Active Illumination Methods

Shree K. Nayar

Monograph: FPCV-3-5

Module: Reconstruction I

Series: First Principles of Computer Vision

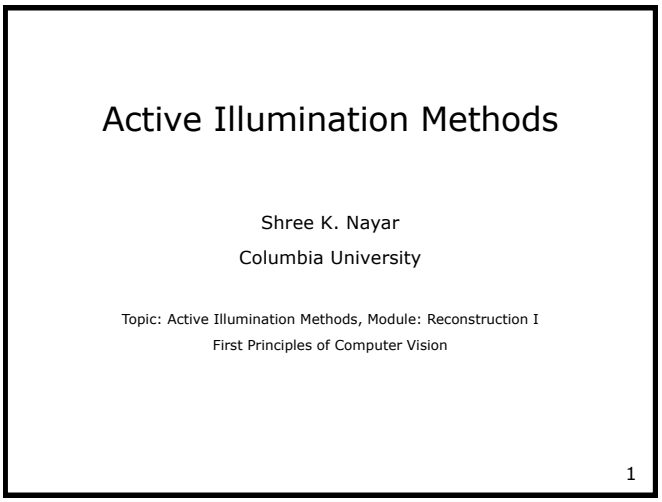
Computer Science, Columbia University

March, 2025

FPCV Channel

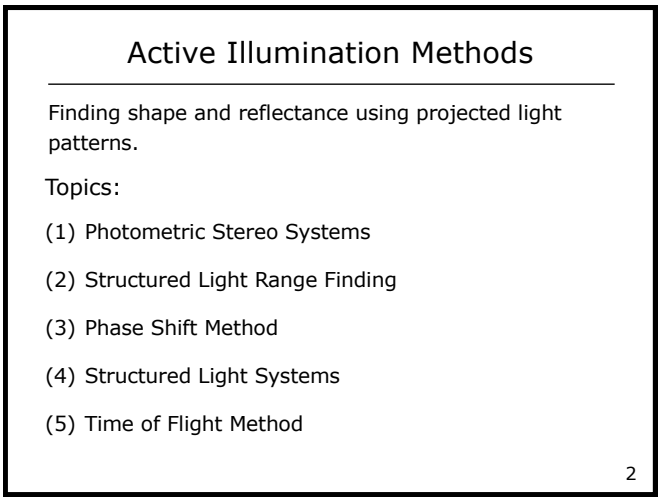
FPCV Website

In many computer vision applications, the camera is a passive observer of the scene, and scene information must be extracted from the images or videos produced by the camera. In some applications, however, we have the freedom to control the illumination of the scene. In such cases, it is possible to develop powerful methods than can extract more detailed and precise information about the scene. We refer to such methods as active illumination methods. They are extensively used for robotics, autonomous driving, factory automation and visual inspection, and represent a multibillion dollar industry.



In the context of factory automation, active illumination methods aid robots in assembling sophisticated devices such as our smartphones. Once these devices are manufactured, they are visually inspected using additional active illumination methods. In driverless cars, active illumination is used to precisely estimate, in real time, the 3D structure of the environment, which is essential for the car to be safe and intelligent. In some applications, it is not obvious that active illumination is being used as the light that is projected onto the scene is outside the visible spectrum (i.e., in the infrared or ultraviolet wavelengths), and hence invisible to the human eye.

In this lecture, we will discuss a variety of active illumination techniques. In particular, we will focus on methods that seek to extract the 3D shape and reflectance properties of objects in the scene. We will start with an active illumination method that has already been discussed—photometric stereo. This method uses a set of images captured using different light source directions to compute the surface normal at each point on an object. The 3D shape of the object is obtained by integrating these surface normals. We will briefly review photometric stereo and then discuss ways in which it can be extended to make it even more powerful.

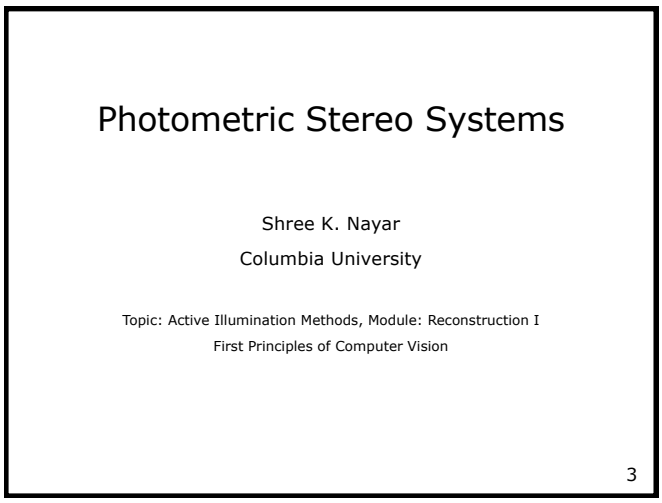


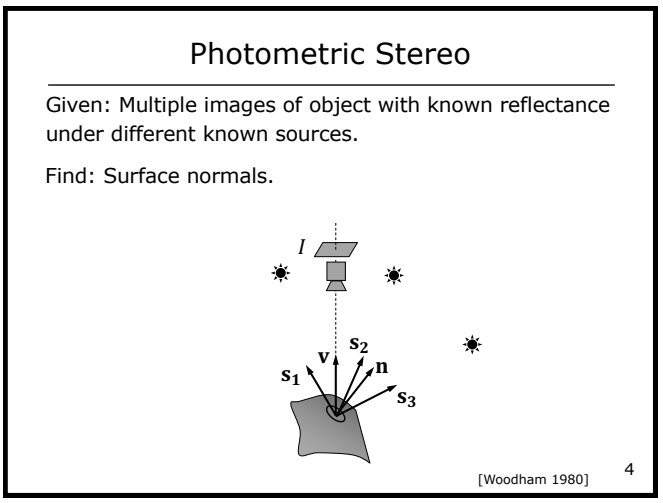
Next, we will discuss the concept of structured light range finding, where range finding is synonymous with depth estimation. The idea is to throw out a small number of light patterns onto the scene using a digital projector, capture the corresponding images, and then compute the depth at each scene point.

In our first approach, we will use a small number of discrete brightness levels (or colors) to design the projected light patterns, and show how depth can be computed at each scene point from the corresponding images. Then, we will design light patterns that are continuous in brightness. In particular, we show that three phase-shifted versions of a sinusoidal illumination pattern are sufficient to compute depth. This technique is called phase shifting.

Subsequently, we will look at a few impressive structured light systems that have been used in different application domains—performance capture, digital archiving of cultural heritage monuments, and factory automation.

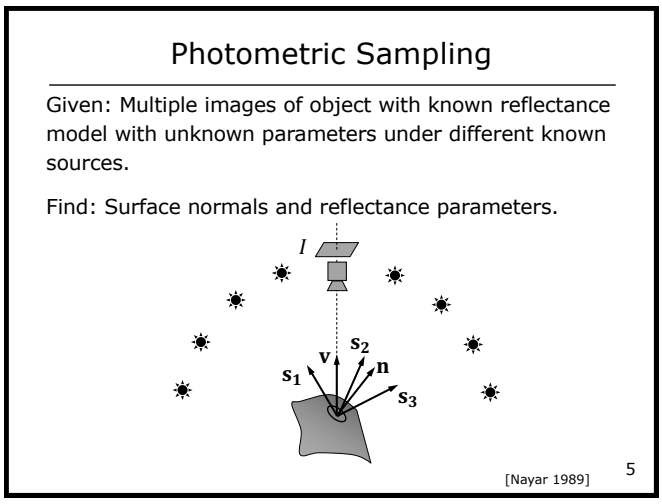
Finally, we will discuss time-of-flight methods, that estimate the time it takes for light to travel from a light source, strike a scene point, and then arrive at a sensor. If we can measure this time accurately, we can use the speed of light to compute the depth of the scene point. Time-of-flight methods are becoming increasingly popular and have begun to appear in consumer devices such as smartphones. They are expected to become ubiquitous in the next decade.



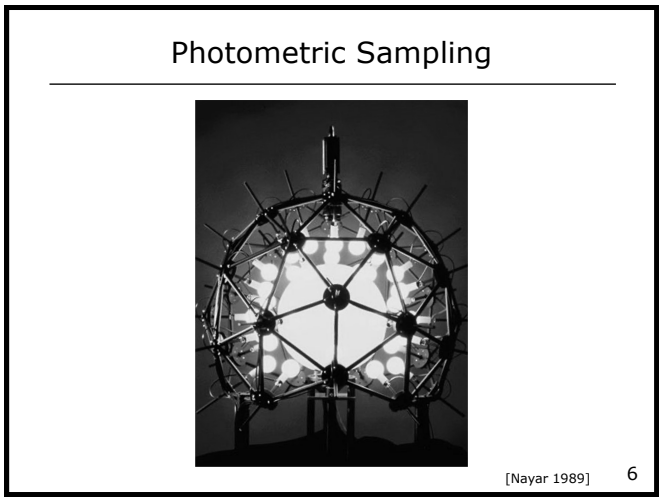


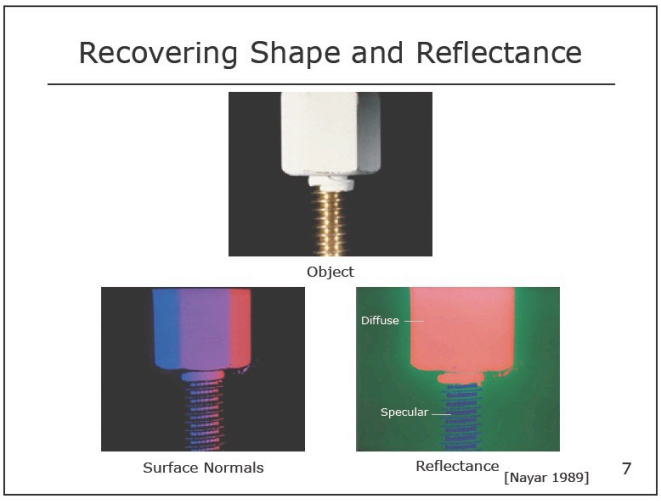
The idea behind photometric stereo is to capture multiple images of an object under light sources in different known directions. We assume that the object has known reflectance properties (BRDF). The known BRDF could be in the form of a model (equation) with known parameters, or in the form of a lookup table computed off-line using a calibration object with the same BRDF as the object of interest. Based on these assumptions, from the captured images, we can compute the surface normal at each point on the surface. One objective in photometric stereo is to keep the number of captured images to a minimum. For instance, we showed that in the case of a Lambertian surface, just three images captured under three sources in known directions are sufficient to compute the normal at each surface point.

Now, let us relax the assumption of known BRDF by assuming that the model of the BRDF is known but its parameters are unknown. For instance, the BRDF may have a diffuse and a specular component where the diffuse and specular albedos and the roughness of the microstructure of the surface are unknown. If we are able to estimate the parameters of such a BRDF model, then the reflectance at each surface point can vary—it can be Lambertian, specular, or any combination of the two. Clearly, to estimate the normal and the above reflectance parameters at each point, we would need a larger number of measurements. This is the idea behind the photometric sampling method shown here, where, compared to photometric stereo, a larger number of sources with known directions are used. Today, with LED technology, it is possible to rapidly switch between light sources and capture the corresponding images using a high-speed camera. Using the larger stack of captured images, the normal and reflectance parameters are computed at each point on the object.

