Chapter 1
3D Depth Cameras in Vision: Benefits and Limitations of the Hardware

With an Emphasis on the First- and Second-Generation Kinect Models

Achuta Kadambi, Ayush Bhandari and Ramesh Raskar

Abstract The second-generation Microsoft Kinect uses time-of-flight technology, while the first-generation Kinect uses structured light technology. This raises the question whether one of these technologies is "better" than the other. In this chapter, readers will find an overview of 3D camera technology and the artifacts that occur in depth maps.

1.1 Introduction

The physical world is three-dimensional. It is surprising then that conventional cameras perceive the world in two dimensions, while even primitive animals perceive the world in its richer three-dimensional form. Today, using technology such as Microsoft Kinect, it is possible to collect 3D data with the ease and cost of point-and-shoot photography.

In today's consumer landscape, there are two primary 3D camera technologies: **structured light** and **time of flight**. A structured light camera projects an active pattern and obtains depth by analyzing the deformation of the pattern. The **first-generation Kinect** (2010) is a structured light camera. In contrast, a time-of-flight

We thank the following people at the Massachusetts Institute of Technology for their contributions to the chapter: Nikhil Naik, Boxin Shi, Ameya Joshi, Genzhi Ye, Amol Mahurkar, Julio Estrada, Hisham Bedri, and Rohan Puri.

A. Kadambi (✉) · A. Bhandari · R. Raskar
Massachusetts Institute of Technology, Cambridge, MA, USA
e-mail: achoo@mit.edu

A. Bhandari
e-mail: ayush@mit.edu

R. Raskar
e-mail: raskar@mit.edu

L. Shao et al. (eds.), *Computer Vision and Machine Learning with RGB-D Sensors*,
Advances in Computer Vision and Pattern Recognition,
DOI: 10.1007/978-3-319-08651-4_1

camera measures the time that light has been in flight to estimate distance. The **second-generation Kinect** (2013) is a time-of-flight camera. Microsoft's strategy for moving to time of flight for their second Kinect raises the question whether one of these technologies is "better" than the other.

By the end of this chapter, we expect that the reader will have a better understanding of the benefits and limitations of different 3D camera technologies. More importantly, readers will learn the underlying cause behind depth artifacts and the accepted strategies to fix them.

1.2 3D Data Storage

1.2.1 3D Point Clouds

A **point cloud** is set of data points defined in a coordinate system. Using M to refer to the number of points and N for the dimensionality of the space, the point cloud is written as follows:

$$P = \{p_1, \ldots, p_M\} \quad p^T \in \mathrm{R}^N. \tag{1.1}$$

A common technique to store point clouds in memory is to form an by array denoted as P where each row vector in corresponds to a point. We require two conditions to define an N-dimensional point cloud:

1. $p_i^T \in \mathrm{R}^N \quad i = 1, \ldots, M.$
2. The object of interest is in the convex hull of the points.

The first condition is satisfied with any camera that acquires depth. The second condition is more restrictive and is discussed in detail in the 3D scanning section (Sect. 1.9). In this chapter, we are particularly interested in 3D point clouds, i.e., the set of points defined by real-world coordinates X, Y, and Z.

Often we are provided additional information for each point in the cloud. Suppose we are provided a K-tuple of "information" for each point in the cloud. This is described as the following set:

$$F = \{f_1, \ldots, f_M\} \quad f^T \in \mathrm{R}^K. \tag{1.2}$$

which is stored in memory as an M by K array. Taken together, the feature array $\boldsymbol{F}$ and the corresponding point cloud array $\boldsymbol{P}$ represent the space of data we consider in this chapter. As a concrete example, if we are provided the 3-tuple of the color reflectance at each point in R^3, we have color and depth. This is known as **RGB-D acquisition**. For simplicity, we refer to the point set as the "depth data" and the feature set as the "RGB data."

1.2.1.1 A Note on Organized Point Clouds

Point clouds from depth cameras, such as Microsoft Kinect, can be further classified into **organized point clouds**. In such point clouds, it is possible to index the spatial X and Y coordinates in a logical manner, e.g., by rows and columns. In contrast, unorganized point clouds have no structure in the spatial coordinates. Organized point clouds are a much more powerful data format that greatly facilitates registration and other types of vision algorithms.

1.2.2 Registering Color and Depth Data

We must have correspondence between the depth and color data. This would be a simple task if the depth and color maps were acquired with the same camera. In practice, when the cameras are separate, the additional step of registration must be performed on the depth and color maps. Depth registration error is a practical phenomenon that can occur spatially, when shadows or occlusions are present, or temporally, when camera frame rates are not synchronized.

For many computer vision researchers, the registration error may not be significant. However, this may not be true when specific correspondences between depth and color edges are required. To ensure proper calibration between depth and color calibration, note that classic checkerboard calibration [35] is imperfect for depth maps (there are no edges). Zhang and Zhang propose a maximum-likelihood solution that is able to take into account uncertainty in depth values (Fig. 1.1).

1.2.2.1 Sparse Versus Dense Depth Maps

In many modalities, we obtain a point cloud that is **sparse**. In such a case, depth values are provided for only a small subset of coordinates. This commonly occurs

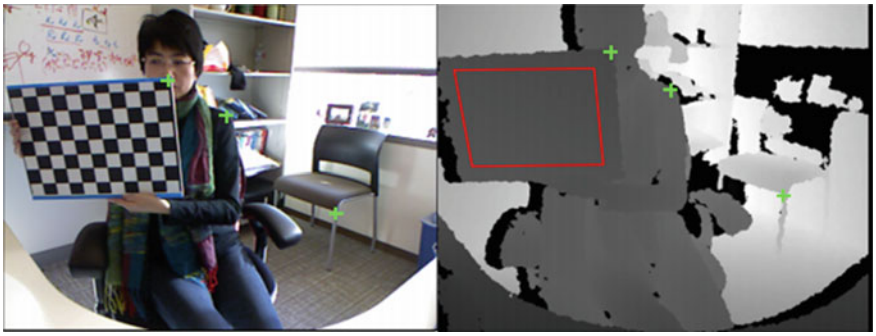


Fig. 1.1 Registering RGB and depth images for commodity depth cameras using a checkerboard. At left is the color image and at right is the depth image [34]

in the case of stereo computer vision where it is challenging to find matches for all points in typical scenes. It can also occur in other modalities to suppress noise, i.e., when many points with a low confidence are redacted. In contrast, a **dense** point cloud has depth for almost every real-world coordinate within the scope of the imaging device. For many applications, it may be desirable to have a few high-confidence 3D coordinates. As a case in point, many finger-tracking algorithms use a sparse depth map to associate hand positions with gestures.

1.3 Universal Artifacts

In this section, we discuss some example artifacts that are present in the depth maps. These artifacts are somewhat universal as they exist in many classes of consumer 3D camera technology (e.g., structured light, time of flight, and stereo). In Sect. 1.3, we discuss artifacts that are relevant to specific computer vision tasks.

1.3.1 Holes

Depth maps can be generated using various methods, including structured light, stereo correspondence, and time of flight. Regardless of the technique, it is common to find **holes** in the depth map, where depth data are missing. An example of this is presented in Fig. 1.2. Observe that the ground truth depth map has no holes, but the recovered depth (with a stereo camera) has many holes.

