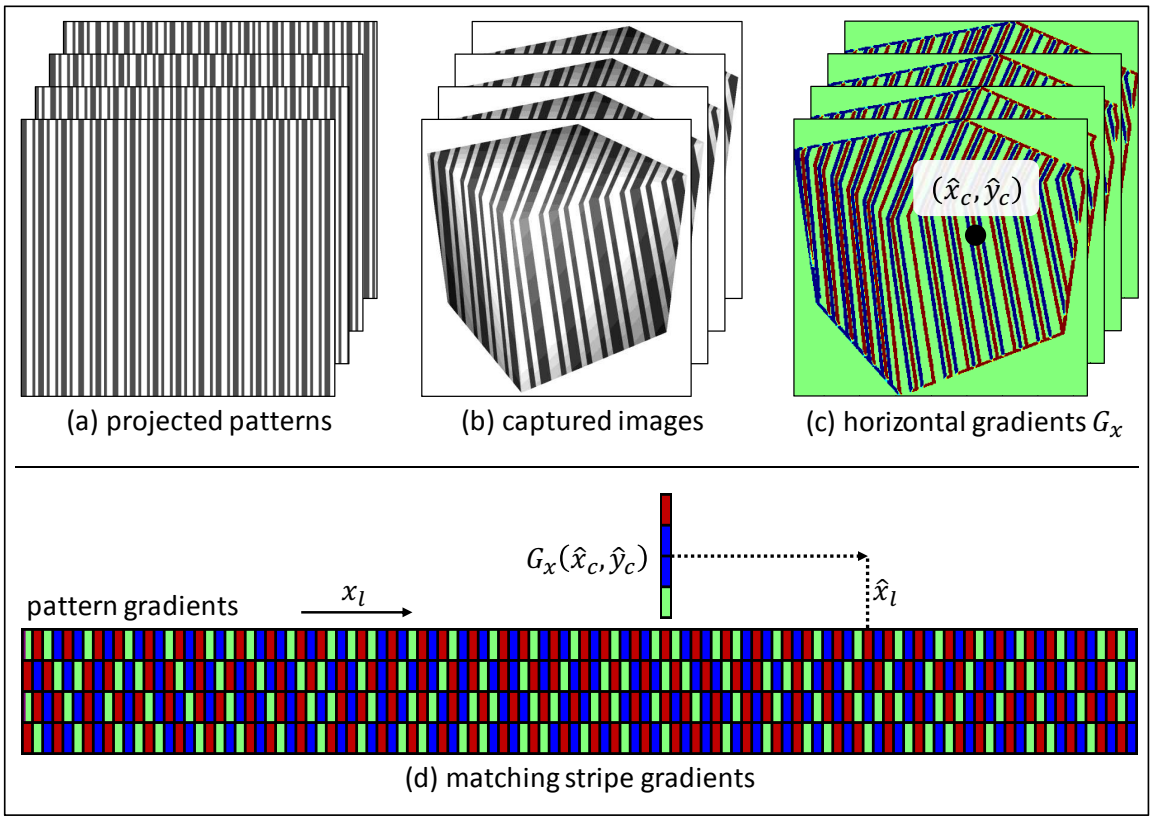


Fig. 23. **Stripe boundary coding.** (a) Multiple patterns with 1D stripes are projected sequentially on the scene. (b) One image is captured for every projected pattern. (c) Horizontal gradients of captured images. (d) Correspondences are computed for each captured image by matching spatial gradients of captured and projected images across spatio-temporal image windows. Although this is not a single-shot method, since matching is performed at every time instant, this technique can capture dynamic scenes.

Resolution Vs. Speed Tradeoff. Although single-shot methods can capture 3D shape of dynamic scenes, they make different assumptions about the scene, and are applicable only if the assumptions are satisfied. Specifically, most single-shot techniques assume that depths and reflectance properties vary smoothly across the scene. As a result, the depths computed by single-shot methods are spatially smoothed, and are often devoid of the fine surface details. Thus, there is a tradeoff between spatial and temporal resolution achieved by structured light techniques. While multi-shot methods achieve high spatial resolution (due to per-pixel processing) and low temporal resolution, single-shot techniques achieve high temporal and low spatial resolution. Recently, a hybrid technique has been developed that adapts the projected patterns according to the scene con-

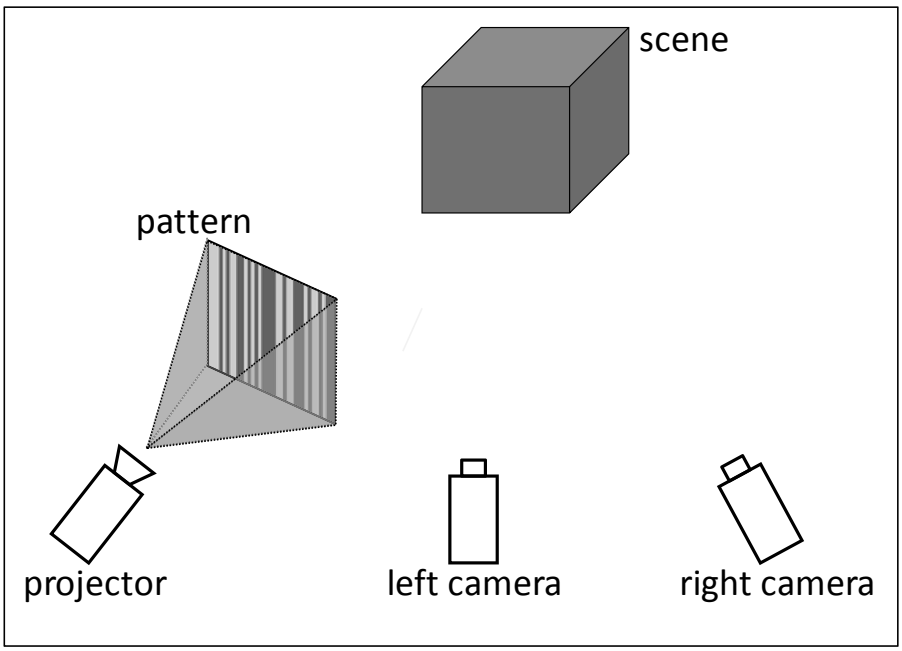


Fig. 24. **Stereo with structured light.** These techniques use spatially coded light sources with a two-camera-stereo setup. Scene depths are computed by triangulating corresponding rays from the two cameras, as in conventional binocular stereo. The light source projects a pattern on the scene which acts as scene texture. This allows the correspondences between the two images to be established reliably even for otherwise textureless scenes.

tent [Taguchi et al. 2012]. With this technique, it is possible to achieve high spatial resolution for static parts of the scene while simultaneously achieving high temporal resolution for dynamic parts of the scene.