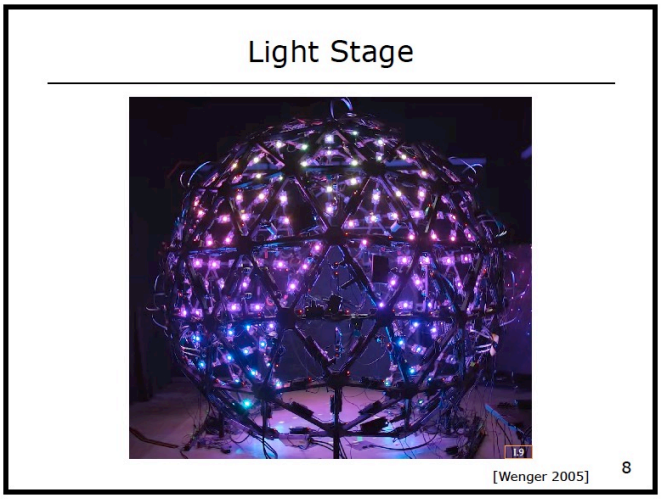


Shown here is a photometric sampler, where the object is surrounded by an array of light sources that are sequentially turned on. In this setup, a diffuser is used between the object and the light sources. The reason behind using a diffuser can be understood by considering the object to be a perfect mirrored hemisphere. To receive a non-zero camera measurement from every point on such an object, we would need an infinite number of point light sources. By using a diffuser between the sources and the object, each source is converted into an “area” source with brightness varying with direction. Therefore, we have a small number of area sources with overlapping brightness functions, and the corresponding stack of captured images is sufficient for estimating the normals at all points on a mirrored hemisphere. More importantly, the photometric sampler can recover the shape of an object where each point on its surface is free to be Lambertian, specular or any combination of the two.



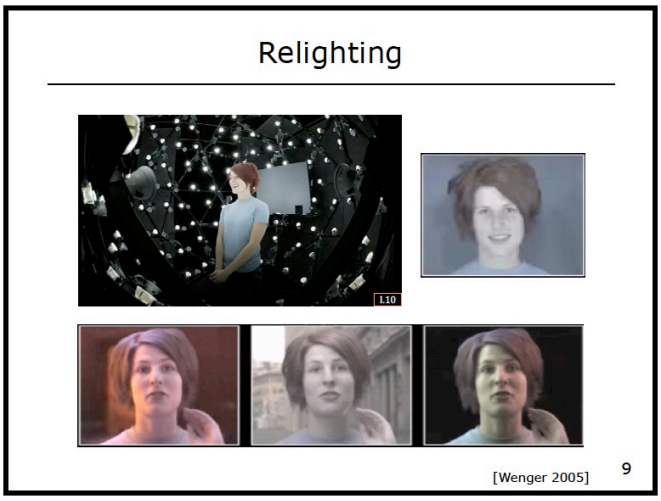


Shown here are the results of photometric sampling applied to an object that is painted (diffuse) at the top and metallic (specular) at the bottom. In the bottom-left image, the computed surface normals are coded by color. Red corresponds to a normal pointing to the right, blue corresponds to a normal pointing to the left, and the colors between red and blue correspond to normals that lie in between. We see that the 3D structure of the object has been recovered with high accuracy. On the right is shown the estimated BRDF parameters—red for Lambertian and blue for pure specular. Hence, photometric sampling is able to compute both the normal and the reflectance parameters at each point of an object.

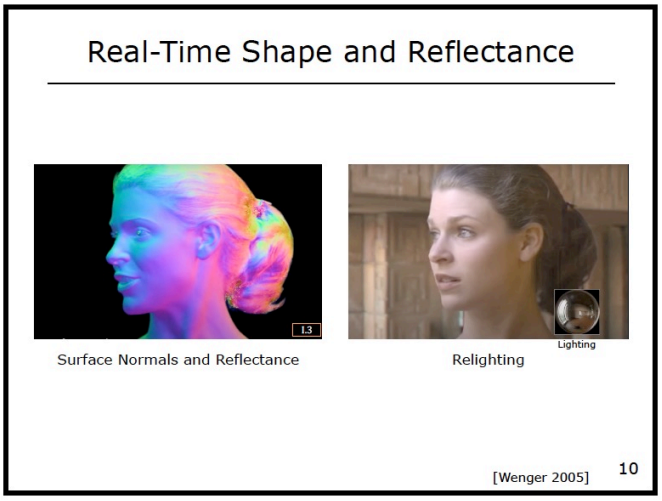


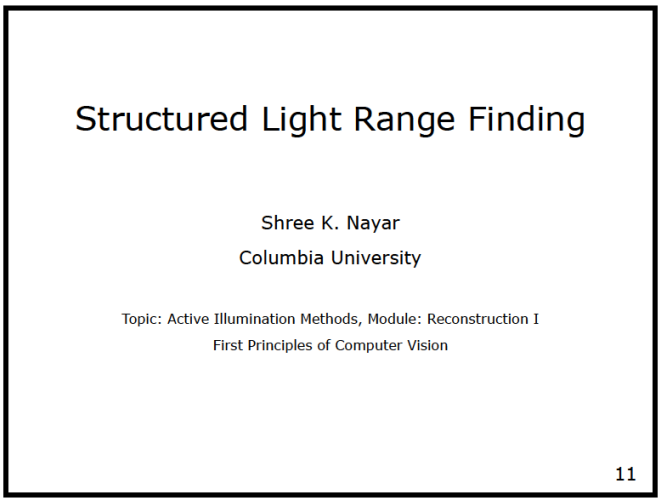
Here is another system based on the idea of photometric sampling. This is called the "light stage" and was developed by Paul Debevec's team. The light stage uses a very large number of controllable LED light bulbs and is large enough in size for a human to sit at the center of it. The light stage also uses a large number of high-speed cameras. The combination of LEDs and high-speed cameras enables the light stage to capture thousands of images per second, where in each image the object is lit or viewed differently. The high speed and accuracy of the light stage enables real-time shape and reflectance capture of moving objects. This has made it a widely used platform for human performance capture. Performances captured by the system have been used for special effects in Hollywood movies.

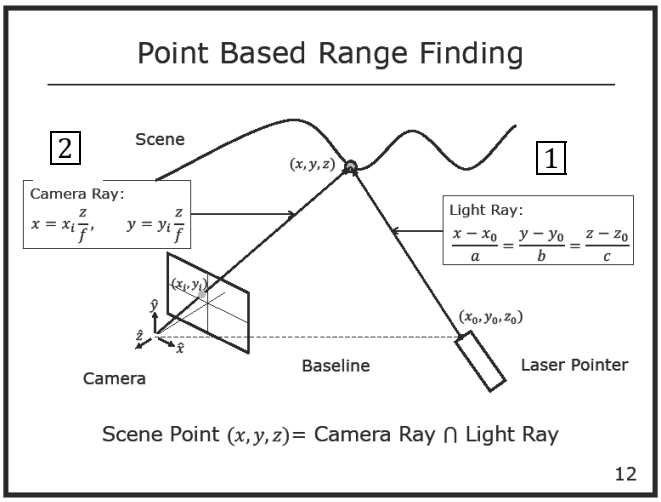
The light stage can be used for a process called relighting, where the captured images are used to render the object under a novel environmental illumination. As long as the new illumination is distance from the object, relighting can be done by simply computing a linear combination of the captured images. This makes it possible to "drop" a human performance in an arbitrary scene while ensuring the lighting of the performance is consistent with that of the scene. At the top here is shown the high-speed performance capture process. At the bottom, the performance is shown under three novel environmental illuminations.



The above relighting method uses just a linear combination of the captured images. This is adequate when the environmental illumination is distant from the object. For a more complex illumination that include proximate light sources, realistic relighting requires complete knowledge of 3D structure and reflectance. Here, on the left, is the recovered shape of an actor, shown as a color-coded normal map. This shape and the estimated reflectance properties are used to render the video on the right, where actor is relit under a complex illumination.



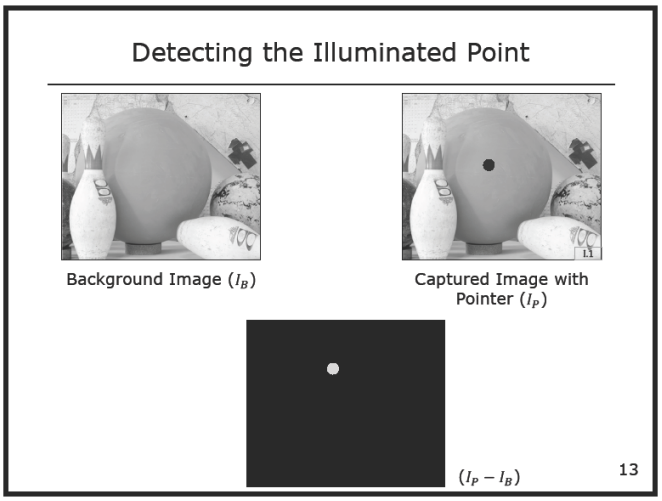




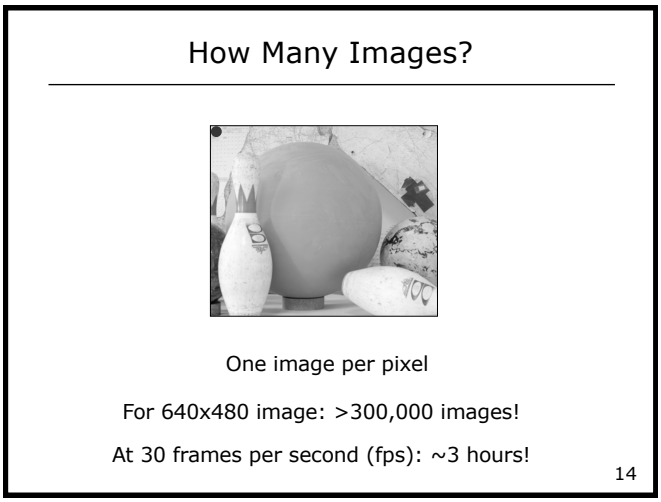
While photometric stereo and photometric sampling are designed to estimate surface normals, controlled illumination methods can also be designed to directly compute the depth of each scene point. These are commonly referred to as structured light methods. We will start with simple approaches and build up to more sophisticated ones.

The simplest structured light method is point-based range finding. As shown above, we have a 3D scene observed by a camera and illuminated by a laser pointer, which emits a single ray of light. Let us assume that the position of the laser pointer and the camera are known with respect to each other. We will assume our coordinate system to be located at the center of projection (pinhole) of the camera. Assume that we know the location (x_0, y_0, z_0) and the orientation (given by the parameters $a, b, \text{and } c$) of the laser pointer in the camera's coordinate frame. The ray it emits is then given by equation $\boxed{1}$. This ray strikes the scene at a point (x, y, z), which produces a bright scene point that is imaged by the camera at location (x_i, y_i). Perspective projection can be used to relate the scene point (x, y, z) and its bright image point (x_i, y_i) (equation $\boxed{2}$). This relationship corresponds to the camera's line of sight on which the scene point lies. Since the scene lies at the intersection of the light ray emitted by the pointer and the camera's line of sight corresponding to the bright image point, the coordinates (x, y, z) of the scene point can be found using equations $\boxed{1}$ and $\boxed{2}$. By varying the orientation of the laser pointer, the depths of all scene points are estimated. The result is a complete depth map.