1.3.1.1 Fixing Holes in Depth Maps

Fixing holes is an important preprocessing step that, if performed correctly, can improve the quality of the depth map. The typical recipe for hole filling is as follows:

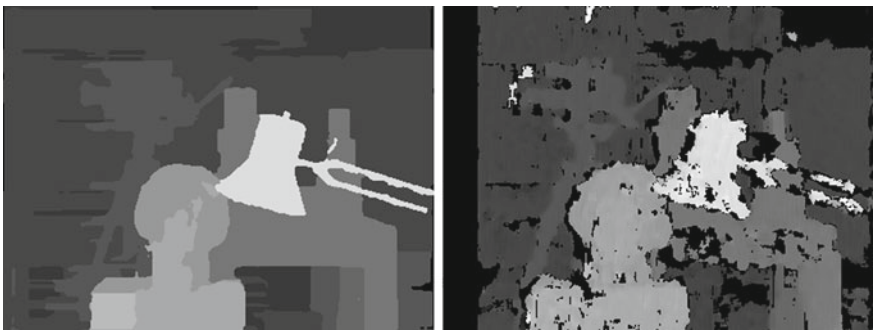


Fig. 1.2 *Left* Ground truth depth map from the Tsukuba dataset. *Right* Measured depth image with a stereo camera. The *black* values are "holes"

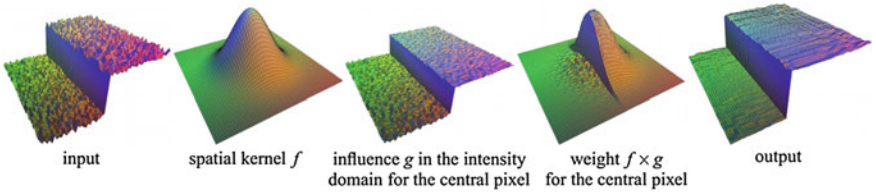


Fig. 1.3 Bilateral filtering. At *left*, the input is noisy but has a clear edge discontinuity. At *right* is the filtered output. Figure from [6]

Fill the holes by exploring correlations between neighboring depth pixels, subject to the constraint that sharp depth edges are preserved.

A reasonable starting point might involve using a median filter with an adaptive window size to correct for the holes. Unfortunately, this method violates the constraint of keeping the depth edges sharp. The accepted techniques can be broadly divided into **filtering methods** or **classification methods**.

1.3.2 Filtering Methods

A popular hole-filling method is to use a modified **bilateral filter**. Readers may be familiar with the bilateral filter from computer graphics [6]. The basic idea is to smooth noise, while preserving sharp edge discontinuities (Fig. 1.3). A bilateral filter can be transposed to operate on 3D point clouds and is often a standard processing step in the API of camera manufacturers.

One extension is to combine a temporal approach with bilateral filtering to achieve the dual goal of preserving edge discontinuities and filling holes [4]. In particular, a joint bilateral filter is used:

$$D_f^p = \frac{H(C_{\text{map}}, \Omega^p)}{k^p} \sum_{q \in \Omega^p} \hat{D}^q f(p,q) h(\| I^p - I^q \|) \tag{1.3}$$

where $H(C_{\text{map}}, \Omega^p)$ is a function that evaluates the reliability of the depth values in the neighborhood Ω^p. This additional function allows the filter weights to be selected by discriminating between pixels that belong to different objects. Figure 1.4 shows the published results of the technique.

Another hole-filling technique combines belief propagation with the additional step of cross-checking two input images to identify the confidence level of a pixel [1]. The depth map is then smoothed by using color information and the confidence labels. In the final step, the authors utilize an anisotropic diffusion technique for disparity map refinement. A key point to note is that in anisotropic diffusion, depth edges are not smoothed. This adaptive smoothing method is called *directed anisotropic diffusion*. Figure 1.5 shows a few of the published results.

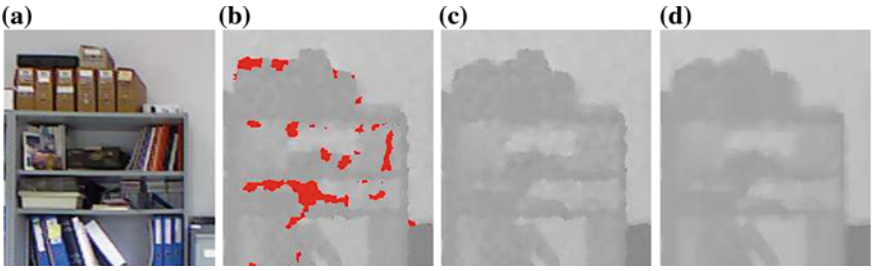


Fig. 1.4 **a** A cluttered scene. **b** Holes in the depth map; observe the co-occurrence of shadows. **c** Intermediate filtering. **d** Filled holes using neighborhood information. Figure from [4]

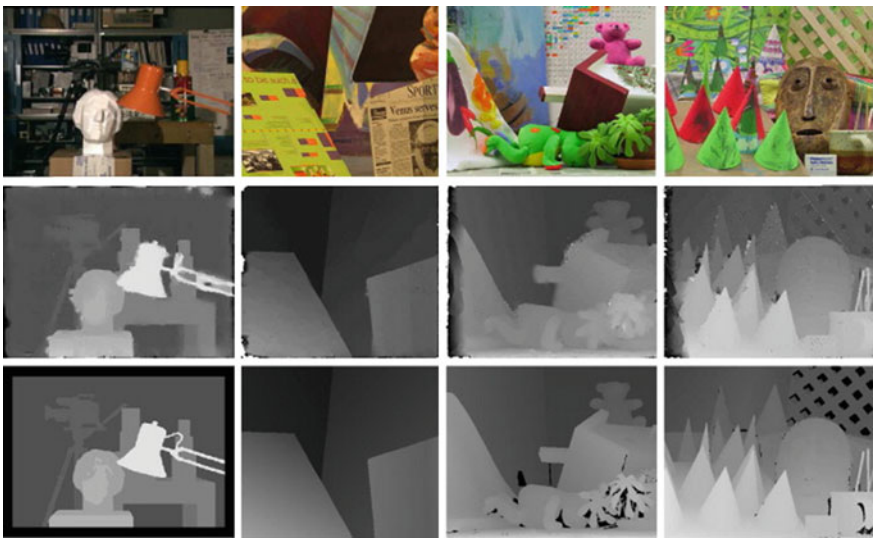


Fig. 1.5 Anisotropic filtering from [1]. The first row consists of reference images, the second row the smoothed depths, and the last row the ground truth

1.3.3 Segmentation/Classification Methods

An alternate strategy to fill holes is to exploit classification information. If two pixels belong to the same object, it is possible that the depth values can be robustly interpolated. As such, color-based segmentation and clustering is an extremely popular hole-filling technique [19, 33]. In [18], a color-based clustering method is combined with change detection to find coherent pixels (Fig. 1.6). The key addition is change detection that allows static objects to be classified together. This result is shown in Fig. 1.7.

1.4 Causes for Holes

Why do holes occur in depth images? The most common reason for holes is **occlusions**. Many 3D cameras require different views of an object to obtain depth. When an occlusion occurs, the object is present in only one view and it is not possible to obtain depth. In a more general sense, any time there is a **matching ambiguity**, holes can be found. Such ambiguities occur in ill-posed matching problems, i.e., when point correspondences cannot be found. Note that because time-of-flight sensors use a single viewpoint to obtain depth, occlusions and matching ambiguities are non-factors. Another source of error occurs at **depth discontinuities** where windowing effects smear the foreground and background depths together. Windowing is a rather broad term and encompasses both hardware-level windowing, e.g., by "flying pixels" in time-of-flight range imaging, and software-level windowing, e.g., by patch-based correspondence.