9. STEREO WITH STRUCTURED LIGHT

So far in this chapter, we have considered techniques where depth is recovered by intersecting a camera ray and the corresponding projector ray/plane. These techniques are implemented with a single camera and a spatially coded light source. A related class of techniques, called *stereo with structured light (SwSL)*, use spatially coded light sources with a two-camera stereo setup. Scene depths are computed by triangulating corresponding rays from the two cameras, as in conventional binocular stereo. This is illustrated in Figure 24. In conventional stereo, pixel correspondence (and hence, depth) cannot be computed reliably if the scene does not have texture. In a SwSL system, the light source projects a pattern on the scene which acts as scene texture. This allows the cor-

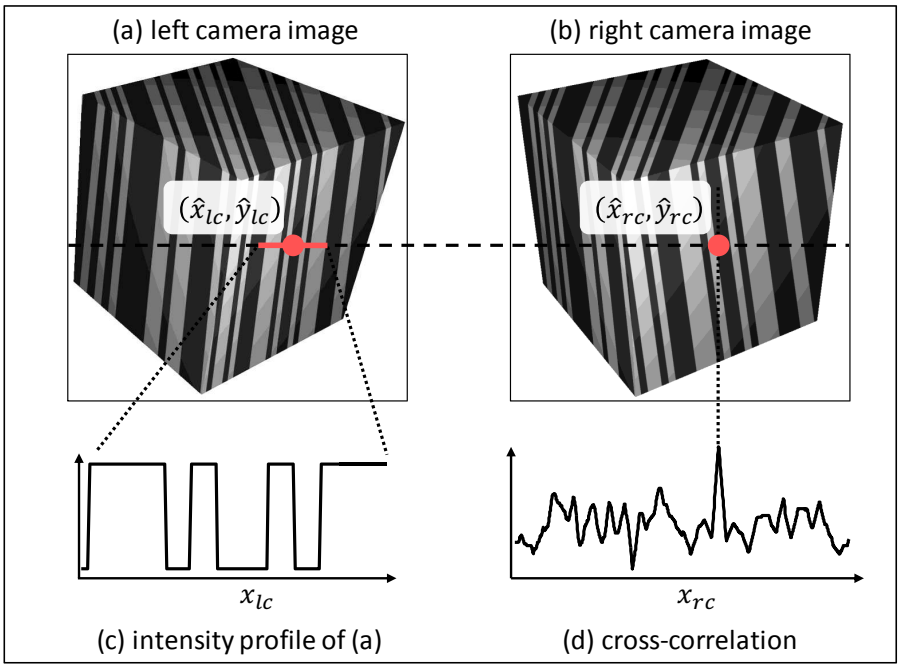


Fig. 25. **Single shot stereo with structured light.** In stereo with structured light (SwSL), a spatially coded light source is used with a two-camera stereo setup. The light source projects a pattern on the scene which acts as scene texture. This allows the correspondences between the two images to be established reliably even for otherwise textureless scenes. Scene depths are computed by triangulating corresponding rays from the two cameras, as in conventional binocular stereo. (a-b) Single shot SwSL methods project a single pattern on the scene, and both the cameras capture an image. Corresponding pixels in the two images are found by matching intensities in local patches around pixels. (c) Intensity profile of a horizontal 1D patch around a pixel in left image. (d) Correspondence is found by computing cross-correlation with 1D patches around pixels in the right image.

respondences between the two images to be established reliably even for otherwise textureless scenes.

As compared to single camera structured light, the advantage of SwSL is that it does not require the light source to be calibrated (geometrically or radiometrically), thus simplifying the hardware implementation. This is because the light source is used only as scene texture, not for triangulation. Moreover, in SwSL, triangulation is performed between two cameras, instead of a camera and a projector. Since cameras typically have a higher spatial resolution than projectors, SwSL can achieve higher depth resolution as compared to single camera structured light.

9.1. Single-Shot Methods

Like structured light, SwSL systems can also be classified into single-shot vs. multi-shot methods. Single-shot methods project a single pattern on the scene, and both the cameras capture an image. Let the two images be I_{lc} and I_{rc} for the left and the right camera, respectively. Two example captured images are shown in Figure 25.

Suppose a scene point S is imaged at pixel $\hat{\mathbf{p}}_{lc} = (\hat{x}_{lc}, \hat{y}_{lc})$ in the left image, and $\hat{\mathbf{p}}_{rc} = (\hat{x}_{rc}, \hat{y}_{rc})$ in the right image. $\hat{\mathbf{p}}_{lc}$ and $\hat{\mathbf{p}}_{rc}$ are corresponding pixels. We assume that the images are rectified, so that corresponding pixels in the two images lie on the same horizontal scan-lines, i.e., $\hat{y}_{lc} = \hat{y}_{rc}$. Thus, given a pixel $\hat{\mathbf{p}}_{lc}$, finding its corresponding pixel $\hat{\mathbf{p}}_{rc}$ in the right image involves a 1D search for the horizontal coordinate $\hat{x}_{rc}$:

$$\hat{x}_{rc} = \arg\max_{x_{rc}} I_{lc}\left(\hat{x}_{lc}, \hat{y}_{lc}; n\right) \star I_{rc}\left(x_{rc}, \hat{y}_{rc}; n\right) \ , \tag{37}$$

where $\star$ is the normalized cross-correlation operator, and

$$I_\gamma(x_\gamma, y_\gamma; n) = \left[I_\gamma\left(x_\gamma - \frac{n}{2}, y_\gamma\right), \ldots, I_\gamma\left(x_\gamma + \frac{n}{2} - 1, y_\gamma\right)\right]$$

for $\gamma = lc, rc$ is the vector of intensities (after thresholding) in a 1D window around pixel $\mathbf{p}_\gamma$ in image I_γ. n is the size of the spatial window. This procedure is illustrated in Figure 25, and is similar to the matching procedure used in single shot structured light techniques. Once the two corresponding pixels are computed, the 3D location of the scene point can be computed by triangulation, as discussed in Section 2.

Projected Patterns. The success of correspondence estimation in binocular stereo depends on scene points having unique spatial texture in their neighborhoods. If there is no texture or if the texture is repeating, there may be multiple matches resulting in depth ambiguities. In SwSL, since the texture is provided by the projected pattern, the pattern must be such that its sub-patterns are unique. As discussed in the previous section, patterns based on De Bruijn sequences have the desired property that their sub-patterns are unique, and thus, can be used in SwSL methods [Lim 2009]. The example shown in Figure 25 uses a 1D De Bruijn pattern.

As with single-shot structured light, a variety of patterns have been used in single-shot SwSL methods. These include a pattern with color stripes [Chen et al. 1997], random dots texture [Nishihara 1984], random frequency modulated sinusoidal pattern [Kang et al. 1995] and 2D binary patterns that are resistant to noise and camera defocus blur [Konolige 2010].

9.2. Multi-Shot Methods

In multi-shot SwSL, the projector projects multiple patterns sequentially, and each camera captures an image for every pattern. For instance, the patterns could be the binary Gray coded sequence [Scharstein and Szeliski 2003] or high frequency random binary patterns [Zhang et al. 2003], as shown in Figure 26. Temporally varying laser speckle patterns can also be used [Schaffer et al. 2014]. The matching process is similar to the single-shot case. The key difference is that instead of comparing spatial image windows, corresponding pixels are computed by comparing their *temporal image windows* [Scharstein and Szeliski 2003]:

$$\hat{x}_{rc} = \arg\min_{x_{rc}} I_{lc}(\hat{x}_{lc}, \hat{y}_{lc}; N) \star I_{rc}(x_{rc}, \hat{y}_{lc}; N)\,, \tag{38}$$

where $I_\gamma\left(\mathbf{p}_\gamma; N\right) = \{I_{\gamma,i}\left(\mathbf{p}_\gamma\right) | 1 \leq i \leq N\}$ is the temporal image window, i.e., the temporal sequence of image intensities captured at a pixel $\mathbf{p}_\gamma = (x_\gamma, y_\gamma)$, for $\gamma = lc, rc$.