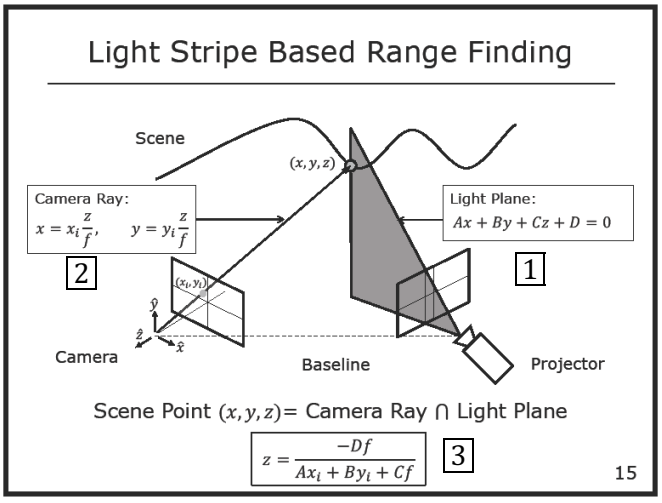
Now we will discuss how we can use point-based range finding in practice. Let's say we have a scene with some ambient illumination, as shown on the left. We refer to this as the background image I_B. When the laser pointer emits its ray, we get the image I_P in which we have a bright spot (shown here as a dark spot to make it visible). To make it easy to find the bright spot, we subtract the two images to get the image shown below, which only includes the bright spot. The centroid of this spot is used to compute (x, y, z) using the method described above. This process is repeated for all scene points. Note that the background image I_B only needs to be captured once before scanning the entire scene.



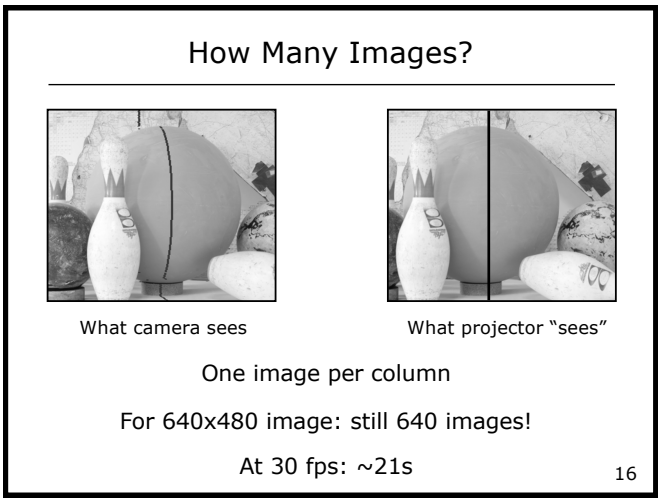
How many images would be needed to recover the depth map of the complete scene using point-based range finding? Since the measurement of each scene point requires the capture of a separate image, the number of images required equals the number of pixels in the acquired depth map. For a depth image of size 640 by 480 pixels, we would need to capture more than 300,000 images. If the camera captures images at 30 frames per second, it would take around three hours to capture a single depth map! This is obviously too long for virtually any application.



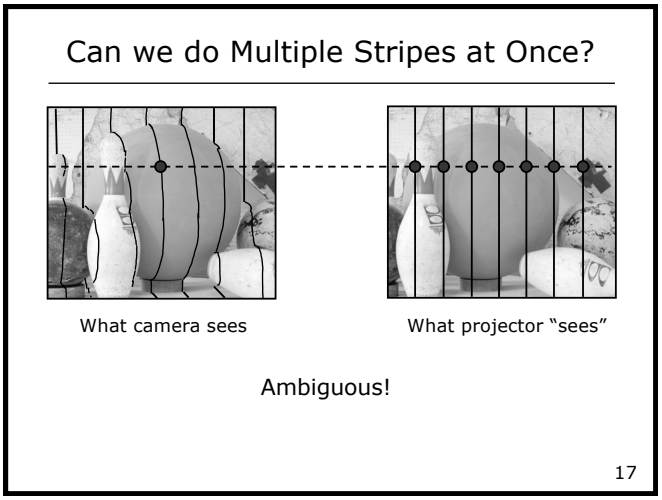
We can do a lot better by using a projector that emits a plane (sheet) of light instead of using a laser pointer that emits a single ray. This method is called light striping. The plane of light emitted by the projector intersects the scene along some curve, which gets imaged by the camera. Now consider a single bright point at (x_i, y_i) on the image curve. The corresponding line of sight is given by equation 2. The corresponding scene point (x, y, z) lies at the intersection of this line of sight and the light plane projected by the projector, which is given by equation 1. The z coordinate of the scene point, given by equation 3, is found from equations 1 and 2. The x and y coordinates can be found by plugging the z coordinate into equation 2. This process is repeated for each bright point along the image curve. To get a complete depth map of the scene, the projected light plane is swept across the scene.



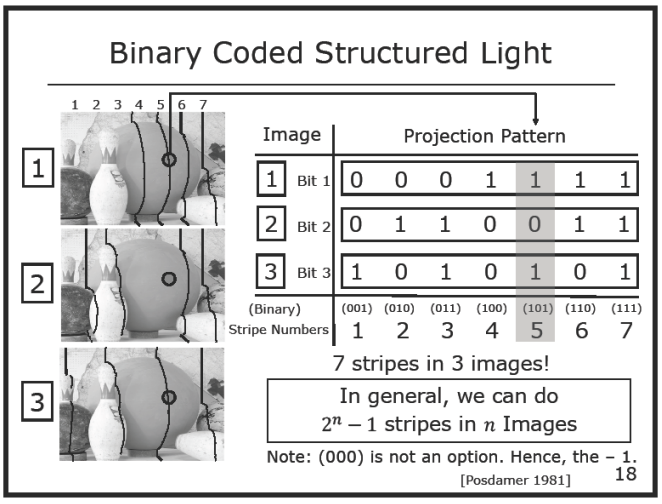
How much better do we do by light striping in terms of capture time? A single projected light plane corresponds to a single bright column in the projector image. Shown on the right is how the scene appears from the viewpoint of the projector, where the light plane is a straight line. This light plane intersects the 3D world along a curve. The image captured by the camera, which is displaced with respect to the projector, is shown on the left. To compute the depth map of the entire scene, we need to project one light plane for each column of the projector. For a projector image of size 640 x 480 pixels, where 640 is the number of columns, we will need to capture 640 images, which take roughly 21 seconds. This is a vast improvement over point-based range finding, but is still way too long for most applications.



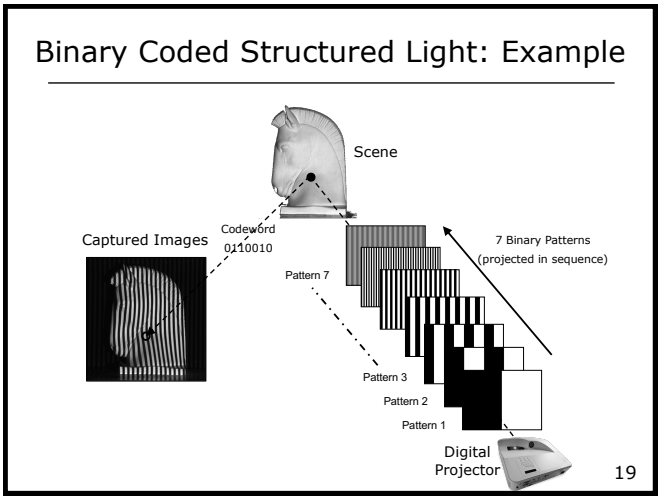
To speed up the capture, could we project multiple stripes at the same time? On the right, seven light planes are simultaneously projected on the scene, and on the left is the captured image. At first glance, it may seem that we should be able to take any one of these curves in the captured image and figure out which column in the projector produced that curve. Unfortunately, there is no way to figure this out, especially when the scene has complex depth variations. If we consider the bright point (large dark dot) shown in the camera image, it could have been produced by any of the seven light planes in the projector image. In fact, each light plane would yield a different, plausible point in the scene. In short, using multiple simultaneously projected light planes results in ambiguities.

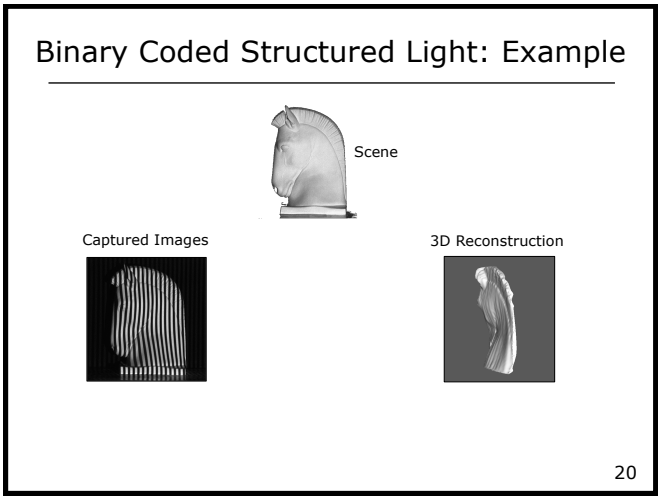


This leads us to the idea of binary coded structured light. Consider the example shown here where we wish to project a total of seven stripes. Each of the seven stripes is numbered from 1 to 7. Consequently, we can represent the seven stripes in binary using three bits. Now, stripe 1 is represented as (001) and stripe 7 as (111). We can therefore represent the seven stripes using three rows, where each row corresponds to a specific bit (Bit 1, Bit 2, and Bit 3).



For our first captured image 1, we activate all the stripes for which Bit 1 is 1 and keep the rest of the stripes off. The same process is repeated using the rows corresponding to Bit 2 and Bit 3, to obtain camera images 2 and 3, respectively. Now, let us consider the point in the three images, denoted by the black circle. This point is illuminated in the first image, not illuminated in the second image, and illuminated in the third image. The only light stripe that was on in the first capture, off in the second, and on in the third, is stripe number 5. Therefore, for seven stripes we are able to uniquely determine which stripe illuminated a scene point using just three images. In general, we can scan the scene with $2^n - 1$ stripes by capturing only n images. The -1 accounts for the fact that we do not use the stripe number 0, since it would result in a captured image in which all the stripes are off, which does not produce any useful information. As an example, we can scan the scene with 255 (i. e., $2^8 - 1$) stripes by capturing just 8 images. Using a high-speed projector and a high-speed camera, a complete depth of the scene can be computed in real-time.