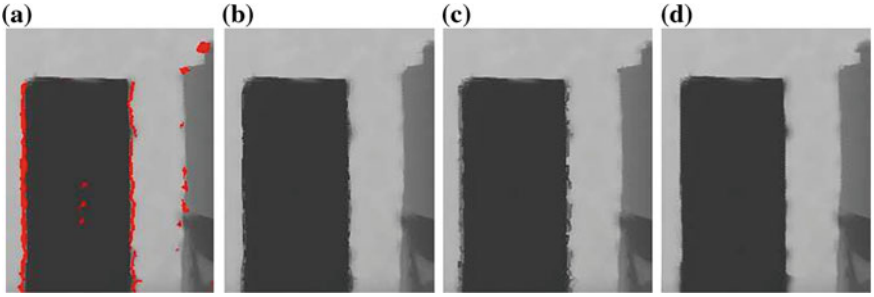


Fig. 1.6 Filling holes on the 2010 Kinect for Xbox 360. **a** Kinect depth map; observe the holes in *red*. **b** A simple median filter. **c** Inpainting algorithms. **d** Strategy from [4]

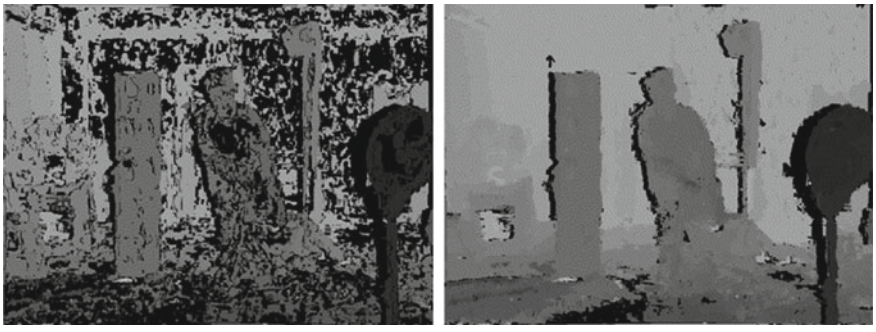


Fig. 1.7 Intensity-based segmentation. *Left* Depth map with many holes. *Right* Processed depth map with the spatiotemporal technique of [18]

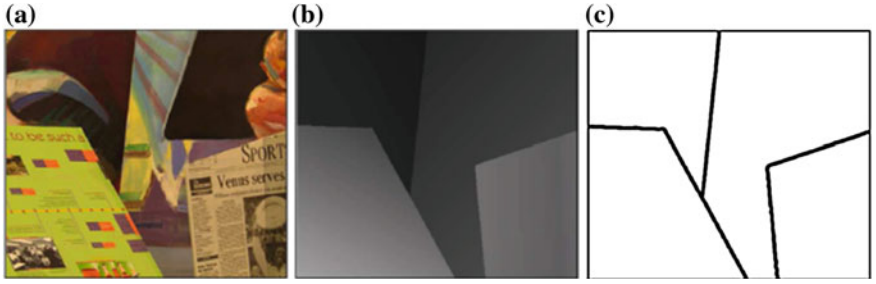


Fig. 1.8 **a** Photograph of a scene. **b** Depth map. **c** Depth edges

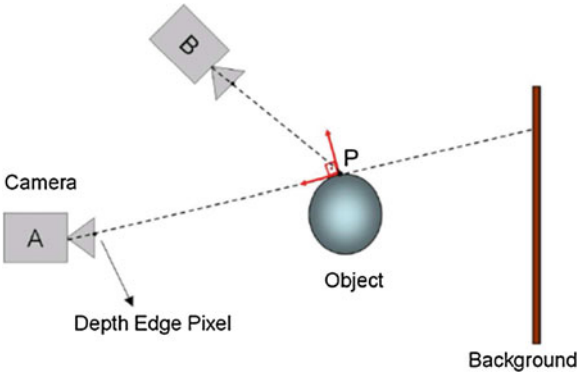


Fig. 1.9 Depth edges are view dependent. The surface point *P* corresponds to an edge in camera *A* but not in camera *B*

1.4.1 Morphological Edge Erosion

Another important artifact in depth images is the lack of sharp edges. Depth discontinuities, also known as depth edges or occluding contours, correspond to sharp changes in a depth map of the scene. An example of a depth map and its depth edges is illustrated in Fig. 1.8. Depth edges, such as intensity edges, form the backbone of several vision algorithms. Thus, it is very desirable to preserve sharp depth edges that correctly map to the scene.

The first thing to note about depth edges is that they are view dependent. As illustrated in Fig. 1.9, depending on the camera viewpoint, surface points may or may not correspond to edges [3]. Parallax-based methods, such as structured light and stereo, use different viewpoints and can therefore lead to smoothing of depth edges. In particular, the tendency of parallax-based methods to use window-based correspondence exacerbates the problem, and recovered edges are often smoothed to an intolerable degree.

Sharpening edges in stereo range images is not new task; we refer to the following classic papers for extensive discussion on the topic: [7, 22, 25]. A reasonable solution is to reduce the patch size (for correspondence) in a carefully chosen manner. For instance, in [14] and [17], the window size is reduced based on the amount of disparity variation. While the edges are indeed sharper, the technique is considered computationally expensive. Another alternative is to rectify the edges in *post-processing* of the depth map. One option is to employ bilateral filtering in a similar fashion to the hole-filling algorithms discussed in Sect. 1.3.1.1. For example, in [5], they combine color data with joint bilateral filtering in the depth images to preserve edges. They demonstrate results on the first-generation Kinect that uses structured light. The idea is as follows: First, using the RGB image, pixels with the wrong depth values are detected and removed with a region growing method. This creates holes, which are now filled with a joint bilateral filter that is carefully chosen in context of the noise properties of Kinect [5].

In short, the basic idea for any type of edge rectification for parallax-based 3D cameras is very similar to the techniques used in hole-filling algorithms. Some key priors include edge correspondence between the color and depth images [5] and color- or motion-based segmentation [18, 33].

In time-of-flight cameras, the situation is a bit different. Since only one viewpoint is used to obtain depth, the edges are often much sharper. We must remark that current generations of ToF sensors have low spatial resolution and this can lead to optical mixing of foreground and background objects [16]. Despite this, depth edges are still better preserved in time-of-flight sensors and the depth edges do not need to be post-processed to the degree of structured light systems. Of course, post-processing can still be applied to improve the acquired depth map. For example, in [31], an algorithm is proposed to find straight edges in the depth image. In [20], a denoising algorithm that implements the total variation image prior is used. The total variation prior minimizes the Manhattan (L1) norm of the gradient and has been shown to promote piecewise smooth functions. The result from [20] is shown in Fig. 1.10.

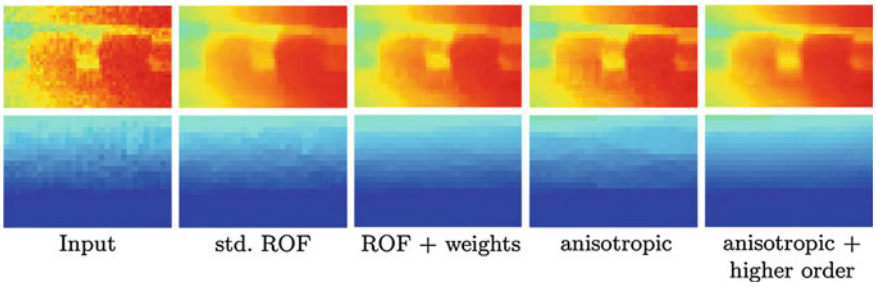


Fig. 1.10 Enhancing edges in time-of-flight imaging. At *left* is the input and at *right* is the anisotropic total variation method from [20]

1.5 Ambient Lighting

Ambient lighting can be a problem for some 3D cameras that use active illumination, such as stereo or structured light sensors. Creating cameras that can operate in ambient lighting fills a practical need. One such example occurs in outdoor robotics, where 3D cameras are trusted to work in sunlight.