As in structured light, there is a speed vs. resolution tradeoff in SwSL based methods. Single-shot SwSL can handle dynamic scenes, but achieve low spatial resolution. In contrast, multi-shot SwSL achieves high spatial resolution at the cost of a higher acquisition time. In general, correspondences can be estimated by comparing spatio-temporal image windows around pixels [Davis et al. 2005; Zhang et al. 2003]. The size and the shape of the windows is determined by motion of objects in the scene. If the object motion is small, windows with smaller spatial extent and larger temporal extent are used. If the object motion is large, windows with larger spatial extent and smaller temporal extent are used.

Related Methods. SwSL can also be used in scenarios when the scene is illuminated by an uncontrolled, natural light source, instead of an artificial source. For instance, underwater natural illumination has strong spatio-temporal brightness variations due to refraction of light at the wavy air-water interface. These variations are called underwater caustics or flicker. [Swirski et al. 2009] used these intensity variations for recovering shape of underwater scenes using a binocular stereo setup. This is illustrated in Figure 27 (a). Due to time varying caustics, the brightness received (and reflected) at a scene point varies temporally. As a result, the intensity at corresponding pixels in the two images have similar temporal variations, as shown in Figure 27 (b). As discussed above, the correspondences can be recovered by comparing the temporal intensity profiles across pixels in the two images.

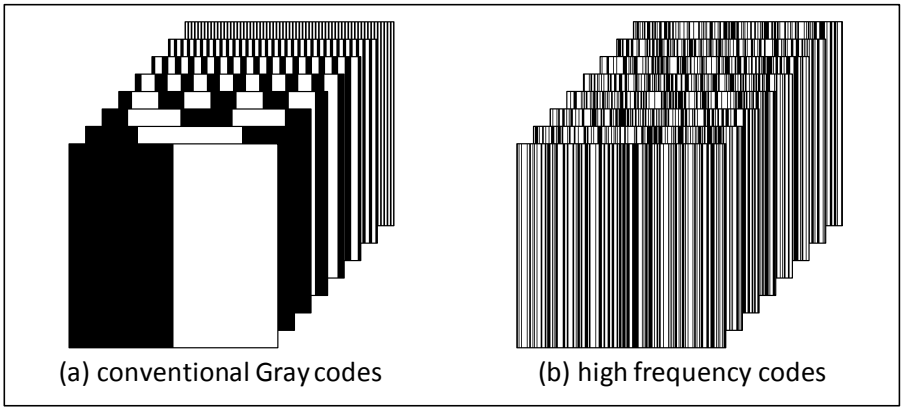


Fig. 26. **Multi shot stereo with structured light.** The projector projects multiple patterns sequentially, and each camera captures an image for every pattern. Corresponding pixels are computed by comparing their temporal intensity windows.

10. DEPTH FROM ILLUMINATION DEFOCUS

In active triangulation based depth recovery, the camera and the projector view the scene along different lines of sight. Hence, there are parts of the scene which are visible to the camera but not to the projector. For these scene parts, depths cannot be estimated, resulting in "holes" in the measured shape. This is called the "missing parts problem", and is illustrated in Figure 28.

10.1. Avoiding the Missing Parts Problem

[Girod and Scherock 1990] introduced the depth from illumination defocus (DfID) technique that avoids the missing parts problem by placing the camera and the projector so that their projection centers at the same virtual location. This is achieved by using a beam-splitter (half-mirror) that reflects part of the light incident on it, and transmits the rest. The camera and the light source are placed so that their optical axes are perpendicular to each other, and the beam splitter is placed at an angle of 45^{deg} to the optical axes. This is illustrated in Figure 29.

Scene depths are computed by using the defocus property of projectors. A projector with a large aperture has a limited depth-of-field, i.e., the projected pattern is focused perfectly only on the projector focal plane [8]. The further a scene point is from the focal plane, the more defocused

[8] So far, we have assumed that light sources have infinite depth-of-field, i.e., the projected pattern remains focused at all scene-depths. This is true only for point light sources or projectors with pin-hole apertures. In practice, projectors have a finite aperture, and a finite depth-of-field.

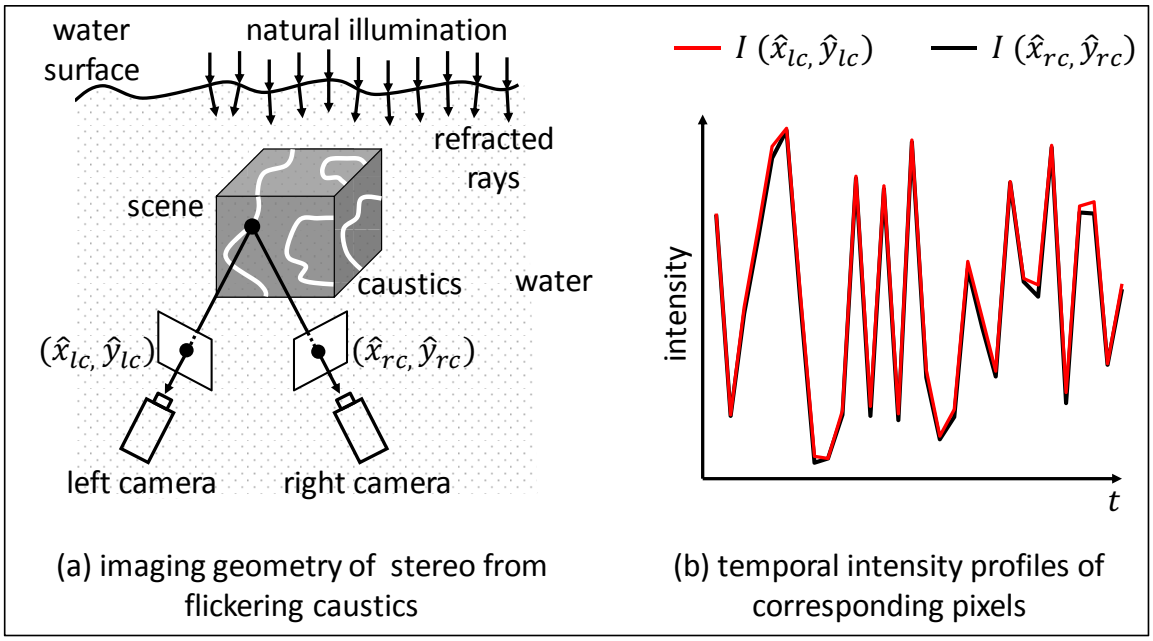


Fig. 27. **Stereo from flickering caustics.** Underwater natural illumination creates strong spatio-temporal brightness variations due to refraction of light. These variations are called underwater caustics or flicker. (a) Flickering caustics can be used for recovering shape using a binocular stereo setup. (b) Due to time varying caustics, the brightness received (and reflected) at a scene point varies temporally. As a result, the intensity at corresponding pixels in the two images have similar temporal variations. The correspondences between camera pixels can be recovered by comparing the temporal intensity profiles across pixels in the two images.

or blurred the projected pattern is. In particular, suppose the scene is illuminated with a pattern consisting of a sparse set of single-pixel dots [Moreno-Noguer et al. 2007]. A 1D illustration of the pattern is illustrated in Figure 29. If the projector has a circular aperture, each projected dot produces a circular patch in the image I captured by the camera [9]. The circular patch, the blurred version of the projected dot, is called the blur circle. If the camera has a pin-hole aperture, the diameter B of the blur circle can be written as [Moreno-Noguer et al. 2007]:

$$B = 2f_c\, r \left| \frac{1}{u} - \frac{1}{u_{foc}} \right| , \tag{39}$$

where f_c is the camera focal length, r is the radius of the projector aperture, u and u_{foc} are the distance of the scene point and the projector focal plane from the projector lens, respectively.