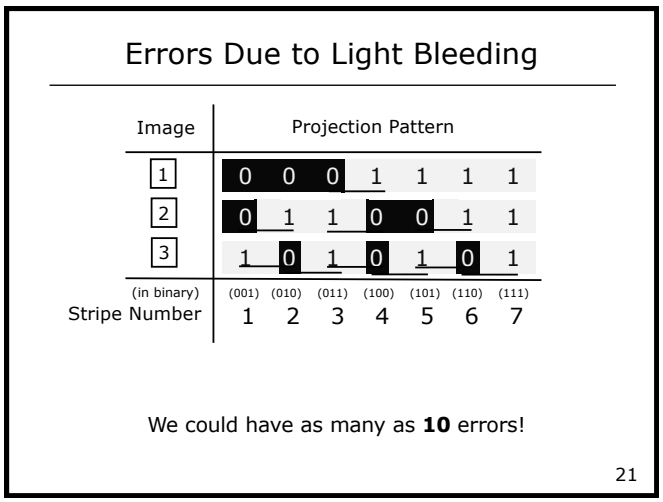




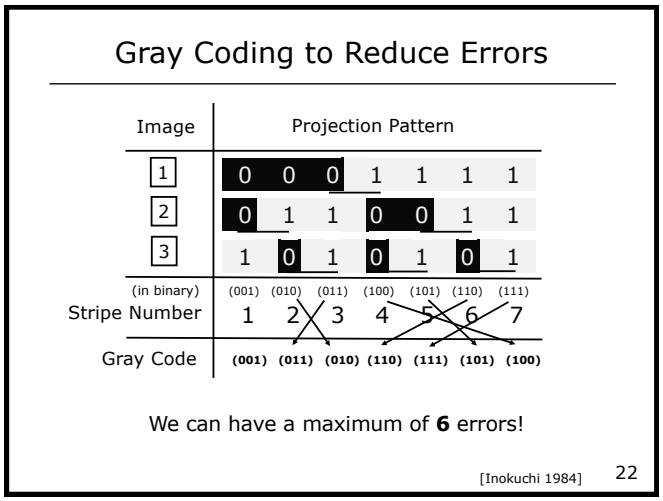
Shown on the left is an example of a binary coded structured light system. The system consists of a digital projector that emits light patterns onto the scene. In this example, seven patterns are projected onto the scene and the corresponding seven images are captured using a camera. The binary pattern (where

1 is bright and 0 is dark) measured at each pixel can be used to uniquely determine the light stripe that illuminated the corresponding scene point. This information is all we need to compute the 3D coordinates of the scene point. The computed 3D shape of the object in the scene is shown on the right.

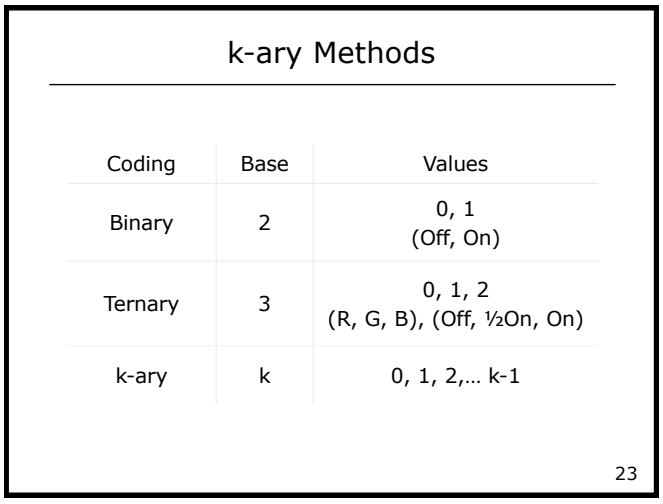
Now, let us discuss a practical problem related to binary coded structured light, called light bleeding. In each captured image, we wish to determine whether the scene point corresponding to each pixel is lit or not. However, a projector has a limited depth of field and hence the binary pattern falling on the scene could be blurred. In addition, the camera also has a limited depth of field and hence the image of the scene could also be blurred. Both these phenomena make pixels corresponding to scene points that lie at transitions of a light pattern from on to off, or vice versa, unreliable. In the example of seven stripes shown here, there are 10 such transitions. At these transitions, the measured image intensities are neither bright nor dark and hence need to be classified as a 0 or a 1 using a threshold value. This naturally leads to errors in detected binary patterns. Note that even a single erroneous bit in a binary pattern can result in a large error in the number of the light stripe, and hence a large error in computed depth.



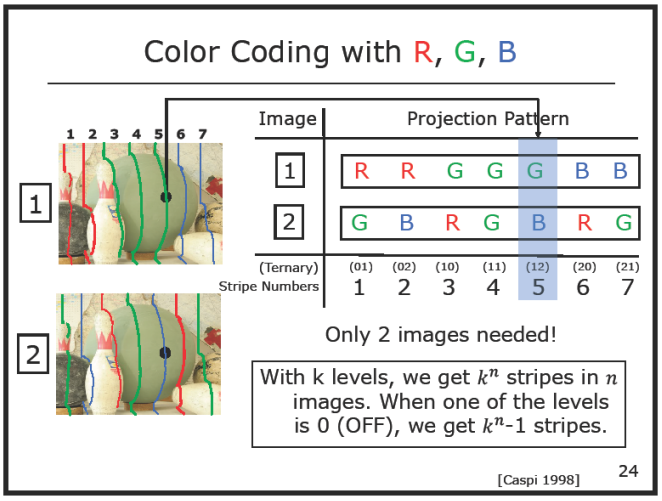
Gray coding is a method used to mitigate errors due to light bleeding. While it does not resolve the problem completely, it can reduce errors. The idea is simply to reduce the number of transitions in the projected light patterns. Consider again the seven stripes that were initially numbered from one to seven. We swap the numbers that are assigned to the seven stripes such that the third stripe is now represented by 2, the seventh stripe by 5, the fifth stripe by 6, and so on. Swapping these numbers essentially changes the binary code associated with each of the seven stripes. This results in a reduction in the number of transitions from 10 to 6. Thus, by simply changing the numbering of the stripes, we reduce the number of transitions, and hence the errors due to light bleeding.



We can further extend the idea of binary coded structured lighting by using projected patterns with more than two levels of brightness. We could use the ternary system that corresponds to three levels of brightness—one of them could be off, another could be a large brightness, and the third could be a brightness in between the previous two levels. The three levels could also be three different colors—say, red, green, and blue. The use of the ternary system requires capturing even fewer images than the binary system. We can further extend this idea to k levels. The number of patterns that need to be projected, and hence the number of images that need to captured, falls dramatically as k increases.

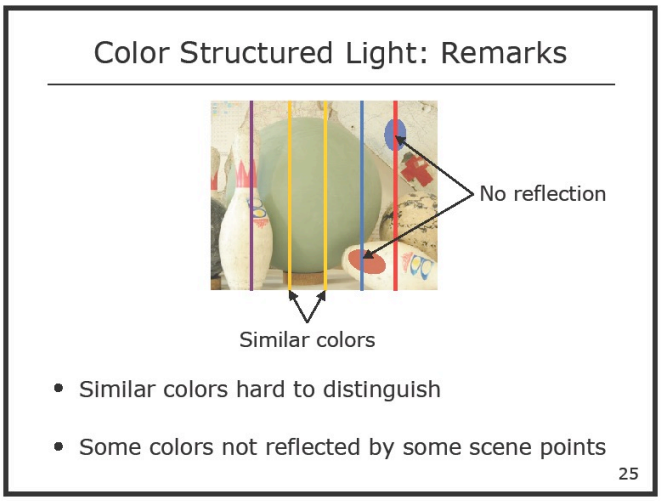


Let us take a look at the use of three colors for structured lighting. Again, we consider the coding of seven stripes. Since we are using the ternary system (three levels), we will express the stripes using ternary digits (trits) as opposed to binary digits (bits). Let 0, 1, and 2 be used to represent the three colors—red, green, and blue. Then, stripe one is represented as (01) and stripe 7 as (21). We only need two trits to represent each one of the seven stripes. Therefore, we only need to project two patterns and capture two images in this case. In the first pattern each stripe is assigned the color corresponding to its first trit, and in the second image the stripe is assigned the color corresponding to its second trit. Consider the scene point (black dot) shown in the captured images. Since it was green in the first image and blue in the second, it has to have been lit by stripe 5.

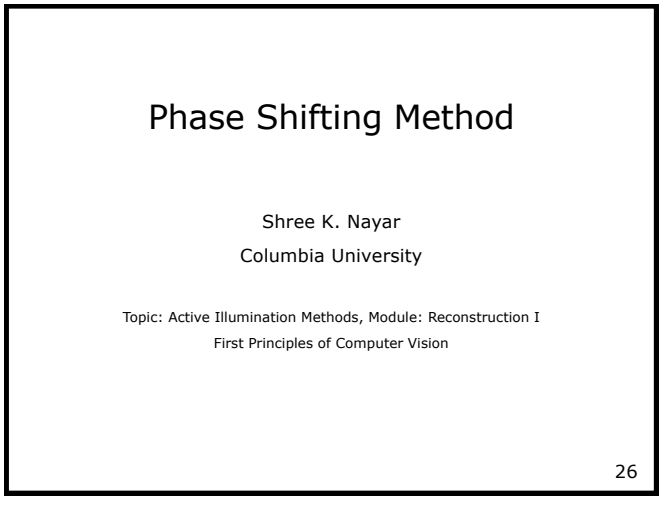


In the case of seven stripes, using the ternary system we only need to capture two images, as opposed to the three images required by the binary system. In general, with k different brightness levels, we can scan k^n stripes in just n images. As k increases, the number of captured images drops dramatically. When one of the k levels is 0 (no light projected), one of the images would have all the stripes turned off, which does not provide useful information. Hence, we can scan $k^n - 1$ stripes in n images.

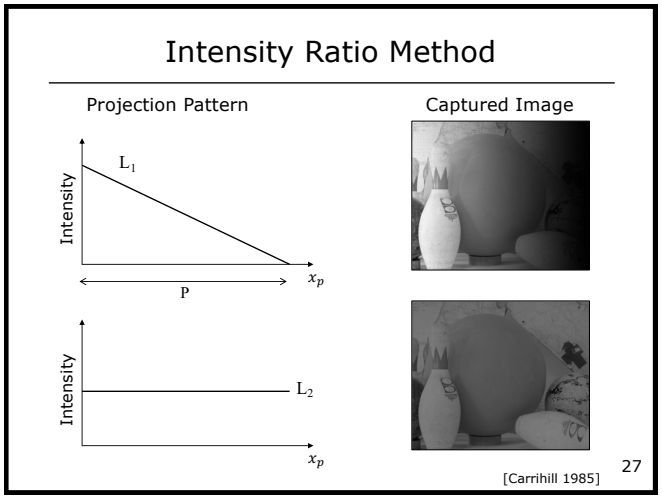
We have seen that increasing the number of colors increases the efficiency of structured lighting. However, there is a trade-off. If we are using a regular projector, each projected color will have a broad spectral band. In addition, the color filters used in a typical camera also have broad spectral bands. Let us say we are using a typical RGB projector and a typical RGB camera, and we decide to assign more than three colors to projected stripes, as shown here. Then, in the captured images different projected colors could end up appearing similar. This could be due to one of several reasons. First, the spectral bands of the projector and the camera may not be perfectly matched. Second, the object we are projecting onto has its own spectral reflectance, causing it to reflect certain wavelengths more than others. Both these effects could make it impossible to distinguish between slightly different projected colors in the captured images. Another problem could be that, because of the spectral reflectance of a scene point, we do not receive a reflection from it even when it is lit. An example of this is when we project a bright red stripe on a deep blue patch, or vice versa, as shown here. Since blue and red are at two different ends of the visible light spectrum, the blue patch is unable to reflect any of the red light, and vice versa. For all these reasons, color coded structured lighting works well only when the scene is made of gray objects, since a gray point preserves the color of the illumination it receives.



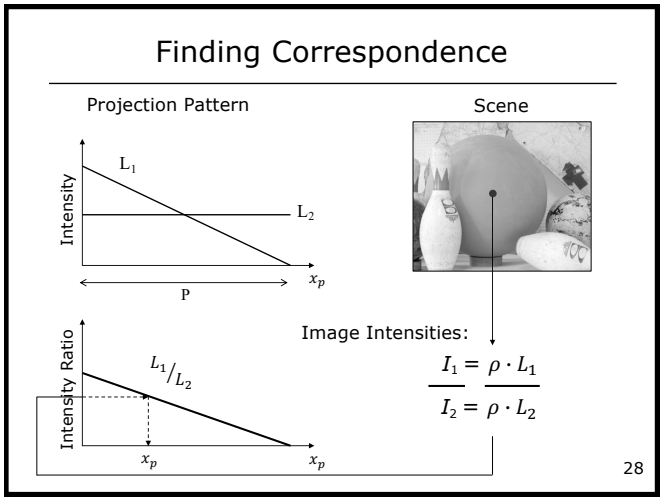
Thus far, we have created light patterns using a small number of discrete brightness levels, or colors. By strategically designing these patterns, we were able to compute the 3D structure of a scene from a small number of images. We will now look at how continuous projected light patterns can be used to estimate depth.