Each 3D technology is affected by ambient lighting in different ways. Computer stereovision does not rely upon active illumination and is therefore robust to ambient lighting. In contrast, structured light relies upon a projected intensity pattern to obtain depth. The pattern can become corrupted by ambient light and ambient shadows. We may therefore state that structured light cameras are, in general, not robust to ambient light. Infrared structured light systems, such as the first-generation Kinect, use an optical filter to reject wavelengths that are not in the active illumination, but this helps only marginally (sunlight contains a significant amount of near-infrared light).

The new Kinect uses time of flight to obtain depth. This technique is resistant to ambient light, allowing the new Kinect to be used outdoors. However, when the scene is too bright, the imaging sensor saturates and all information is lost. In outdoor experiments, the time-of-flight Kinect outperforms the old structured light Kinect. For details on why the new Kinect is resistant to ambient light, please refer to Sect. 1.11.

1.6 Motion

The 3D cameras that we consider are all designed to operate on **dynamic scenes** where there is motion. That this capability exists is a testament to strong hardware design and development: Many of the 3D sensors are inherently multishot techniques. Concretely, this means that multiple photographs of the scene (these are called subframes) are required to obtain a single depth frame. If there is substantial motion between the subframes, severe artifacts will be seen in the depth frame.

What is the limit of consumer technology? Qualitatively, a speeding car at 60 mph cannot be captured with an active 3D depth camera (consumer grade), but a person performing an exercise activity can be captured. A separate issue is in synchronization of the optical and imaging elements. In the standard computer vision practice, motion blur has not been, to the best of our knowledge a key issue. For less common applications, such as using 3D time-of-flight sensors for heart motion detection, it is necessary to preprocess the subframes using an algorithm like optical flow [26]. A similar "flowlike" preprocessing step is also described in [11].

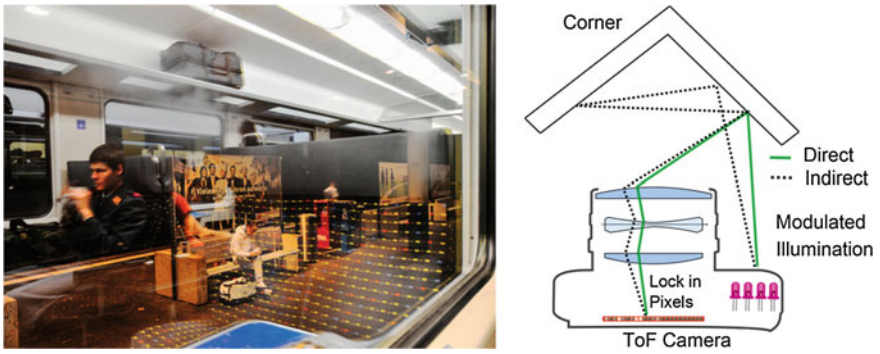


Fig. 1.11 *Left* Example scene with multiple reflections. Photograph courtesy Ayush Bhandari. Shot in a train station in Switzerland. *Right* Multiple interreflections smear together at a corner and provide incorrect depth readings

1.6.1 Multiple Reflections

When multiple reflections return to the camera, it is challenging for 3D cameras to obtain the correct depth. As an example scenario, consider a translucent object like a glass cup which contributes reflections from the glass and the background behind. As a real-world scene, consider Fig. 1.11, where the glass window of the train exhibits multiple reflections. Another realistic scene occurs at a corner, where multiple interreflections smear together (Fig. 1.11).

Handling multiple reflections is a very active problem in 3D camera design. However, the solutions are specific to the type of 3D technology. We discuss these problems in more detail in the specific sections on time of flight and structured light (Sects. 2.4 and 2.5, respectively).

1.7 Depth Artifacts for Specific Vision Tasks

1.7.1 Scene Understanding

Scene understanding is a core problem of high-level computer vision. In recent years, RGB-D data are being widely adapted for object recognition and scene-understanding applications in computer vision, due to availability of affordable depth cameras such as Microsoft Kinect. RGB-D systems provide both 3D information and an ambient illumination-independent channel, increasing the accuracy and robustness of these applications as compared to just RGB image data. However, these methods need to deal with inaccuracies and noise introduced in depth capture process. In this section, we summarize the commonly observed artifacts in depth maps. We focus on indoor scenes, which feature predominantly in these research problems (Fig. 1.12).

Fig. 1.12 Artifacts commonly occur in depth capture of natural scenes using commercial 3D cameras. *Left* RGB image. *Right* Depth map captured by the first-generation Kinect of the same scene. Pixels in *white* are holes in the depth map which arise due to specularities occlusion or clutter. Image courtesy [9]

The artifacts in depth maps that are introduced by the scene being observed are mainly related to ambient illumination, reflectance properties of objects in the scene, and their spatial configuration. Bright ambient illumination can affect the contrast of infrared images in active light sensors, resulting in outliers or holes in the depth map. A cluttered spatial configuration of objects can create occlusions and shadows, which also produces holes in the depth image. Moreover, smooth and specular surfaces appear overexposed in the infrared image, generating holes in the depth map. Currently, computer vision algorithms deal with holes in depth maps using simple inpainting methods [9, 28] (PrimeSense). Applying some of the more sophisticated filtering techniques may improve accuracy for scene-understanding tasks.[1,2]

1.7.2 3D Indoor Mapping

Obtaining a full 3D map of an indoor scene is a classic use case of 3D cameras. We now describe in specific detail the artifacts that occur when obtaining a full 3D map of an indoor environment.

1.8 The Drift Problem

The drift problem usually occurs when the depth camera is facing a scene with poor geometric variation, such as a planar wall. These scenes lack constraints on the degrees of freedom, and thus, estimating the camera pose is challenging. As a

[1] This is an open question.

[2] A related question: what impact will the new Kinect have on the accuracy of current scene-understanding techniques?

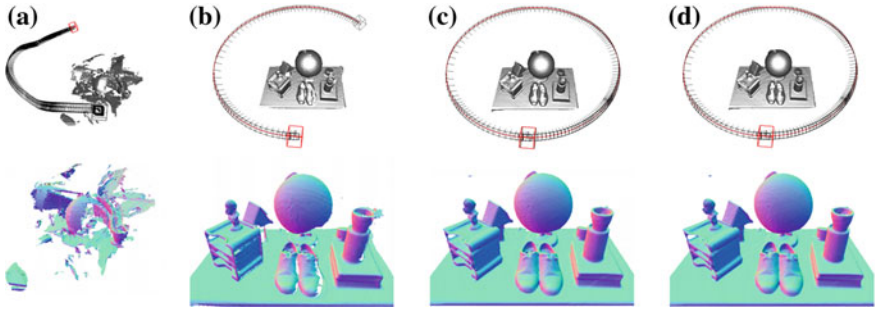


Fig. 1.13 Comparison between frame-to-frame alignment and Kinect fusion [13]. **a** Frame to frame tracking. **b** Partial loop. **c** Full loop. **d** M times duplicated loop

result, the camera trajectory undergoes an uncontrolled drift in front of the scene. To overcome this issue, data from the color camera are necessary to constrain the problem. Since the features extracted in the RGB image will provide a strong localization cue, one can design a combined RGB-D camera system for localization and mapping [10]. By fusing data from the RGB camera and contour information from the depth camera, it is possible to mitigate the drift problem.