[9] The shape of the illuminated patch on the scene depends on the local surface geometry. However, since the projector and the camera are co-located, the shape of the patch in the camera image remains circular.

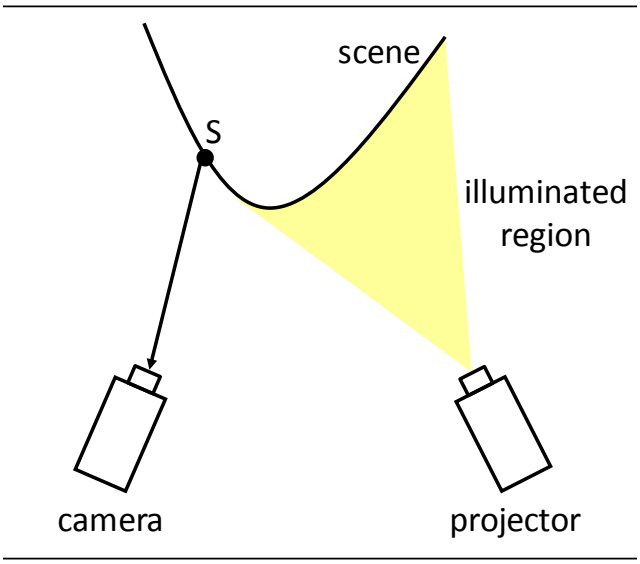


Fig. 28. **Missing parts problem.** In active triangulation based depth recovery, the camera and the projector view the scene along different lines of sight. Hence, there are parts of the scene which are visible to the camera but not to the projector. For these scene parts, depths cannot be estimated, resulting in "holes" in the measured shape. This is called the "missing parts problem".

B increases monotonically as the scene points's distance $d_{foc} = |u_{foc} - u|$ from the projector focal plane increases. The mapping between B and d_{foc} depends on the system's parameters (focal length and aperture sizes of the projector and camera), and can be estimated via a one time calibration step [Moreno-Noguer et al. 2007]. Scene depths (relative to the projector focal plane) can then be estimated by measuring the sizes of the blur circles in the captured image using standard image processing techniques.

Resolving The Two Way Depth Ambiguity By Using Asymmetric Apertures. Consider two scene points, one in front and one behind the projector focal plane so that both are equidistant from the focal plane. For both these points, the size of the blur circle in the captured image will be the same. Thus, if the projector focal plane is placed within the range of scene depths, the depths estimated by the DfID approach have a two-way ambiguity. One way to avoid the ambiguity is to focus the projector either behind the most distant scene point, or in front of the closest scene point. However, this reduces the depth resolution [Girod and Scherock 1990].

[Girod and Adelson 1990] proposed using asymmetric projector apertures to resolve this ambiguity. For instance, if the projector has a T-shaped aperture, the blurred pattern received at scene

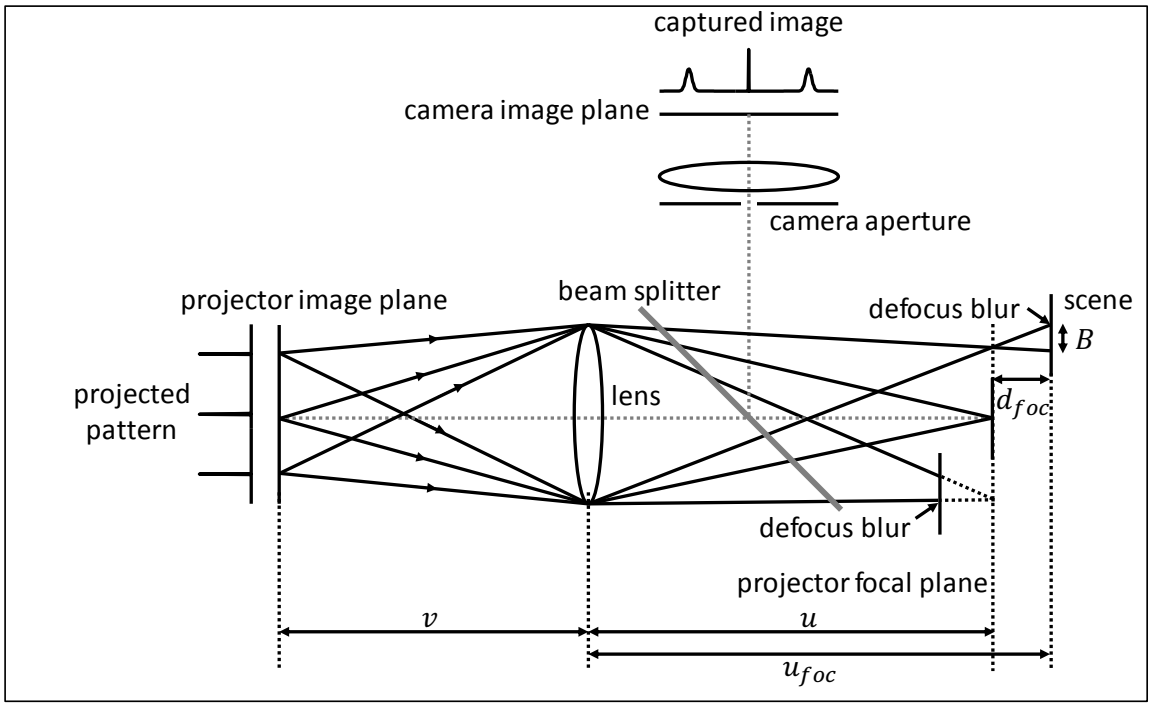


Fig. 29. **Depth from illumination defocus (DfID).** This technique avoids the missing parts problem by placing the camera and the projector so that their projection centers at the same virtual location. This is achieved by using a beam-splitter (half-mirror) that reflects part of the light incident on it, and transmits the rest. Scene depths are computed by using the defocus property of projectors. The scene is illuminated with a pattern consisting of a sparse set of single-pixel dots. The further a scene point is from the projector focal plane, the more defocused or blurred the projected pattern is. If the projector has a circular aperture, each projected dot produces a circular patch in the image captured by the camera. The diameter of the patch is proportional to the scene points's distance from the projector focal plane. Scene depths can be estimated by measuring the sizes of the circular patch in the captured image.

points is T-shaped (instead of the circular disk received with a symmetric circular aperture). This is illustrated in Figure 30. The size of the T-shaped blur received at a point is a function of its depth. For scene points behind the focal plane, T's are upside down, while for points in front, T's are upright. Thus, by using a suitable asymmetric aperture, the usable depth range of the DfID technique can be approximately doubled.

Related Methods. As in passive binocular stereo, passive depth-from-defocus approaches [Pentland 1987] need scene texture for estimating scene depths reliably [10]. Spatially coded illumination

[10] For a discussion on similarities between stereo and depth-from-defocus methods, see [Schechner and Kiryati 2000].