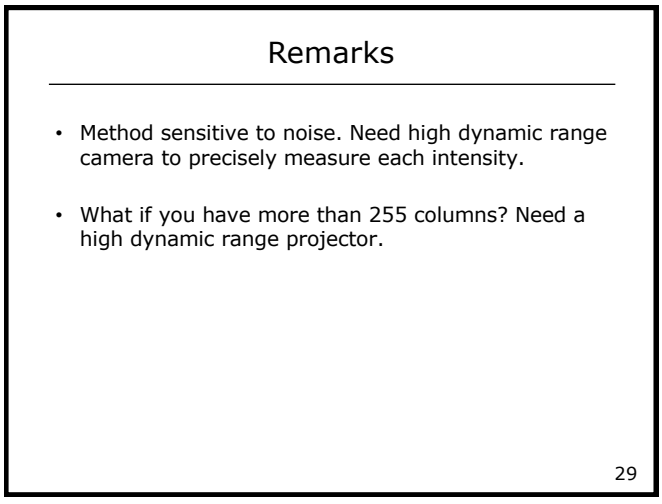
Shown here is a structured lighting method called intensity ratio. The first projection pattern L_1 is a simple ramp. It decreases linearly from a large value down at one end of the projected image to zero at the other end. The coordinate x_p corresponds to the column of the projected image. As before, our goal is to find, for each pixel in the camera, the projector column x_p that illuminated the corresponding scene point. The camera image for the projection pattern L_1 is shown on the right. Next, we take a second image (bottom-right) with projection pattern L_2, which is of uniform brightness.



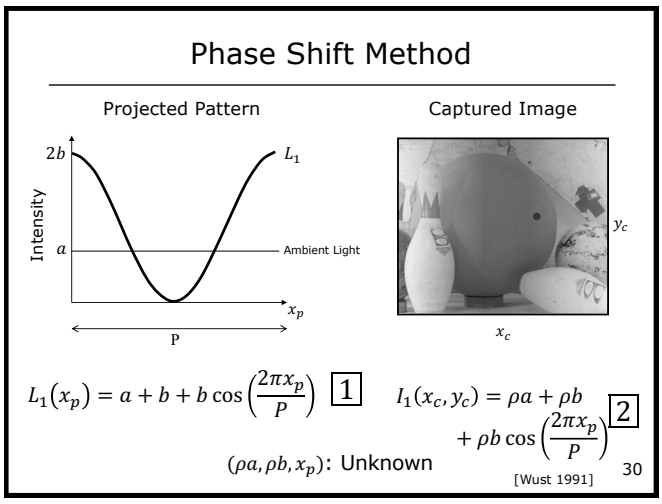
If we simply take the ratio of the functions L_1 and L_2, we get the intensity ratio function shown in the bottom-left, which is a monotonic function. Now, let us consider the intensity I_1 for a scene point. It can be expressed as an unknown constant ρ times L_1, where ρ depends on the BRDF of the scene point, its surface normal, and the gain of the camera. The intensity I_2 of the same scene point in the second image is ρ times L_2. On dividing I_1 by I_2, we get L_1 / L_2 as ρ cancels out. We can now use the intensity ratio function to find x_p from L_1 / L_2. Given x_p, we know the projector light stripe that illuminated the scene point, and hence find its 3D coordinates.



The intensity ratio method is simple as it enables us to compute a complete depth map using just two images. Unfortunately, this method requires us to use a projector and a camera of high quality. If the sensitivity of the camera is low, image noise will cause the depth map to be noisy. If the dynamic range of the projector is not high, the number of projector columns would be larger than the number of brightness levels the projector can produce. In this case, adjacent projector columns would be forced to project the same brightness.



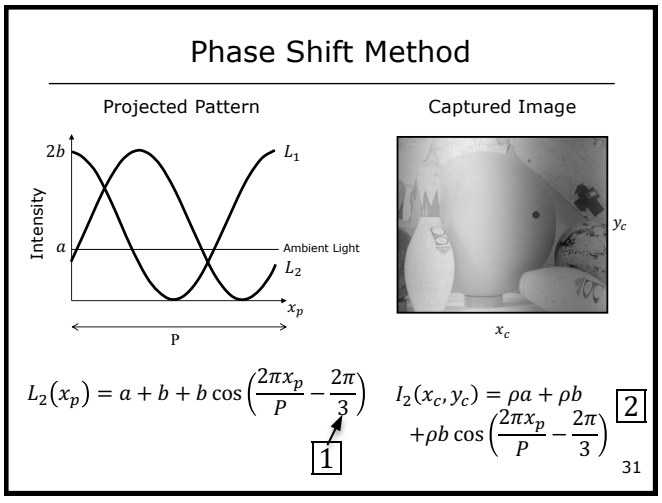
Phase shifting is another method that uses continuous functions and is very widely used in industry. Instead of designing multiple functions, the idea is to capture a set of images by simply changing the phase of one function. Most often the cosine function is used in this method. On the left is shown a cosine function with respect to the projector column x_p. It has a period P, an amplitude b, and an offset which is also b.



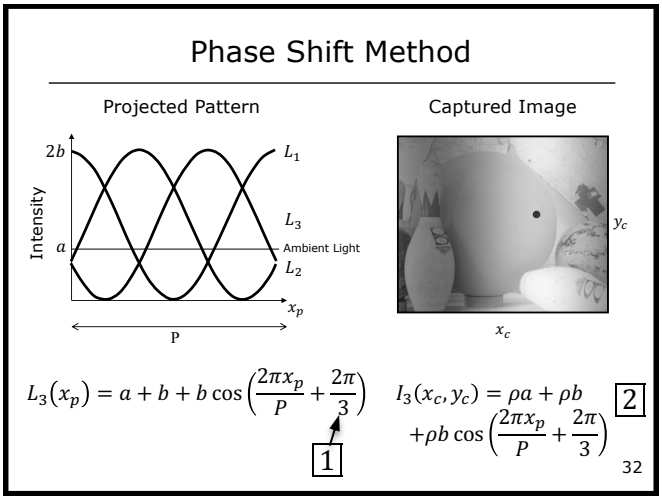
Let us assume that the scene is also lit by ambient illumination, which could vary over the scene and is unknown to us. Let a represent the ambient light that falls on a scene point, which we will treat as an unknown. Thus, the illumination $L_1(x_p)$ of a scene point due to the projector column x_p can be expressed as the sum of the ambient light a, the offset of the cosine function b, and the amplitude b times the cosine of $2\pi x_p$ divided by the period P (equation 1).

On the right is shown the camera image for a scene lit by the above illumination pattern. The intensity at an image pixel (x_c, y_c) is represented by $I_1(x_c, y_c)$. This intensity is equal to the albedo ρ times the illumination $L_1(x_p)$ 2. As discussed earlier, ρ depends on the BRDF of the scene point, its surface normal, and the gain of the camera. Our goal is to find x_p, but we have three unknowns in equation 2—ρa, ρb, and x_p.

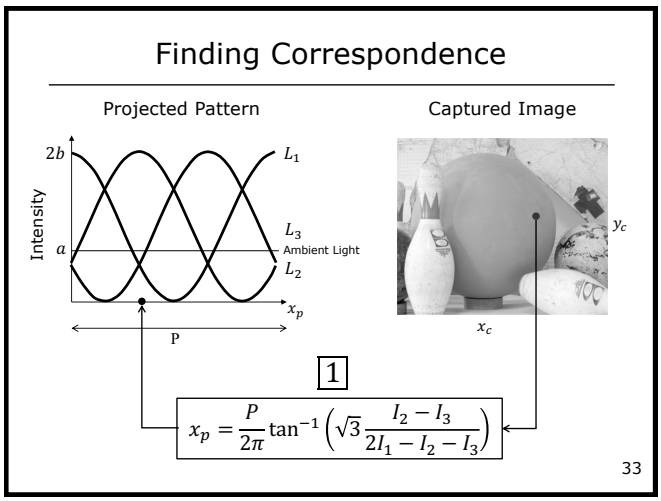
To find x_p, we will capture a second image with a second illumination pattern, which is the original cosine function shifted by $-\frac{2\pi}{3}$. Thus, the only difference between the equations for the illumination $L_1(x_p)$ and the illumination $L_2(x_p)$ is the phase shift 1. Consequently, we get another equation, 2, for the second measured intensity $I_2(x_c, y_c)$ at our point of interest.



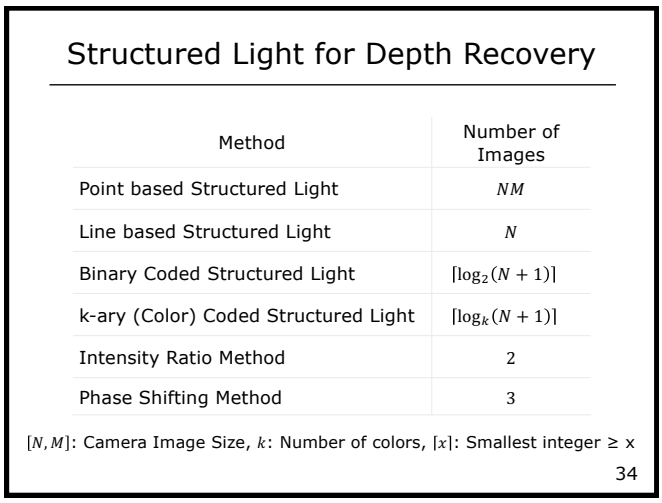
We now capture a third image using a third illumination pattern $L_3(x_p)$ that is obtained by shifting the first pattern $L_1(x_p)$ by $+\frac{2\pi}{3}$ (see 1). Now, we can get a third equation, 2, for the third measured intensity $I_3(x_c, y_c)$ at our point of interest.



We have captured three images corresponding to three phase shifted patterns. For each point in the scene, we have three equations and three unknowns (ρa, ρb, and x_p). Using the three equations, we can find a unique solution for x_p (see 1). Since x_p represents the projector column illuminating the scene, the (x, y, z) coordinates of the point can be found as the intersection of the plane that corresponds to the projector column and the camera line of sight that corresponds to the image point.



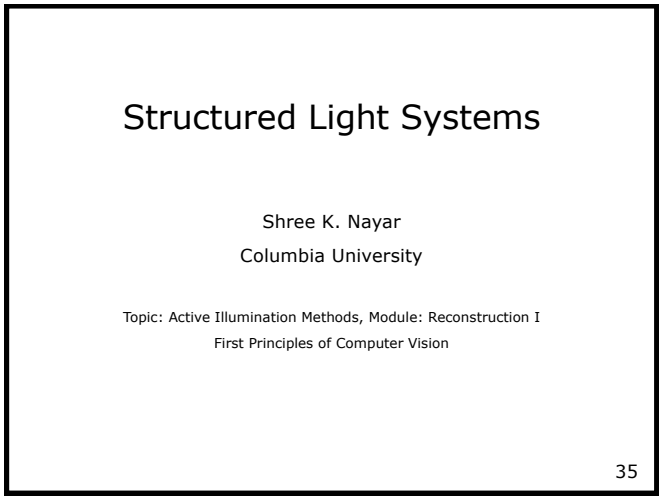
We have discussed several techniques for recovering the 3D structure of a scene using structured light. Let us discuss how these techniques compare to one another. Assume the depth map we are trying to compute has size N times M pixels, where N is the number of columns. Since the point-based method requires the scene points to be lit in sequence, it will require N times M images. In the case of the line-based method, we need to generate N stripes, which means we need N images. Binary coded structured light requires the ceiling of $\log_2(N+1)$ images. The ceiling function is the smallest integer greater than or equal to its input.