1.8.1 Accumulation Error

Due to memory constraints, most localization methods operate on a frame-to-frame basis. This is a sort of Markovian property: To recover the camera parameters for the current frame, we only consider shape alignment for the previous frame. Unfortunately, errors accumulate during the alignment process which can increase into significant artifacts over time. This is known as **accumulation error**, and the severity of artifacts is depicted in Fig. 1.13a. To resolve this problem, [24] and [13] propose a **global alignment** framework to eliminate the accumulation error. Instead of aligning the current frame only to previous frame, Kinect Fusion will align to all the frames which are already aligned. Here, the error introduced by each alignment step is not added together. The improvement in indoor mapping and camera localization is quite dramatic: Compare Fig. 1.13a with Fig. 1.13b–d.

1.8.1.1 Non-rigid Effects

A **rigid** alignment is one where a single transformation operates on each point. In contrast, in a **non-rigid** alignment, the transformation varies for different points. When performing indoor mapping, a typical assumption is that the scene is static. When this is not true a part of the scene is moving. Therefore, the transformation should be different for the moving portions: This is an example of non-rigidity in the mapping

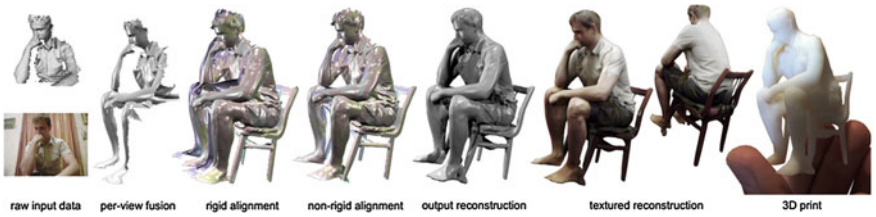


Fig. 1.14 3D scanning pipeline using the first-generation Kinect. The end result is a high-quality fabrication that can be 3D printed. Figure from [21]

process. In [32], a dynamic model of the scene is used as a template. By explicitly factorizing motion using such models and priors, it is possible to mitigate non-rigid artifacts.

1.9 3D Scanning for Fabrication

The previous section discussed indoor mapping of rooms. In this section, we discuss a related problem: high-quality 3D scanning of an object using consumer 3D cameras. Three-dimensional cameras are often described as "**2.5-dimensional**" cameras. Concretely, when using a Kinect to scan a human being, the camera is unable to obtain point coordinates for the backside of the human. In practice, the object of interest is rotated and the resulting point clouds are aligned.

The same artifacts that are present in Sect. 1.7.2 are also germane to 3D scanning. In particular, in Fig. 1.14, a high-quality 3D scan of a person is obtained by using a non-rigid alignment (this mitigates the motion of the person during and between scans). An additional design constraint is in the quality. Since 3D scanning often leads to 3D fabrication, high accuracy and detail are required. In [21], watertight algorithms are used to close up holes.

1.9.1 Finger Tracking

Another common application is finger tracking. Not surprisingly, 3D imaging of fingers is a challenging task due to the small scale of the scene (a finger is approximately 1 cm thick). The list of artifacts is long. Motion artifacts persist, especially when multipattern structured light techniques are used. Lack of clean depth edges due to parallax effects is also prevalent. In addition, line-of-sight obstruction, holes, and poor behavior in sunlight are further sources of artifacts.

With the advent of the new time-of-flight Kinect, several of these problems may be fixed. For the computer vision practitioner, we suggest using a time-of-flight 3D camera for finger tracking as occlusions, ambient light, and motion artifacts are mitigated (Fig. 1.15).

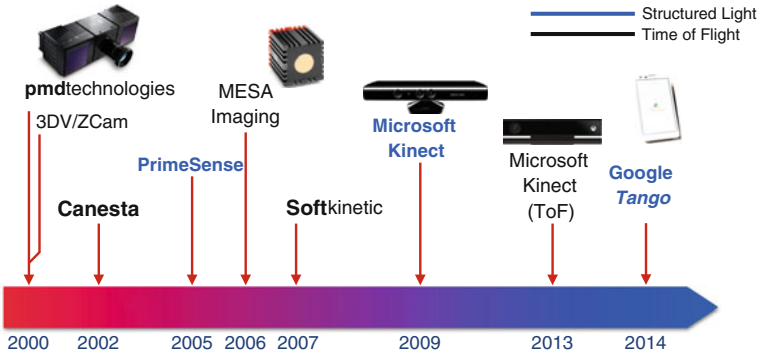


Fig. 1.15 Time line of structured light (*blue*) and time-of-flight sensors (*black*)

1.10 Time-of-Flight Depth Sensors

Time-of-flight cameras are a recent development that offer an interesting alternative to other ranging methods. As the name suggests, this optical instrument operates on the **time-of-flight principle**, where the depth of the object corresponds to the time that the light has been in flight. If the time delay is known, elementary kinematic equations are used to obtain the distance to the object. Consumer quality time-of-flight cameras are developed by a number of firms including MESA imaging system, SoftKinetic, PMD Technologies, and Microsoft, to name a few. One notable example of a consumer-grade sensor based on time-of-flight principle is Microsoft Kinect for Xbox One (2013).

In principle, we must mention that the time-of-flight principle is not a new technology. It has been used for decades in SONAR, spectrometry, spectroscopy and velocimetry, to name a few. However, recent advances in hardware have facilitated the inclusion of **time-of-flight cameras** into the consumer arena. To illustrate the rigor placed on the hardware, consider the following: Light travels one foot in a nanosecond. Thus, to obtain depth to centimeter accuracy, the hardware must be able to detect electronic delays on the order of 100 picoseconds, with the additional constraint that this must be done for each pixel in the array. The data that are obtained from time-of-flight cameras include an intensity image or amplitude image and an image of the time delays, which we refer to as the depth image.

1.10.1 Second-Generation Kinect

Although there are many flavors of "time of flight," we discuss only the operating principle that forms the technological core of the Kinect One. The basic idea is as follows. The Kinect contains a **reference signal** that modulates at a modulation frequency ω (the typical range of ω is between 50 and 150 MHz). The reference signal

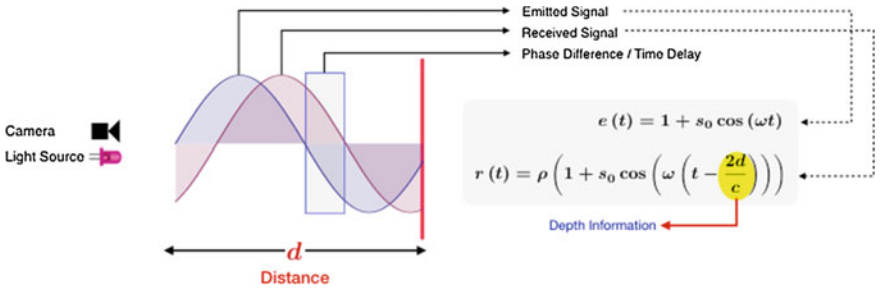


Fig. 1.16 Operating principle of the Kinect One. A phase difference between emitted and received signals encodes the distance

also drives a synchronized solid-state light source to strobe at a sinusoidal pattern. When the optical signal hits an object and returns to the camera, the waveform is offset in phase from the reference signal (Fig. 1.16). The phase offset encodes the depth of the object, and the intensity of the reflection encodes the albedo.