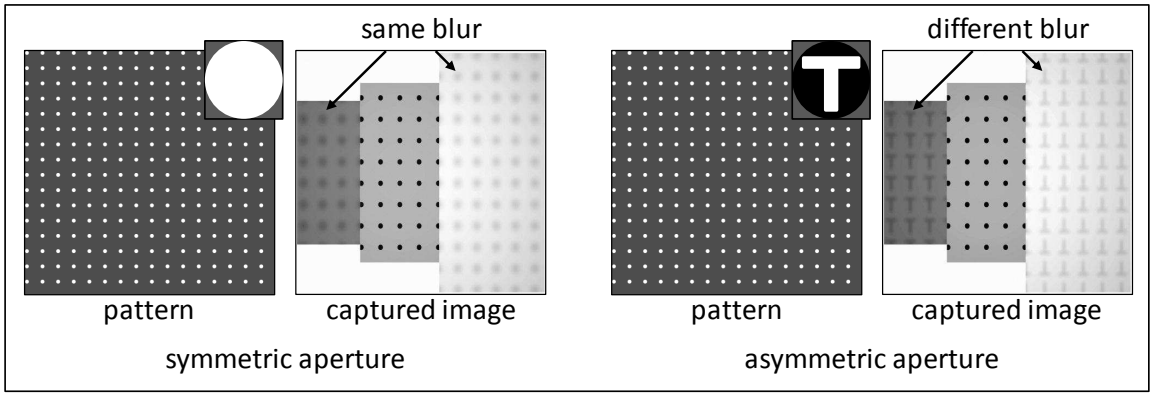


Fig. 30. **Depth from illumination defocus with asymmetric apertures.** If the projector has a symmetric circular aperture, the blur kernel in camera images has a circular shape. This results in depth ambiguities since points that are equidistant from the projector focal plane, but on its opposite sides will produce the same blur in the camera image. In contrast, if the projector has an asymmetric T-shaped aperture, the blurred pattern received at scene points is T-shaped. For scene points behind the focal plane, T's are upside down, while for points in front, T's are upright. Thus, by using a suitable asymmetric aperture, the depth ambiguity can be resolved.

has been used for adding scene texture in passive depth-from defocus techniques as well [Nayar et al. 1995]. Scene depths are computed by using camera defocus, instead of projector defocus. This is similar to techniques in Section 9 where coded illumination was used to aid correspondence computation in passive multi view stereo methods.

10.2. Recovering Per-Pixel Scene Depths

The DfID technique described above requires capturing only a single image. However, the projected pattern needs to be sparse in order to avoid overlap of the blur circles [11] in the captured image. As a result, scene depths are recovered only at a sparse set of pixels in the camera image. While an approximate dense depth map can be recovered by using a color segmentation of the image to propagate the sparse depth estimates to the entire image [Moreno-Noguer et al. 2007], this approach is applicable only to scenes with smoothly varying depths. Moreover, in order to estimate the size of the blur from the captured image, it is assumed that both scene depths and texture

[11] In general, the shape of the blurred pattern depends on the size of the aperture and the projected pattern. The pattern could be a set of dots [Moreno-Noguer et al. 2007] or a set of evenly spaced lines [Girod and Scherock 1990]. The aperture could be circular or asymmetric shaped [Girod and Scherock 1990; Girod and Adelson 1990].

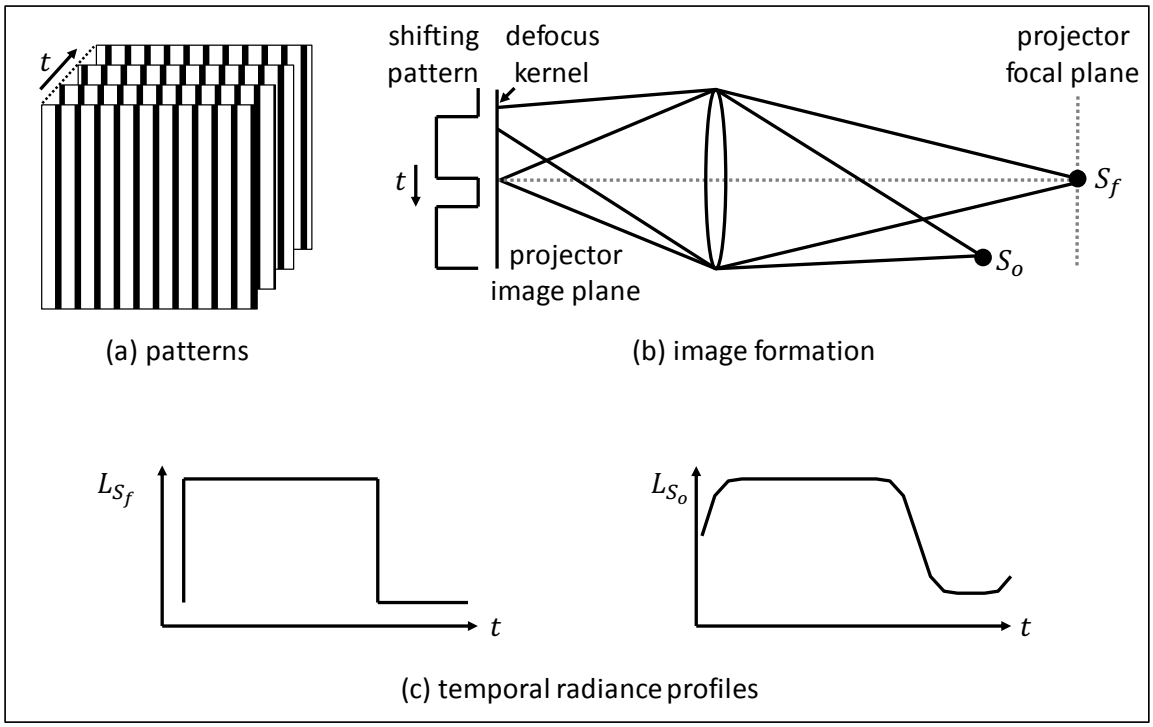


Fig. 31. **Depth from illumination defocus with multiple images.** (a) This technique involves projecting a binary pattern on the scene, for example, a set of parallel vertical stripes. The pattern is shifted one pixel at a time in the horizontal direction (perpendicular to the stripes), and an image is captured for every shift. Since the projected pattern is binary, the temporal intensity profile emitted by a projector pixel is a square wave. (b) The irradiance received at a scene point is the convolution of its projector defocus kernel with the projected pattern. (c) As a result, its temporal irradiance profile is blurred. The support size of the temporal blur is directly proportional to the scene point's distance from the focal plane, and can be estimated by analyzing the irradiance profile in the temporal frequency domain. This technique can achieve hole-free shape even for scenes with arbitrary textures and sharp depth discontinuities.

are locally smooth. This restricts the applicability of the approach as depths may be estimated incorrectly at texture/depth boundaries.

[Zhang and Nayar 2006] presented a DfID approach where scene depths are computed independently at each image pixel, without making any smoothness assumptions on the scene. The key idea is to project and capture multiple images and estimate the blur size at each camera pixel by performing *temporal image processing*, instead of spatial processing as done in single image DfID.

In particular, the technique involves projecting a binary pattern on the scene, for example, a set of parallel vertical stripes, as shown in Figure 31 (a). The pattern is shifted one pixel at a time

in the horizontal direction (perpendicular to the stripes), and an image is captured for every shift. Since the projected pattern is binary, the temporal intensity profile $P(t)$ emitted by a projector pixel is a square wave. A scene point S_f lying on the projector focal plane receives irradiance from a single projector pixel. Thus, its temporal irradiance profile $L_{S_f}(t)$ is also a square wave. However, for a scene point S_o that is out of focus, its irradiance is the convolution of its defocus kernel with the projected pattern (Figure 31 (b)). As a result, its temporal irradiance profile $L_{S_o}(t)$ is blurred, as shown in Figure 31 (c). In general, the temporal intensity profile $L_S(t)$ can be written as a convolution between the emitted profile $P(t)$ and the *temporal defocus kernel* $B_S^{temporal}(t)$:

$$L_S(t) = P(t) * B_S^{temporal}(t)\,. \tag{40}$$

Intuitively, the spatial defocus blur manifests as a temporal blur in the temporal irradiance profiles. The support size of the temporal blur $B_S^{temporal}(t)$ is directly proportional to the scene point's distance from the focal plane, and can be estimated by analyzing $L_S(t)$ in the temporal frequency domain. For details, see [Zhang and Nayar 2006]. This approach requires capturing several, approximately $20-25$ images, but the advantage is that hole-free depths can be computed for a wide range of scenes, even those with arbitrary textures and sharp depth discontinuities.