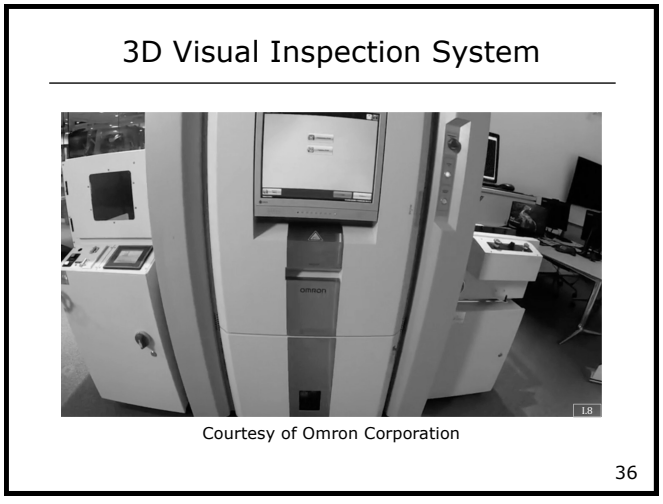
When using multiple levels of intensity (or color), we have a k-ary system rather than the binary system. The k-ary coded structured light method requires the ceiling of $\log_k(N+1)$ images, which is a significant drop in the number of images. In the case of the intensity ratio method, we require only two images, and the phase shifting method requires three images.

While the intensity ratio method is the most efficient, it is very sensitive to image noise and projector quantization. In both the intensity based and phase shifting methods, since they depend on the precise values of measured intensities, we need to ensure the camera and the projector are well calibrated in terms of their radiometric properties. This is one disadvantage of using continuous functions as opposed to light patterns that use a discrete number of brightness values, or colors.

Today, structured light systems are widely used for many different applications and represent a multibillion dollar industry. The design and implementation of these systems is highly advanced in terms of the positioning, lighting, and imaging systems they use. They play a crucial role in manufacturing and inspection, and without them we would not have most of the electronic devices we use on a daily basis. We will now discuss a few examples of structured light systems that represent the state of the art.

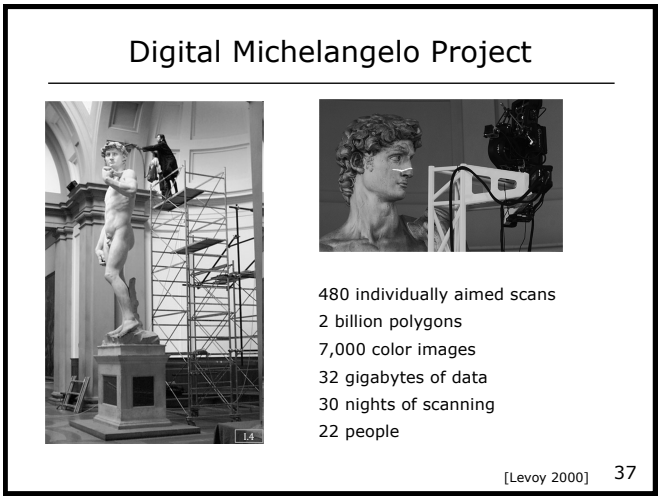


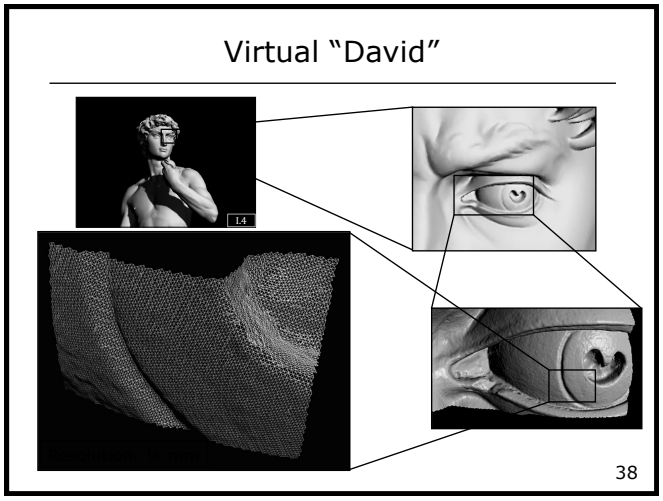
Shown here is a system that has been developed for 3D visual inspection by Omron Corporation. Such systems are large in size as they automatically take in an entire printed circuit board, precisely position it, and inspected it in real-time. By the time the circuit board comes out of this system, we have a complete 3D model of the board, and the system has inspected it in minute detail.



The right side of the system consists of a feeder, into which the printed circuit board is fed. A structured light system consisting of several cameras and controlled light sources is used to apply phase shifting. To obtain 3D structure at a high resolution, the area of the circuit board is partitioned into many small tiles and phase shifting is applied

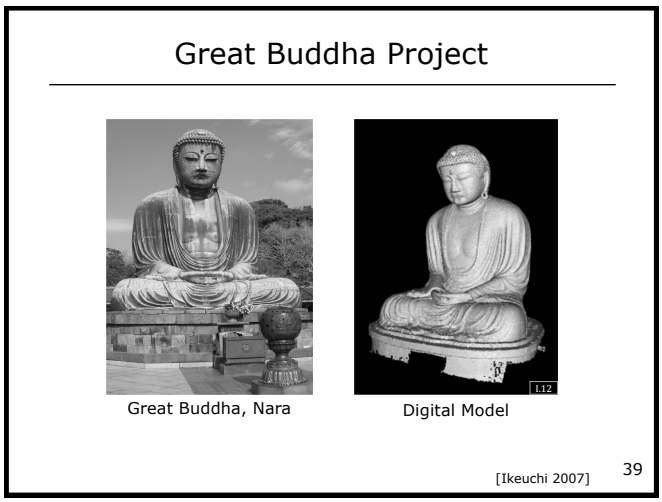
to the tiles in sequence. In a matter of seconds, the 3D structure of the entire circuit board is obtained. When a defect is found, the board is channeled through a different part of the production pipeline so that it can be further inspected.



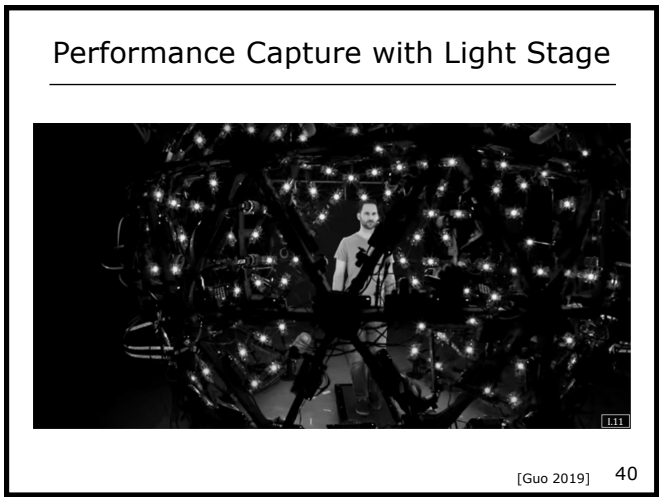


Shown here are results from the Digital Michelangelo Project which was done in 2000. A team of 22 engineers and researchers led by Marc Levoy used a variety of structured light methods to scan well known statues and various artifacts in Italy. The Virtual David, shown on the right, is a precise model obtained by applying structured light to David's face. The depth of each point on the statue was measured with an accuracy of one-fourth of a millimeter. Shown here are zoomed-in views of an eye which reveal the accuracy of the model. The models scanned by the team were used to create a virtual museum.

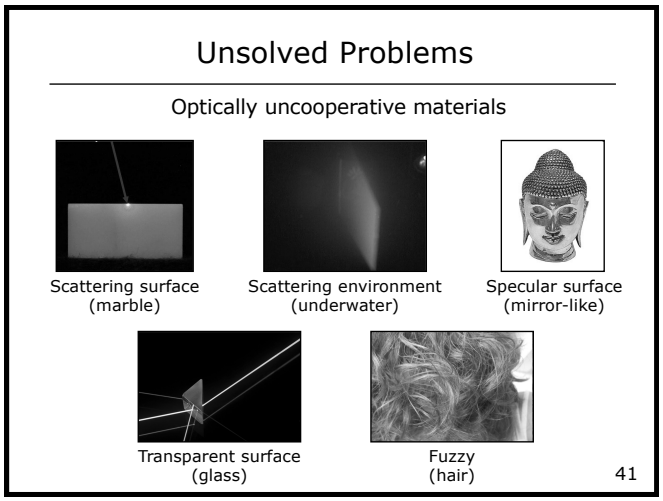
With goals similar to the Digital Michaelangelo Project, Katsushi Ikeuchi led an effort called the Great Buddha Project. His team was able to create digital models of various monuments in Japan, Cambodia, and other sites. Shown here is the Great Buddha in Nara alongside its scanned digital model. These projects are challenging as they require the use of many different structured light techniques. In the Great Buddha Project, the size of the statue is so large that parts of it needed to be scanned using drones with structured light scanners mounted on them.



In slide 8, we described the light stage developed by Paul Debevec and his team at USC. Shown here is another version of the light stage developed by the same team that uses several structured light techniques. The stage uses a large array of LED light sources that can be controlled in terms of color and brightness. In addition, it includes an array of color and infrared cameras. Textured illumination patterns in the infrared domain are projected by the system and the images captured by the infrared cameras are used to estimate detailed 3D structure. In addition, the diffuse and specular reflectance components are estimated at each scene point. The high-speed capture of this system enables it to recover the geometry and reflectance of a dynamic scene (such a walking person). Such a computed model can then be dropped into another environment with different lighting. Performance captures done using the light stage have been used in several Hollywood movies.



Let us now discuss some of the unsolved problems related to structured light. When an object is mirror-like (top, right), structured light methods produce incomplete depth maps since a specular surface point may not be oriented appropriately to reflect the projected illumination in the direction of the camera. Another challenge is posed by objects that are not opaque (top, left). In this case, much of the light striking a point can enter the surface and remerge at neighboring points. This phenomenon is called subsurface scattering and it can make it difficult to determine whether a surface point is bright because it is lit by a source, or because it receives light from neighboring points.



Another situation wherein structured lighting has difficulty is when the object of interest is in a participating medium, such as fog or murky water (top, middle). In such cases, a structured light pattern projected on to the object is attenuated as the light travels through the medium. In addition, the medium itself may glows as it scatters some of the light back in the direction of the camera. For these reasons, it is particularly hard to use structured light methods underwater.

In the case of completely transparent objects (bottom, left), the projected pattern simply passes through the object. The light may undergo multiple refractions but is unlikely to make its way back to the camera. Finally, recovery of fine-scale structures, such as human hair (bottom, right), is a hard problem. The projection of one strand of hair onto the image can be much smaller than the size of a pixel—there could be multiple strands of hair running through a single image pixel. To make a human appear realistic in a special effect, his or her hair needs to recovered with high fidelity. This remains a challenge for structured light methods.