In particular, the light source emits a continuous-wave, periodic signal of form, $e(t) = 1 + s_0 \cos(\omega t)$ Upon reflection from an object at some given distance, the received signal assumes the form of

$$r(t) = \rho\left(1 + s_0 \cos\left(\omega\left(t - \frac{2d}{c}\right)\right)\right) \tag{1.4}$$

Each pixel in the time-of-flight camera samples the cross-correlation function between the transmitted signal and the received signal:

$$\underbrace{C_{e,r}(\tau\kappa)}_{\text{Cross-Correlation}} = \rho\left(1 + \frac{s_0^2}{2}\cos(\omega\tau + \phi)\right), \tau\kappa = \frac{k\pi}{2\omega}, k = 0, \ldots, 3 \tag{1.5}$$

where the phase of the fundamental frequency bin of the cross-correlation function encodes the depth. In principle, note that the fundamental frequency bin includes contributions only from optical sources that are modulated at frequency ω.

1.11 Benefits and Limitations of Time of Flight

A key advantage of time-of-flight cameras is that only a single viewpoint is used to compute depth. This allows robustness to occlusions and shadows and preservation of sharp depth edges. Recall that the existence of holes was largely due to occlusions, and as such, the single viewpoint of a ToF depth sensor is a very significant benefit. Another advantage is that ambient light is rejected (since it does not modulate at frequency ω) (Fig. 1.17).

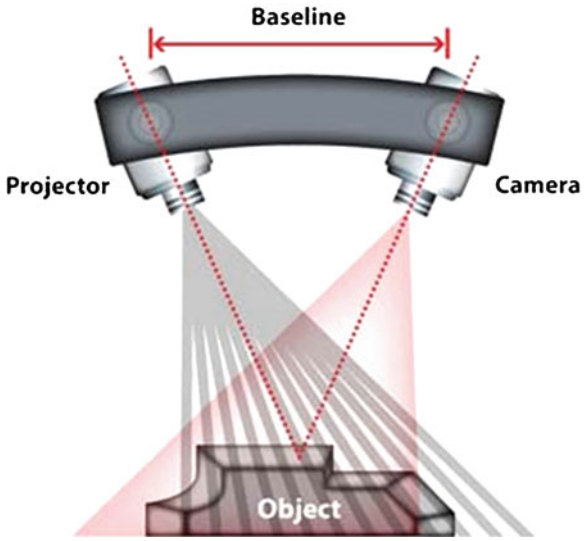


Fig. 1.17 Structured light system. Instead of using two cameras, the vision system uses one projector and one camera

Currently, time-of-flight cameras suffer from a low spatial resolution due to the extra processing that happens at the sensor level. Ambient light can still saturate the sensor, in which case the depth values are meaningless. Like multipattern structured light systems, time-of-flight cameras also require multiple shots. Therefore, the technique is susceptible to motion artifacts.

1.12 Structured Light Depth Sensors

Structured light-based depth estimation can be viewed as a specific case of stereo-vision system where a projector replaces one of the two cameras (Fig. 1.2). In this setting, structured light works by projecting a known pattern onto the subject of interest and the distortion in the project pattern encodes the depth information of the scene.

1.13 First-Generation Kinect (2010)

The first-generation Microsoft Kinect-based sensors use structured light for depth estimation. The system involves a colored camera, an infrared camera, and an infrared laser. The infrared laser emits a beam of light that in turn is split into multiple

beams of light through the diffraction grating mechanism. This assembly mimics the operation of a projector. The projected pattern is captured by the infrared camera and is compared against the known reference code. The disparity between the projected code and the observed code accounts for the depth information of the scene.

1.14 Structured Light Benefits and Limitations

A major advantage of structured light is that it can achieve very high spatial resolution since it uses a conventional imaging device. Structured light systems do not require any special treatment at the sensor level. These systems are simple enough to construct at home using a projector and camera. One of the major advantages of the structured light-based imaging systems is that it circumvents the correspondence problem of stereovision-based systems. This is because any disparity in the setup can be calibrated from the knowledge of distortions in the projected pattern.

For comparable accuracy to a ToF camera, sequential projection of coded or phase-shifted patterns is often required to extract a single depth frame, which leads to *lower frame rate*. This is a restriction in that the subject is required to be relatively still during the projection phase. If the subject moves during the time when the pattern is projected, then the measurements will be characterized by motion artifacts. Another limitation is that the reflected pattern is sensitive to optical interference from the environment. As a result, structured light-based methods tend to be more suited for indoor applications where the environment is controlled. Because structured light systems still rely on optical disparity, the technique is sensitive to occlusions and, by extension, to holes.

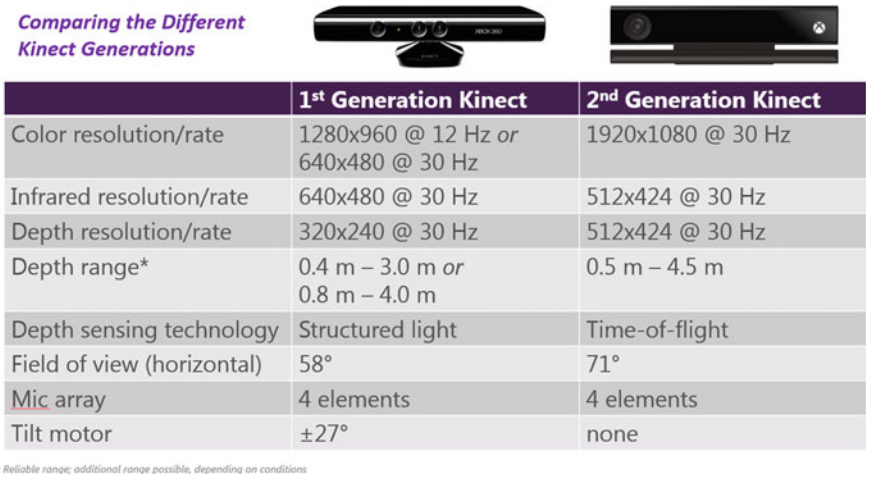
1.15 Comparison of First- and Second-Generation Kinect Models

The first-generation Kinect is a structured light vision system, while the second-generation Kinect is a time-of-flight system. Figure 1.18 illustrates the time line of the Kinect projects. Notable landmarks include the development of the first-generation Kinect in 2010, the Kinect Fusion algorithm in 2013 (Sect. 1.7.2), and the second-generation Kinect in 2013. In summer of 2014, an API for the second-generation Kinect is scheduled for release.