11. COMPARISON OF DIFFERENT METHODS

The table in Figure 32 provides a comparison among different spatial coding based depth recovery methods, according to the following five criteria:

(1) Number of images: The acquisition speed of a method is directly proportional to the number of images that are required. Thus, point and line scanning, while capable of achieving high quality results, may not be suitable for applications where a small acquisition time is critical.

Binary and K-ary coding, intensity ratio and phase-shifting require a relatively small number of images. Although these methods are amenable to fast acquisition, they may still not be applicable to acquisition of scenes with very high speed motion. In contrast, single-shot methods can achieve real-time acquisition and can handle dynamic scenes. Moreover, since single-shot methods require projecting a single pattern, they can be implemented with a low-cost static slide projector or even a single diffraction grating.

(2) Number of intensity levels: This factor determines whether the method requires radiometrically calibrating the projector and camera. Methods that use only two intensity levels (binary pat-

	Number of Images	Number of Intensity Levels	Per-pixel Depth	Continuous Coding	Avoids Missing Parts
Point Scanning	$O(MN)$	2	✓	✓	X
Stripe Scanning	$O(N)$	2	✓	✓	X
Binary Coding	$O(\log_2 N)$	2	✓	X	X
K-ary Coding	$O(\log_K N)$	> 2	✓	X	X
Intensity Ratio	≥ 2	> 2	✓	✓	X
Phase Shifting	≥ 3	> 2	✓	✓	X
Single Shot Coding	1	2 or more	X	X	X
SwSL (Single Shot)	1	2 or more	X	X	X
SwSL (Multi Shot)	≥ 2	2 or more	✓	X	X
DfID (Single Shot)	1	2	X	–	✓
DfID (Multi Shot)	≥ 3	2	✓	–	✓

Fig. 32. **Comparison table.** This table provides a comparison among different spatial coding based depth recovery methods.

terns) do not need the projector and camera to be radiometrically calibrated. Methods that need to project patterns with more than two intensity levels require the projector and camera to be radiometrically calibrated, and the projected pattern to be radiometrically corrected before projection [12].

(3) Per-pixel depth estimation: In single-shot methods, depth estimation at a camera pixel requires performing computations on a spatial image window around the pixel. These computations implicitly assume that the entire window belongs to a single scene surface with approximately the same depth and surface reflectance. Scenes with high frequency texture and strong depth variations violate this assumption, often resulting in erroneous or over-smoothed depth estimates.

Multi-shot methods estimate scene depths independently at each camera pixel, without considering its spatial neighborhood. Only the temporal brightness profile at each camera pixel is decoded to compute the projector correspondence. Thus, these methods can handle scenes with arbitrary texture and shape.

[12] If a K-ary coding scheme is such that every projector column emits all K intensity levels, then decoding can be performed by simple thresholding, without radiometric calibration. However, most practical K-ary coding methods do not have this property, and thus require radiometric calibration.

(4) Continuous vs. discrete coding: This factor determines the depth resolution achieved by different coding methods. Discrete schemes such as binary and K-ary assume that the light source can emit a discrete set of light planes. This limits their achievable depth resolution. On the other hand, continuous schemes such as intensity ratio and phase shifting, and also point and line scanning are compatible with light sources that can emit a continuous set of light planes (or beams). Thus, these schemes can achieve a higher depth resolution as compared to discrete methods.

(5) Avoid missing parts problem: In methods that recover depth from triangulation, the projector and the camera (or two cameras) see the scene from different view-points. There can be parts of the scene that are visible to the camera but not to the projector. Depths for these points cannot be computed (see Figure 28). In contrast, for defocus based methods (Section 10), projector and camera can be co-located. This results in *complete* depth maps, and avoids the missing parts problem.

Further Reading And Resources

Surveys and review papers: There are several papers that review different spatial light coding based shape recovery methods. Two early papers [Jarvis 1983; Besl 1988] provided an overview of several active range finding techniques, including active triangulation, depth from defocus, and time-of-flight scanning. Blais reviewed the development of active range scanners in the subsequent 20 years [Blais 2004]. Several more recent surveys have focused specifically on the coding approaches for active triangulation [Salvi et al. 2010; Sansoni et al. 2009; Geng 2011]. A review of Fourier transform profilometry is given by Su *et al.* [Su and Chen 2001].

Projector camera calibration: There are several tutorials and methods available for performing geometric calibration of the projector and the camera [Lanman and Taubin 2009; Morano and Taubin 2012]. A software for performing projector-camera calibration is available at http://mesh.brown.edu/calibration/.

Accuracy and resolution analysis: The accuracy of active triangulation based depth recovery depends on a variety of factors, including sensor and projector resolution, sensor and photon noise, speckle noise if a coherent (laser) light source is used, and the baseline. For details, the reader is referred to [Trobina 1995; Drouin and Beraldin 2012]. See also [Seitz 2007] for analyzing the effect of photon noise (shot noise) on the depth resolution of an active triangulation based depth recovery system.

REFERENCES

AGIN, G. J. AND BINFORD, T. O. 1976. Computer description of curved objects. *IEEE Trans. Comput. 25,* 4, 439–449.

ARAKI, K., SATO, Y., AND PARTHASARATHY, S. 1987. High speed rangefinder.

BESL, P. 1988. Active, optical range imaging sensors. *Machine Vision and Applications 1,* 2, 127–152.

BLAIS, F. 2004. Review of 20 years of range sensor development. *Journal of Electronic Imaging 13,* 1, 231–243.

BOUGUET, J. Y. AND PERONA, P. 1998. 3d photography on your desk. In *Proc. IEEE International Conference on Computer Vision*. 43–50.

BOUGUET, J.-Y. AND PERONA, P. 1999. 3d photography using shadows in dual-space geometry. *Int. J. Comput. Vision 35,* 2, 129–149.

BOYER, K. AND KAK, A. 1987. Color-encoded structured light for rapid active ranging. *IEEE Transactions on Pattern Analysis and Machine Intelligence 9,* 1, 14–28.

CARRIHILL, B. AND HUMMEL, R. 1985. Experiments with the intensity ratio depth sensor. *Computer Vision, Graphics, and Image Processing 32,* 3, 337 – 358.

CASPI, D., KIRYATI, N., AND SHAMIR, J. 1998. Range imaging with adaptive color structured light. *IEEE Trans. Pattern Anal. Mach. Intell. 20,* 5, 470–480.

CHAZAN, G. AND KIRYATI, N. 1995. Pyramidal intensity ratio depth sensor. *Technical Report No. 121, Department of Electrical Engineering, Technion, Haifa*.

CHEN, C.-S., HUNG, Y.-P., CHIANG, C.-C., AND WU, J.-L. 1997. Range data acquisition using color structured lighting and stereo vision. *Image and Vision Computing 15,* 6, 445 – 456.