The last active illumination technique we will discuss is the time-of-flight (ToF) method. The idea behind this method is that, if we can measure the total time it takes light to travel from a source to a scene point and then to a sensor, then we can use the known speed of light to compute the total distance traveled by light. This distance is related to the depth of the scene point.

Time of Flight Method

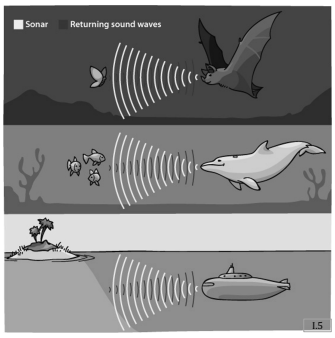
Shree K. Nayar
Columbia University

Topic: Active Illumination Methods, Module: Reconstruction I
First Principles of Computer Vision

42

Time-of-flight is an old idea that is extensively used in nature, especially in the domain of sound. Animals such as bats, hedgehogs and whales throw out calls and then listen to the echoes of these calls to understand where things are in their 3D surroundings. This method is called echolocation, or biosonar, and for many animals it is their primary means of perceiving their environment. It is also used by man-made systems, such as submarines, to figure out how far land or a ship on the surface of the water may be.

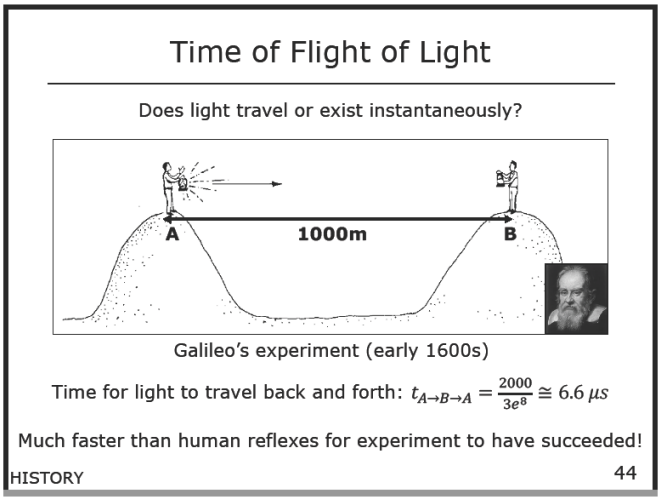
Time of Flight



Echolocation
Measuring Time of Flight of Sound Waves

43

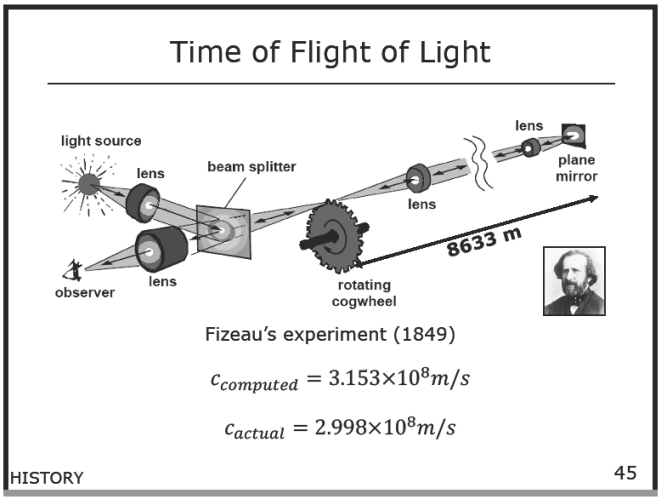
We want to use time-of-flight in the context of light. To measure the time-of-flight of light, we need to know the speed of light. We will discuss two early experiments done to measure the speed of light. These experiments are fascinating given the periods of history in which they were conducted.



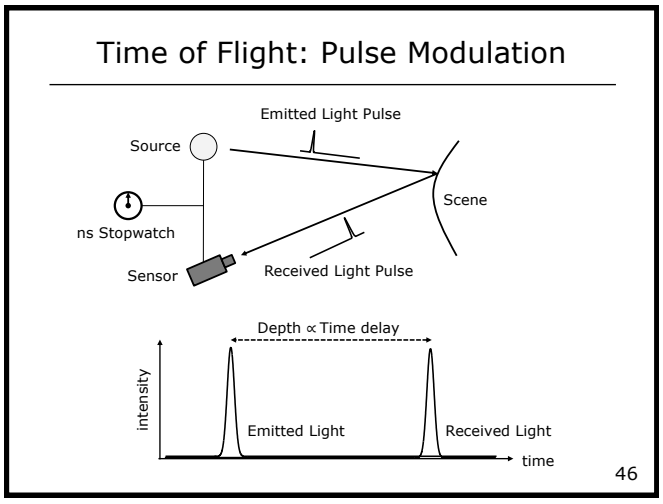
Shown here is the first experiment done by Galileo in the early 17th century. There are two people, A and B, separated by 1000 meters. The idea was to measure the speed of light by having person A open their lantern by releasing a shutter and letting it emit light. As soon as person B sees the light, they would open their shutter, and person A would measure the time it took to see this light. In this way, person A would measure the total time it took light to travel from A to B and back, which was 2000 meters. It is interesting to note that Galileo made use of his pulse to measure this time.

Given what we now know about the speed of light, it takes light 6.6 microseconds to travel a distance of 2000 meters, which is much smaller than the physical reaction time it would take a person to open a shutter. In other words, human reflexes are just too slow for Galileo to have made a meaningful measurement.

Two hundred years after Galileo's attempt, French physicist Fizeau constructed a remarkable experiment to measure the speed of light. Shown here is his setup, which includes an observer looking through a series of lenses at a distant plane mirror. A light source is used with a beam splitter (a half-mirror) to illuminate the plane mirror. Now, consider what happens when the cogwheel is rotated. Light from the source will pass through the cogwheel as pulses—an opening in the cogwheel allows light to pass through and a tooth blocks the light. For some speed of the cogwheel a pulse that leaves an opening, makes its way to the mirror, gets reflected by it, and is blocked by a tooth on the cogwheel. At this speed of the cogwheel, all the pulses leaving it get blocked upon return and hence the observer does not see any light. Using this cogwheel speed, and other dimensions in the experiment, Fizeau computed the speed of light to be $3.153 \times 10^8 m/s$. This is a truly impressive estimate as the actual speed of light is now known to be $2.998 \times 10^8 m/s$!

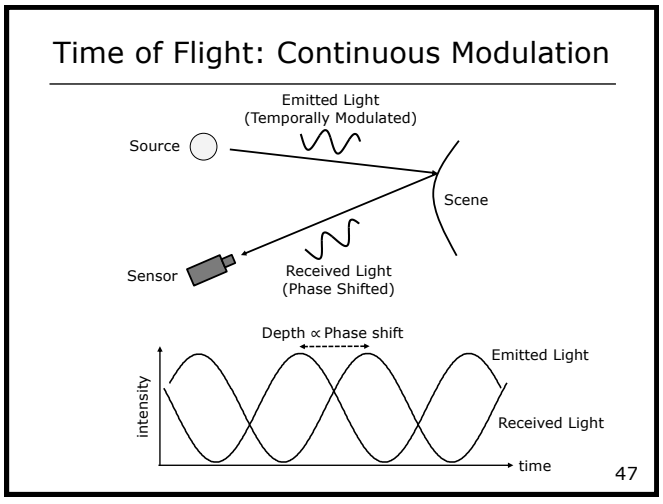


The first technique we will discuss for measuring the distance of scene points using the time-of-flight of light is called pulse modulation. As shown here, a short but bright flash of light (pulse) is emitted by a source. This light pulse travels at the speed of light, hits a scene, and gets reflected by the scene in many directions. The sensor receives the light pulse after reflection by a single scene point that lies on its line of sight. The time delay between the emitted pulse and the received pulse can be used with the known speed of light to compute the total distance traveled by the pulse. This distance is the sum of the distances from the source to the scene point and from the scene point to the sensor. This total distance and the line of sight of the sensor uniquely determine the depth of the scene point from the sensor.



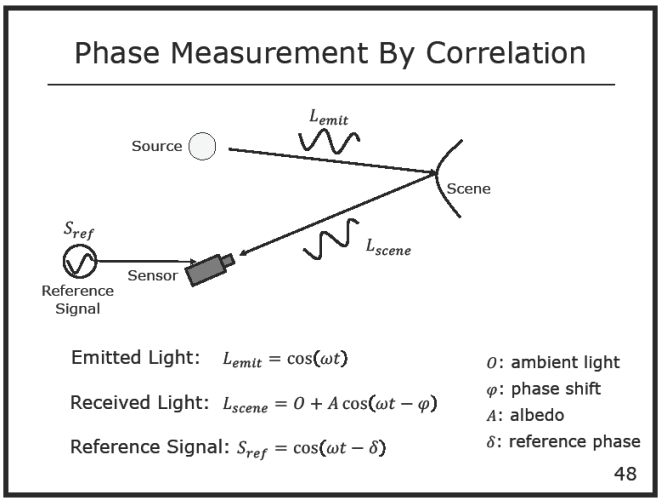
To measure the time delay between the emitted and received pulses, a stopwatch capable of measuring pulses with nanosecond accuracy is used. For the above method to work well, we need the emitted pulse to be infinitesimal in width and very bright to be clearly detectable after reflection.

The second approach to measuring depth using time-of-flight is called continuous modulation. In this method, the source emits light such that the intensity (brightness) of the light is temporally modulated. This emitted light is being shown here as a wave, but the modulation has nothing to do with the frequency, or wavelength, of the light. The sensor receives the temporally modulated light after reflection by a single scene point that lies on its line of sight.



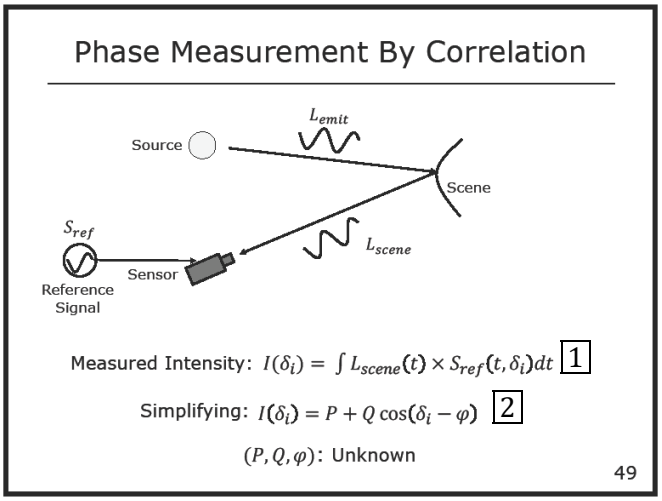
In the plot shown below, the emitted and the received light intensities are plotted as a function of time. The phase difference between the two waves determines the delay, or the time taken by light to get from the source to the sensor. This time and the speed of light can be used to determine the distance that light has traveled. As before, this distance uniquely determines the depth of the scene point from the sensor.

To measure the phase difference between the emitted and received light, we will use a correlation-based approach. Let us represent the emitted light L_{emit} as $\cos(\omega t)$, where ω refers to the frequency of the source's intensity modulation. The received light L_{scene} can be written as O plus A times $\cos(\omega t - \varphi)$, where A is the attenuation of the light due to reflection by the scene point, φ is the phase shift, and the constant O accounts for the unknown ambient illumination of the scene point.