1.15.1 Hardware Specifications

Figure 1.19 tabulates the hardware specifications of the Kinect models. Note that despite the low spatial resolution that usually plagues ToF cameras, both sensors have similar spatial resolutions. The second-generation Kinect has a larger field of view, and therefore, it does not require a tilt motor. Beyond the inherent advantages

Fig. 1.18 Time line of the Kinect projects. The structured light Kinect was released in 2010, while the time-of-flight Kinect was released in 2013. Graphic courtesy John Elsbree



	1st Generation Kinect	2nd Generation Kinect
Color resolution/rate	1280x960 @ 12 Hz *or* 640x480 @ 30 Hz	1920x1080 @ 30 Hz
Infrared resolution/rate	640x480 @ 30 Hz	512x424 @ 30 Hz
Depth resolution/rate	320x240 @ 30 Hz	512x424 @ 30 Hz
Depth range*	0.4 m – 3.0 m *or* 0.8 m – 4.0 m	0.5 m – 4.5 m
Depth sensing technology	Structured light	Time-of-flight
Field of view (horizontal)	58°	71°
Mic array	4 elements	4 elements
Tilt motor	±27°	none

* Reliable range; additional range possible, depending on conditions

Fig. 1.19 Hardware specifications of the first- and second-generation Kinect models

of time-of-flight technology, the second-generation Kinect also boasts a higher depth resolution, though this has not yet been verified.

1.16 3D Cameras of the Future

Kinect-style 3D cameras represent only a small part of 3D camera technology. In this section, we cover some technologies that may be of interest for applications outside the living room.

Fig. 1.20 A modified time-of-flight camera is able to recover the depth of the translucent unicorn. This is an example of emerging multidepth cameras. Figure from [16]

1.16.1 Multidepth 3D Cameras

Figure 1.11 provided brief insight into the problem of multiple reflections. Recently, several papers extend the time-of-flight camera paradigm to handle multiple depths that arise in a variety of scenarios. One such scenario is shown in Fig. 1.20, where a modified time-of-flight camera is able to recover the range of a translucent horse [16]. Other related papers include [2, 8, 23] and [29].

In Sect. 1.2.2, the registration of color and depth data was discussed. An interesting topic is whether it is possible to acquire both color images and depth maps with a single camera. In [15], a real-time implementation was proposed that extends time-of-flight technology to multiplex active light sources to provide color and range simultaneously.

1.16.2 Photometric Approaches

In addition to the depth sensor based on geometric approach, photometric cues play an important role in estimating the 3D information. The intuition here is that the object appearance viewed from a fixed position will be different, depending on the various lighting conditions. For example, given a white sphere illuminated by a point source which is far away, the surface points whose normal orientations have smaller angles with the lighting direction look brighter and vice versa. The shape estimation methods relying on the photometric information mainly include two types: (1) shape-from-shading [12], which uses only one image and can only recover the shape up to some ambiguities, and (2) photometric stereo [30], which uses at least three images from fixed viewpoint and different lighting directions to uniquely determine the shape. Note that the direct output of photometric approaches is surface normal, which is the derivative of surface. In order to reconstruct the 3D points or depth, integration has to be applied on surface normal.

Compared to a geometric approach, the main advantage of the photometric approach lies in its ability to recover delicate surface structures. Since the photometric approach relies on the analysis of how light interacts with surface reflectance, the quality of shape estimation depends heavily on the reflectance of target object.

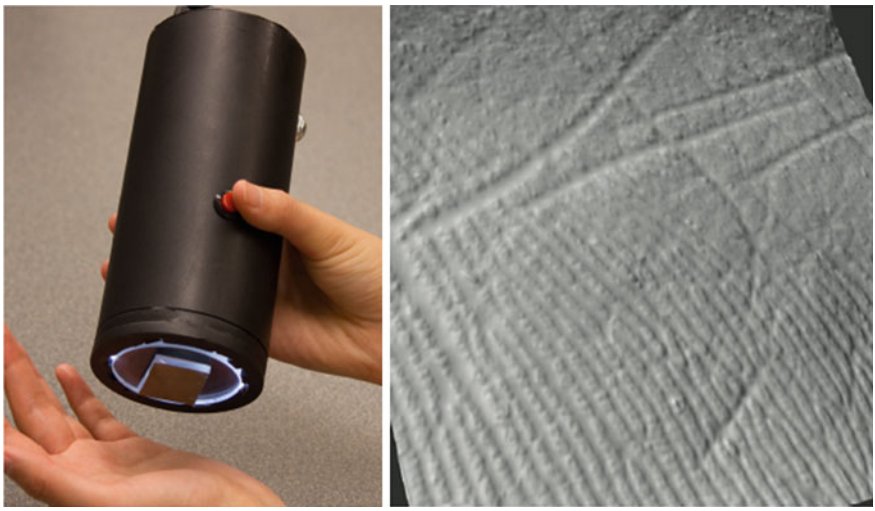


Fig. 1.21 The photometric stereo 3D camera—GelSight (Johnson 2011) and its reconstruction of human skin

A comprehensive evaluation on surface normal estimation accuracy versus real-world reflectances can be found in [27].

If the surface reflectance can be carefully calibrated, the photometric-based 3D camera can be applied to sense microscopic surface structure (Johnson 2011). As shown in Fig. 1.21, the GelSight camera demonstrates a depth resolution of 2 microns, which is superior to geometric-based approaches (and of course Kinect-style cameras).

1.17 Mobile 3D Cameras

As the cost and size of mobile cameras and 3D sensors continue to decrease, consumer electronic manufacturers are finding it increasingly more feasible to incorporate 3D cameras into the next generation of mobile devices. In particular, industry trends show an intensified focus on applications within the context of augmented reality, image refocusing, and real-time key point tracking.

The LG Thrill was one of the first consumer mobile 3D devices (2010). This Android-powered smartphone featured two 5-megapixel cameras that used the stereoscopic effect to capturing 3D pictures and video. Even though the device was equipped with a glasses-free 3D display to match, it ultimately failed to take hold in the market because of its excessive weight and size and relatively poor photograph quality. In 2012, Pelican Imaging released a small 3-mm-thick light field camera array that can be embedded into mobile platforms. To calculate depth, Pelican uses integrated parallax detection and super-resolution (Fig. 1.22).

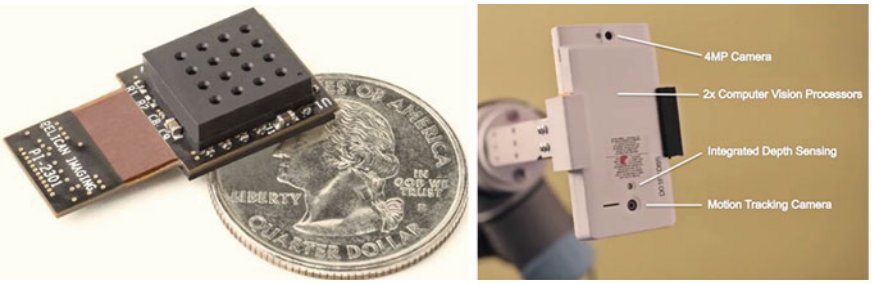


Fig. 1.22 *Left* Pelican's mobile light field camera. *Right* Google's Project Tango is an integrated mobile phone and 3D camera that projects an active pattern

In early 2014, Google's "Project Tango" smartphone, designed to calculate its orientation and location in three-dimensional space, was announced. This device is capable of collecting 3D data using structured IR and near-IR light. Alongside the device are two dedicated computer vision processors and a wide-angle motion-tracking camera that allow it to collect 250,000 3D measurements per second (Fig. 1.22). Most recently, HTC's One M8, released in early 2014, uses high-resolution stereoscopic cameras for refocus and background manipulation.

1.18 Conclusion

Three-dimensional cameras are primed to change computer vision in a meaningful way. However, 3D data must be used with caution. Rather than simply plugging in a depth map into an application, it is important to ensure that the depth map is a meaningful 3D representation of the scene; this usually requires correction of artifacts (the two main artifacts in depth maps are holes and edge erosion).