CURLESS, B. AND LEVOY, M. 1995. Better optical triangulation through spacetime analysis. In *Proceedings of IEEE International Conference on Computer Vision*.

DAVIS, J., NEHAB, D., RAMAMOORTHI, R., AND RUSINKIEWICZ, S. 2005. Spacetime Stereo: A unifying framework for depth from triangulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 27,* 2, 296–302.

DROUIN, M.-A. AND BERALDIN, J.-A. 2012. Active 3D imaging systems. In *3D Imaging, Analysis and Applications*, N. Pears, Y. Liu, and P. Bunting, Eds. Springer, 95–138.

FORSEN, G. 1968. Processing visual data with an automaton eye. In *Pictorial Pattern Recognition*. 246–251.

FURUKAWA, R., SAGAWA, R., KAWASAKI, H., SAKASHITA, K., YAGI, Y., AND ASADA, N. 2010. One-shot entire shape acquisition method using multiple projectors and cameras. In *Pacific-Rim Symposium on Image and Video Technology (PSIVT)*. 107–114.

GENG, J. 2011. Structured-light 3d surface imaging: A tutorial. *Adv. Opt. Photon. 3,* 2, 128–160.

GHIGLIA, D. C. AND PRITT, M. D. 1998. *Two-Dimensional Phase Unwrapping: Theory, Algorithms, and Software*.

GIROD, B. AND ADELSON, E. H. 1990. System for ascertaining direction blur in a range-from-defocus camera. *US Patent 4965442*.

GIROD, B. AND SCHEROCK, S. 1990. Depth from defocus of structured light.

GOLDSTEIN, R. M., ZEBKER, H. A., AND WERNER, C. L. 1988. Satellite radar interferometry: Two-dimensional phase unwrapping. *Radio Science 23,* 4, 713–720.

GROSSBERG, M. D., PERI, H., NAYAR, S. K., AND BELHUMEUR, P. N. 2004. Making One Object Look Like Another: Controlling Appearance using a Projector-Camera System. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. I. 452–459.

GUPTA, M., AGRAWAL, A., VEERARAGHAVAN, A., AND NARASIMHAN, S. G. 2013. A practical approach to 3d scanning in the presence of interreflections, subsurface scattering and defocus. *International Journal of Computer Vision 102,* 1-3, 33–55.

GUSHOV, V. I. AND SOLODKIN, Y. N. 1991. Automatic processing of fringe patterns in integer interferometers. *Optics and Lasers in Engineering 14,* 4-5.

HALL-HOLT, O. AND RUSINKIEWICZ, S. 2001. Stripe boundary codes for real-time structured-light range scanning of moving objects. In *Proc. IEEE ICCV*. 1–8.

HARTLEY, R. AND GUPTA, R. 1993. Computing matched-epipolar projections. In *IEEE CVPR*. 549–555.

HORN, E. AND KIRYATI, N. 1997. Toward optimal structured light patterns. In *International Conference on Recent Advances in 3-D Digital Imaging and Modeling*. 28–35.

HUANG, P. S., HU, Q., JIN, F., AND CHIANG, F.-P. 1999. Color-encoded digital fringe projection technique for high-speed three-dimensional surface contouring. *Optical Engineering 38,* 6, 1065–1071.

HUGLI, H. AND MAITRE, G. 1989. Generation and use of color pseudo random sequences for coding structured light in active ranging. *Proc. SPIE 1010*, 75–82.

INOKUCHI, S., SATO, K., AND MATSUDA, F. 1984. Range imaging system for 3-d object recognition. In *International Conference Pattern Recognition*. 806–808.

JARVIS, R. A. 1983. A perspective on range finding techniques for computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence 5,* 2, 122–139.

JONGENELEN, A. P. P., BAILEY, D. G., PAYNE, A. D., DORRINGTON, A. A., AND CARNEGIE, D. A. 2011. Analysis of errors in tof range imaging with dual-frequency modulation. *IEEE Transactions on Instrumentation and Measurement 60,* 5.

JONGENELEN, A. P. P., CARNEGIE, D., PAYNE, A. D., AND DORRINGTON, A. A. 2010. Maximizing precision over extended unambiguous range for tof range imaging systems. In *IEEE Instrumentation and Measurement Technology Conference (I2MTC)*.

KANADE, T., GRUSS, A., AND CARLEY, L. 1991. A very fast vlsi rangefinder. In *IEEE International Conference on Robotics and Automation*. 1322–1329.

KANG, S. B., WEBB, J. A., ZITNICK, C., AND KANADE, T. 1995. A multibaseline stereo system with active illumination and real-time image acquisition. In *Proceedings of International Conference on Computer Vision*. 88–93.

KONINCKX, T. AND VAN GOOL, L. 2006. Real-time range acquisition by adaptive structured light. *IEEE Transactions on Pattern Analysis and Machine Intelligence 28,* 3, 432–445.

KONOLIGE, K. 2010. Projected texture stereo. In *IEEE International Conference on Robotics and Automation*. 148–155.

KOPPAL, S. J., YAMAZAKI, S., AND NARASIMHAN, S. G. 2012. Exploiting DLP illumination dithering for reconstruction and photography of high-speed scenes. *International Journal of Computer Vision 96,* 1, 125–144.

LANMAN, D. AND TAUBIN, G. 2009. Build your own 3d scanner: 3D photograhy for beginners. In *SIGGRAPH '09: ACM SIGGRAPH 2009 courses*. 1–87.

LEVOY, M., PULLI, K., CURLESS, B., RUSINKIEWICZ, S., KOLLER, D., PEREIRA, L., GINZTON, M., ANDERSON, S., DAVIS, J., GINSBERG, J., SHADE, J., AND FULK, D. 2000. The digital michelangelo project: 3d scanning of large statues. In *SIGGRAPH*. 131–144.

LIM, J. 2009. Optimized projection pattern supplementing stereo systems. In *IEEE International Conference on Robotics and Automation*. 2823–2829.

MARUYAMA, M. AND ABE, S. 1993. Range sensing by projecting multiple slits with random cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence 15,* 6, 647–651.

MORANO, D. AND TAUBIN, G. 2012. Simple, accurate, and robust projector-camera calibration. In *Proc. 3DIMPVT*.

MORENO-NOGUER, F., BELHUMEUR, P. N., AND NAYAR, S. K. 2007. Active refocusing of images and videos. *ACM Trans. Graph. 26,* 3.

NARASIMHAN, S. G., KOPPAL, S. J., AND YAMAZAKI, S. 2008. Temporal dithering of illumination for fast active vision. In *European Conference on Computer Vision*. 830–844.

NAYAR, S. K., PERI, H., GROSSBERG, M. D., AND BELHUMEUR, P. N. 2003. A Projection System with Radiometric Compensation for Screen Imperfections. In *ICCV Workshop on Projector-Camera Systems (PROCAMS)*.

NAYAR, S. K., WATANABE, M., AND NOGUCHI, M. 1995. Real-time focus range sensor. *IEEE Transactions on Pattern Analysis and Machine Intelligence 18,* 12, 1186–1198.

NISHIHARA, H. K. 1984. Prism: A practical real-time imaging stereo matcher. Tech. rep.

OIKE, Y., IKEDA, M., AND ASADA, K. 2003a. A cmos image sensor for high-speed active range finding using column-parallel time-domain adc and position encoder. *IEEE Transactions on Electron Devices 50,* 1, 152–158.