We now introduce a reference signal S_{ref} which will be correlated by the sensor with the received light L_{scene}. The reference signal S_{ref} has the same frequency ω as the received light L_{scene} and hence can be written as $\cos(\omega t - \delta)$, where δ refers to the reference phase which is controlled by the sensor. This correlation can be implemented in the electronics of the sensor by, in effect, varying the gain of the sensor as a function of time, during the exposure time of the sensor.

The measured intensity $I(\delta_i)$ at a pixel is then the incoming light L_{scene} multiplied with the reference signal S_{ref} with phase δ_i, and integrated over the exposure time of the sensor [1]. The exposure time could, for instance, be 30 milliseconds, which is substantially greater than the period of the source's brightness modulation. Upon simplifying [1], we get the expression [2] for the measured intensity $I(\delta_i)$. In [2], we know the reference signal phase δ_i as it is controlled by the sensor. So, we have one equation with three unknowns—the constants P and Q, and the phase shift φ.



To solve for the above three unknowns, we use three different phases, δ_1, δ_2 and δ_3, and their corresponding intensity measurements $I(\delta_1)$, $I(\delta_2)$, and $I(\delta_3)$. To find the distance traveled by light from the source to the sensor, we need to convert the computed phase shift φ into a travel time, which can be done by dividing φ by $4\pi f$, where f is $\omega/2\pi$. Then, the total distance traveled from the source to the sensor is the travel time times the speed of light, c. If the source and sensor are placed close to each other, we can assume that the scene point is equidistant from both of them. Therefore, the depth of the scene point from the sensor is the above travel distance divided by two (equation 1). For example, if we have a phase difference φ of π, and a frequency of modulation f equal to 30 megahertz, we get a scene depth d of 2.5 meters. A major advantage of the time-of-flight method over structured light techniques is that we can measure depth with great accuracy, even for large depths.

Computing Depth From Phase

$$I(\delta_i) = P + Q\cos(\delta_i - \varphi)$$

Three measurements $I(\delta_1)$, $I(\delta_2)$, $I(\delta_3)$ are sufficient for computing phase φ.

Depth of scene point:

$$d = c\frac{\varphi}{4\pi f} \quad (1)$$

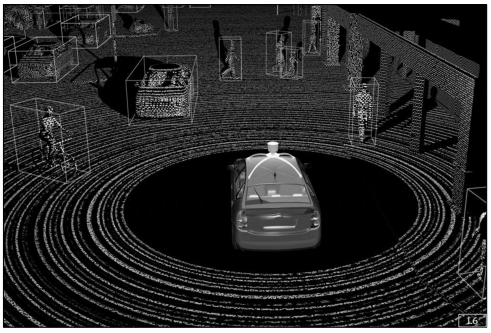
c: speed of light
φ: phase shift
$f = \omega/2\pi$: modulation frequency

Example: If $\varphi = \pi, f = 30MHz,$ then $d = 2.5m.$

50

Shown here is an example of a depth map (shown as a point cloud) recovered by a time-of-flight camera mounted on a driverless car. As can be seen, the depth map is extremely detailed, enabling the car to recognize and accurately estimate the depths of distant cars and pedestrians.

Time-of-Flight Result: Example



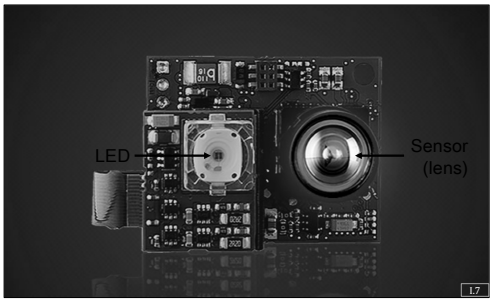
Google Self-Driving Car Project

51

A typical time-of-flight system used in a driverless car does not measure the depth of the entire scene in one shot. Instead, the time-of-flight sensor is mechanically scanned to get a complete 360-degree depth map like the one shown here. One of the challenges for driverless cars is to come up with affordable time-of-flight systems. Until recently, the cost of a time-of-flight sensor was greater than the cost of a typical car. A significant effort is underway in industry to find ways to dramatically reduce the cost of time-of-flight sensors without compromising their accuracy.

For measuring depths within a shorter range, there are commercial cameras that use an array of ToF sensors to measure a complete depth map without scanning. Shown here is such a 3D camera from PMD Technologies which consists of an LED that can emit a flash (pulse modulation) of light, or amplitude modulated light (continuous modulation). The camera has an image sensor that has a two-dimensional array of pixels, where each pixel can measure depth along its line of sight. Such sensors have also started making their way into mobile devices like smartphones and tablets. In the coming decade, time-of-flight cameras are expected to become ubiquitous.

Time of Flight: 3D Camera



PMD Technologies

52

References and Credits

Shree K. Nayar

Columbia University

Topic: Active Illumination Methods, Module: Reconstruction I

First Principles of Computer Vision

53

References

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

[Caspi 1998] D. Caspi, N. Kiryati, and J. Shamir. *Range imaging with adaptive color structured light*. IEEE Trans. on PAMI, 20(5), 1998, pp.470-480.

[Guo 2019] K. Guo, P. Lincoln, P. Davidson, J. Busch, X. Yu, M. Whalen, G. Harvey, S. Orts-Escolano, R. Pandey, J. Dourgarian, M. DuVall, D. Tang, A. Tkach, A. Kowdle, E. Cooper, M. Dou, S. Fanello, G. Fyffe, C. Rhemann, J. Taylor, P. Debevec, and S. Izadi. *The Relightables: Volumetric Performance Capture of Humans with Realistic Relighting*. ACM Transactions on Graphics, November 2019.

[Ikeuchi 2007] K. Ikeuchi, T. Oishi, J. Takamatsu, R. Sagawa, A. Nakazawa, R. Kurazume, K. Nishino, M. Kamakura, Y. Okamoto. *The Great Buddha Project: Digitally Archiving, Restoring, and Analyzing Cultural Heritage Objects.* International Journal of Computer Vision, 75(1), 2007.

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

[Levoy 2000] M. Levoy, K.Pulli, B. Curless, S.Rusinkiewicz, D. Koller, L. Pereira, Lucas, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, D. Fulk. *The Digital Michelangelo Project: 3D Scanning of Large Statues*. ACM SIGGRAPH, 2000.

54

References

[Nayar 1989] S.K. Nayar, K. Ikeuchi and T. Kanade. *Shape and Reflectance from an Image Sequence Generated using Extended Sources.* IEEE International Conference on Robotics and Automation, May. 1989.

[Posdamer 1981] J. L. Posdamer and M. D. Altschuler. *Surface measurement by space-encoded projected beam systems*. Computer Graphics and Image Processing, 18(1), pp.1-17. 1981.

[Salvi 2004] J. Salvi, J. Pages, and J. Batlle. *Pattern codification strategies in structured light systems.* Pattern Recognition, vol.37, no.4, 2004, pp.827-849.

[Sanderson 1988] A.C. Sanderson, L.E. Weiss and S.K. Nayar. *Structured Highlight Inspection of Specular Surfaces.* IEEE PAMI, Vol. 10, No. 1, Jan. 1988.

[Wenger 2000] A. Wenger, A. Gardner, C. Tchou, J. Unger, T. Hawkins, P. Debevec. *Performance Relighting and Reflectance Transformation with Time-Multiplexed Illumination*. ACM SIGGRAPH, 2005.

[Woodham 1980] R. Woodham. *Photometric Method for Determining Surface Orientation from Multiple Images.* Optical Engineering, 1980.

[Wust 1991] C. Wust and D. W. Capson. *Surface profile measurement using color fringe projection*. Machine Vision and Applications, vol.4, 1991, pp.193-203.

55

Image Credits

I.1 Daniel Scharstein. Used with permission.

I.3 Paul Debevec. Used with permission.

I.4 Marc Levoy. Used with permission.

I.5 https://askabiologist.asu.edu/sites/default/files/resources/articles/bats/ecolocation_types.jpg Arizona Board of Regents / ASU Ask a Biologist. Licensed under CC BY-SA 3.0.

I.6 http://www.popsci.com/cars/article/2013-09/google-self-driving-car. Popular Science, 2013.

I.7 http://www.pmdtec.com/html/img/camboard_nano.jpg. PMD Technologies.

I.8 Omron Corporation. Used with permission.

I.9 Paul Debevec. Used with permission.

I.10 Paul Debevec. Used with permission.

I.11 Paul Debevec. Used with permission.

I.12 Katsushi Ikeuchi. Used with permission.

56

Acknowledgements: Thanks to Roshan Kenia, Ayush Sharma and Nikhil Nanda for their help with transcription, editing and proofreading.

References

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

[Caspi 1998] D. Caspi, N. Kiryati, and J. Shamir, Range imaging with adaptive color structured light, IEEE Trans. on PAMI, 20(5), pp.470-480, 1998.

[Guo 2019] K. Guo, P. Lincoln, P. Davidson, J. Busch, X. Yu, M. Whalen, G. Harvey, S. Orts-Escolano, R. Pandey, J. Dourgarian, M. DuVall, D. Tang, A. Tkach, A. Kowdle, E. Cooper, M. Dou, S. Fanello, G. Fyffe, C. Rhemann, J. Taylor, P. Debevec, and S. Izadi, The Relightables: Volumetric Performance Capture of Humans with Realistic Relighting, ACM Transactions on Graphics, November 2019.

[Ikeuchi 2007] K. Ikeuchi, T. Oishi, J. Takamatsu, R. Sagawa, A. Nakazawa, R. Kurazume, K. Nishino, M. Kamakura, and Y. Okamoto, The Great Buddha Project: Digitally Archiving, Restoring, and Analyzing Cultural Heritage Objects, International Journal of Computer Vision, 75(1), 2007.

[Inokuchi 1984] S. Inokuchi, K. Sato, and F. Matsuda, Range imaging system for 3-D object recognition, ICPR, pp.806-808, 1984.

[Levoy 2000] M. Levoy, K.Pulli, B. Curless, S.Rusinkiewicz, D. Koller, L. Pereira, Lucas, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk, The Digital Michelangelo Project: 3D Scanning of Large Statues, ACM SIGGRAPH, 2000.

[Nayar 1989] S.K. Nayar, K. Ikeuchi, and T. Kanade, Shape and Reflectance from an Image Sequence Generated using Extended Sources. IEEE International Conference on Robotics and Automation, May. 1989.

[Posdamer 1981] J. L. Posdamer and M. D. Altschuler, Surface measurement by space-encoded projected beam systems, Computer Graphics and Image Processing, 18(1), pp.1-17, 1981.

[Salvi 2004] J. Salvi, J. Pages, and J. Batlle, Pattern codification strategies in structured light systems, Pattern Recognition, Vol.37, No.4, pp.827-849, 2004.

[Sanderson 1988] A.C. Sanderson, L.E. Weiss, and S.K. Nayar, Structured Highlight Inspection of Specular Surfaces, IEEE PAMI, Vol. 10, No. 1, Jan. 1988.

[Wenger 2000] A. Wenger, A. Gardner, C. Tchou, J. Unger, T. Hawkins, and P. Debevec, Performance Relighting and Reflectance Transformation with Time-Multiplexed Illumination, ACM SIGGRAPH, 2005.

[Woodham 1980] R. Woodham, Photometric Method for Determining Surface Orientation from Multiple Images, Optical Engineering, 1980.

[Wust 1991] C. Wust and D. W. Capson, Surface profile measurement using color fringe projection, Machine Vision and Applications, Vol.4, pp.193-203, 1991.