Let us return to the query that was posed in the initial stages of this chapter: Is time of flight "better" than structured light? We can now provide the following answer: *It depends on the task*. Concretely, Sect. 1.9.1 mentioned that finger tracking with a structured light camera is prone to holes caused by line-of-sight obstruction. Therefore, we can recommend using a comparable time-of-flight sensor for this application.

A final question that is raised is whether the technology used to capture the 3D data has any impact on the higher-level algorithms that are used. As a concrete example, if a scene-understanding researcher is collecting a dataset, does she need to discriminate between data that is collected with the first-generation Kinect or the second-generation Kinect? Will she see a significant improvement in performance? This is an open question.

References

1. Banno A, Ikeuchi K (2011) Disparity map refinement and 3d surface smoothing via directed anisotropic diffusion. Comput Vis Image Underst 115(5):611–619
2. Bhandari A, Kadambi A, Whyte R, Barsi C, Feigin M, Dorrington A, Raskar R (2014) Resolving multipath interference in time-of-flight imaging via modulation frequency diversity and sparse regularization. Opt Lett 39(6):1705–1708
3. Birchfield S, Tomasi C (1999) Depth discontinuities by pixel-to-pixel stereo. Int J Comput Vis 35(3):269–293
4. Camplani M, Salgado L (2012) Efficient spatio-temporal hole filling strategy for kinect depth maps. In: Proceedings of SPIE, vol 8920
5. Chen L, Lin H, Li S (2012) Depth image enhancement for kinect using region growing and bilateral filter. In: 21st international conference on pattern recognition (ICPR), 2012, pp 3070–3073. IEEE
6. Durand F, Dorsey J (2002) Fast bilateral filtering for the display of high-dynamic-range images. In: ACM transactions on graphics (TOG), vol 21, pp 257–66. ACM
7. Grimson WEL (1985) Computational experiments with a feature based stereo algorithm. IEEE Trans Pattern Anal Mach Intell 1:17–34
8. Heide F, Hullin MB, Gregson J, Heidrich W (2013) Low-budget transient imaging using photonic mixer devices. ACM Trans Graph (TOG) 32(4):45
9. Henry P, Krainin M, Herbst E, Ren X, Fox D (2014) RGB-D mapping: using depth cameras for dense 3D modeling of indoor environments. In: Experimental robotics, pp 477–491. Springer, Berlin
10. Henry P, Krainin M, Herbst E, Ren X, Fox D (2012) RGB-D mapping: using kinect-style depth cameras for dense 3D modeling of indoor environments. Int J Robot Res 31(5):647–63
11. Hoegg T, Lefloch D, Kolb A (2013) Real-time motion artifact compensation for PMD-ToF images. In: Time-of-flight and depth imaging. Sensors, algorithms, and applications, pp 273–288. Springer, Berlin
12. Horn BKP (1970) Shape from shading: a method for obtaining the shape of a smooth opaque object from one view
13. Izadi S, Kim D, Hilliges O, Molyneaux D, Newcombe R, Kohli P, Shotton J, et al (2011) KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera. In: Proceedings of the 24th annual ACM symposium on user interface software and technology, pp 559–568. ACM
14. Jones DG, Malik J (1992) Computational framework for determining stereo correspondence from a set of linear spatial filters. Image Vis Comput 10(10):699–708
15. Kadambi A, Bhandari A, Whyte R, Dorrington A, Raskar R (2014) Demultiplexing illumination via low cost sensing and nanosecond coding. In: 2014 IEEE international conference on computational photography (ICCP). IEEE
16. Kadambi A, Whyte R, Bhandari A, Streeter L, Barsi C, Dorrington A, Raskar R (2013) Coded time of flight cameras: sparse deconvolution to address multipath interference and recover time profiles. ACM Trans Graph (TOG) 32(6):167
17. Kanade T, Okutomi M (1994) A stereo matching algorithm with an adaptive window: theory and experiment. IEEE Trans Pattern Anal Mach Intell 16(9):920–932
18. Kauff P, Atzpadin N, Fehn C, Müller M, Schreer O, Smolic A, Tanger R (2007) Depth map creation and image-based rendering for advanced 3DTV services providing interoperability and scalability. Sig Process Image Commun 22(2):217–234
19. Klaus A, Sormann M, Karner K (2006) Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In: 18th international conference on pattern recognition, 2006. ICPR 2006, vol 3, pp 15–18. IEEE
20. Lenzen F, Schäfer H, Garbe C (2011) Denoising time-of-flight data with adaptive total variation. In: Advances in visual computing, pp 337–346. Springer, Berlin
21. Li H, Vouga E, Gudym A, Luo L, Barron JT, Gusev G (2013) 3D self-portraits. ACM Trans Graph (TOG) 32(6):187

22. Marr D, Poggio T (1976) Cooperative computation of stereo disparity. Science 194(4262):283–287
23. Naik N, Zhao S, Velten A, Raskar R, Bala K (2011) Single view reflectance capture using multiplexed scattering and time-of-flight imaging. In: ACM Transactions on Graphics (TOG), vol 30, p 171. ACM
24. Newcombe RA, Davison AJ, Izadi S, Kohli P, Hilliges O, Shotton J, Molyneaux D, Hodges S, Kim D, Fitzgibbon A (2011) KinectFusion: real-time dense surface mapping and tracking. In: 2011 10th IEEE international symposium on mixed and augmented reality (ISMAR), pp 127–136. IEEE
25. Ohta Y, Kanade T (1985) Stereo by intra-and inter-scanline search using dynamic programming. IEEE Trans Pattern Anal Mach Intell 2:139–154
26. Penne J, Schaller C, Hornegger J, Kuwert T (2008) Robust real-time 3D respiratory motion detection using time-of-flight cameras. Int J Comput Assist Radiol Surg 3(5):427–431
27. Shi B, Tan P, Matsushita Y, Ikeuchi K (2014) Bi-polynomial modeling of low-frequency reflectances. IEEE Trans Pattern Anal Mach Intell 36(6):1078–1091
28. Silberman N, Hoiem D, Kohli P, Fergus R (2012) Indoor segmentation and support inference from RGBD images. In: Computer vision-ECCV 2012, pp 746–760. Springer, Berlin
29. Velten A, Wu D, Jarabo A, Masia B, Barsi C, Joshi C, Lawson E, Bawendi M, Gutierrez D, Raskar R (2013) Femto-photography: capturing and visualizing the propagation of light. ACM Trans Graph (TOG) 32(4):44
30. Woodham RJ (1980) Photometric method for determining surface orientation from multiple images. Opt Eng 19(1):191139
31. Ye C, Hegde GPM (2009) Robust edge extraction for SwissRanger SR-3000 range images. In: IEEE international conference on robotics and automation, 2009. ICRA'09, pp 2437–2442. IEEE
32. Ye G, Liu Y, Hasler N, Ji X, Dai Q, Theobalt C (2012) Performance capture of interacting characters with handheld kinects. In: Computer vision-ECCV 2012, pp 828–841. Springer, Berlin
33. Yoon K-J, Kweon IS (2006) Adaptive support-weight approach for correspondence search. IEEE Trans Pattern Anal Mach Intell 28(4):650–656
34. Zhang C, Zhang Z (2011) Calibration between depth and color sensors for commodity depth cameras. In: 2011 IEEE international conference on multimedia and expo (ICME), pp 1–6. IEEE
35. Zhang Z (2000) A flexible new technique for camera calibration. IEEE Trans Pattern Anal Mach Intell 22(11):1330–1334