OIKE, Y., IKEDA, M., AND ASADA, K. 2003b. A row-parallel position detector for high-speed 3-d camera based on light-section method. *IEICE Transactions on Electronics E86-E*, 2320–2328.

OIKE, Y., IKEDA, M., AND ASADA, K. 2004. A 120 x 110 position sensor with the capability of sensitive and selective light detection in wide dynamic range for robust range finding. *IEEE Journal of Solid-State Circuits 39,* 1, 246–251.

PAGES, J., SALVI, J., COLLEWET, C., AND FOREST, J. 2005. Optimised de bruijn patterns for one-shot shape acquisition. *Image and Vision Computing 23,* 8, 707 – 720.

PAN, J., HUANG, P. S., AND CHIANG, F.-P. 2006. Color phase-shifting technique for three-dimensional shape measurement. *Optical Engineering 45,* 1, 013602–9.

PAPADIMITRIOU, V. AND DENNIS, T. J. 1996. Epipolar line estimation and rectification for stereo image pairs. *IEEE Transactions on Image Processing 5,* 4, 672–676.

PENTLAND, A. P. 1987. A new sense for depth of field. *IEEE Trans. Pattern Anal. Mach. Intell. 9,* 4, 523–531.

POSDAMER, J. L. AND ALTSCHULER, M. D. 1982. Surface measurement by space-encoded projected beam systems. *Computer Graphics and Image Processing 18,* 1, 1 – 17.

PROESMANS, M., VAN GOOL, L., AND OOSTERLINCK, A. 1996a. Active acquisition of 3d shape for moving objects. In *Proceedings of the International Conference on Image Processing*. Vol. 3. 647–650 vol.3.

PROESMANS, M., VAN GOOL, L., AND OOSTERLINCK, A. 1996b. One-shot active 3d shape acquisition. In *Proceedings of the International Conference on Pattern Recognition*. Vol. 3. 336–340 vol.3.

RUSINKIEWICZ, S., HALL-HOLT, O., AND LEVOY, M. 2002. Real-time 3d model acquisition. *ACM Trans. Graph. 21,* 3, 438–446.

SAGAWA, R., OTA, Y., YAGI, Y., FURUKAWA, R., ASADA, N., AND KAWASAKI, H. 2009. Dense 3d reconstruction method using a single pattern for fast moving object. In *Proc. IEEE ICCV*. 1779–1786.

SALVI, J., BATLLE, J., AND MOUADDIB, E. 1998. A robust-coded pattern projection for dynamic 3d scene measurement. *Pattern Recognition Letters 19,* 11, 1055 – 1065.

SALVI, J., FERNANDEZ, S., PRIBANIC, T., AND LLADO, X. 2010. A state of the art in structured light patterns for surface profilometry. *Pattern Recognition 43,* 8, 2666 – 2680.

SANSONI, G. AND PATRIOLI, A. 2000. Noncontact 3d sensing of free-form complex surfaces. *Proc. SPIE 4309*, 232–239.

SANSONI, G., TREBESCHI, M., AND DOCCHIO, F. 2009. State-of-the-art and applications of 3d imaging sensors in industry, cultural heritage, medicine, and criminal investigation. *Sensors 9,* 1, 568–601.

SATO, K. AND INOKUCHI, S. 1985. 3d surface measurement by space encoding range imaging. *Journal of Robotic Systems 2,* 1, 27–39.

SCHAFFER, M., GROE, M., HARENDT, B., AND KOWARSCHIK, R. 2014. Statistical patterns: An approach for high-speed and high-accuracy shape measurements. *Optical Engineering 53,* 11.

SCHARSTEIN, D. AND SZELISKI, R. 2003. High-accuracy stereo depth maps using structured light. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. Vol. 1. I–195–I–202.

SCHECHNER, Y. AND KIRYATI, N. 2000. Depth from defocus vs. stereo: How different really are they? *International Journal of Computer Vision 39,* 2, 141–162.

SEITZ, P. 2007. Photon-noise limited distance resolution of optical metrology methods. In *Proc. SPIE Vol. 6616*.

SHIRAI, Y. AND SUWA, M. 1971. Recognition of polyhedrons with a range finder. In *Proceedings of the International Joint Conference on Artificial Intelligence*. 80–87.

SRINIVASAN, V., LIU, H. C., AND HALIOUA, M. 1985. Automated phase-measuring profilometry: a phase mapping approach. *Appl. Opt. 24,* 2, 185–188.

SU, X. AND CHEN, W. 2001. Fourier transform profilometry:: a review. *Optics and Lasers in Engineering 35,* 5, 263 – 284.

SWIRSKI, Y., SCHECHNER, Y., HERZBERG, B., AND NEGAHDARIPOUR, S. 2009. Stereo from flickering caustics. In *IEEE International Conference on Computer Vision*. 205–212.

TAGUCHI, Y., AGRAWAL, A., AND TUZEL, O. 2012. Motion-aware structured light using spatio-temporal decodable patterns. In *Proc. European Conference on Computer Vision*. 832–845.

TAJIMA, J. AND IWAKAWA, M. 1990. 3-D data acquisition by rainbox range finder. In *Proceedings of the International Conference on Pattern Recognition*. 309–313.

TAKEDA, M., GU, Q., KINOSHITA, M., TAKAI, H., AND TAKAHASHI, Y. 1997. Frequency-multiplex fourier-transform profilometry: a single-shot three-dimensional shape measurement of objects with large height discontinuities and/or surface isolations. *Appl. Opt. 36,* 22, 5347–5354.

TOWERS, C. E., TOWERS, D. P., AND JONES, J. D. C. 2005. Absolute fringe order calculation using optimised multi-frequency selection in full-field profilometry. *Optics and Lasers in Engineering 43,* 7, 788 – 800.

TROBINA, M. 1995. Error model of a coded light range sensor. Tech. rep.

VUYLSTEKE, P. AND OOSTERLINCK, A. 1990. Range image acquisition with a single binary-encoded light pattern. *IEEE Transactions on Pattern Analysis and Machine Intelligence 12,* 2, 148–164.

WANG, Y., LIU, K., LAU, D. L., HAO, Q., AND HASSEBROOK, L. G. 2010. Maximum snr pattern strategy for phase shifting methods in structured light illumination. *J. Opt. Soc. Am. A 27,* 9, 1962–1971.

WUST, C. AND CAPSON, D. W. 1991. Surface profile measurement using color fringe projection. *Machine Vision and Applications 4,* 3, 193–203.

YAMAZAKI, S., NUKADA, A., AND MOCHIMARU, M. 2011. Hamming color code for dense and robust one-shot 3d scanning. In *Proceedings of the British Machine Vision Conference*. 96.1–96.9.

ZHANG, L., CURLESS, B., AND SEITZ, S. M. 2002. Rapid shape acquisition using color structured light and multi-pass dynamic programming. In *IEEE International Symposium on 3D Data Processing, Visualization, and Transmission*. 24–36.

ZHANG, L., CURLESS, B., AND SEITZ, S. M. 2003. Spacetime stereo: Shape recovery for dynamic scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*. 367–374.

ZHANG, L. AND NAYAR, S. 2006. Projection defocus analysis for scene capture and image display. *ACM Trans. Graph. 25,* 3, 907–915.

ZHANG, S., WEIDE, D. V. D., AND OLIVER, J. 2010. Superfast phase-shifting method for 3-d shape measurement. *Opt. Express 18,* 9, 9684–9689.