Chapter 3
Microsoft Kinect™ Range Camera

The Microsoft Kinect™ range camera (for simplicity just called Kinect™ in the sequel) is a light-coded range camera capable to estimate the 3D geometry of the acquired scene at 30 *fps* with VGA (640×480) spatial resolution. Besides its light-coded range camera, the Kinect™ also has a color video-camera and an array of microphones. In the context of this book the Kinect™ light-coded range camera is the most interesting component, and the name Kinect™ will be often referred to it rather than to the whole product.

From the functional point of view the Kinect™ range camera is very similar to the ToF cameras introduced in the previous chapter, since they both estimate the 3D geometry of dynamic scenes, but current Kinect™ technology is totally different. The Kinect™ range camera is based on the Primesensor™ chip produced by Primesense [1]. Although the Kinect™ (Figure 3.1) is the most popular consumer electronic product based on such a chip, it is not the only one. Asus X-tion Pro and X-tion Pro Live [2] are other products with a range camera based on the same chip. All these range cameras, as it will be seen, support an IR video-camera and an IR

Fig. 3.1 Microsoft Kinect™.

projector projecting on the scene IR light coded patterns in order to obtain a matricial implementation of the active triangulation principle for estimating scene depth and 3D geometry.

This chapter firstly describes how active triangulation can be implemented in a

C.D. Mutto et al., *Time-of-Flight Cameras and Microsoft Kinect™*, SpringerBriefs in Electrical and Computer Engineering, DOI 10.1007/978-1-4614-3807-6_3, © The Author(s) 2012

matricial form by means of different light coding techniques. The general operation of Kinect™ can be inferred from that of light-coding systems, of which the Kinect™ is a special case, but its implementation details are still undisclosed. Nevertheless some characteristics of its functioning can be derived from its operation as reported by some reverse engineering studies [3, 4].

The practical imaging issues responsible for the Kinect™ error sources and data characteristics are finally considered in the last section of this chapter, as previously done for ToF cameras.

3.1 Matricial active triangulation

The Kinect™ range camera has the structure shown in Figure 1.7, reported also in Fig. 3.2, with a camera C and a projector A, and in principle it implements active triangulation. Namely, given a point p_C in the image acquired by C and its conjugate point p_A in the pattern projected by A, the depth z of the 3D point P associated to p_C with respect to the C-3D reference system is computed by active triangulation as described in Section 1.2.4.
In order to use a pattern invisible to human eyes, both projector and camera operate at IR wavelengths. The data made available from the Kinect™ range camera at $30\,[fps]$ are:

- the image acquired by C, that in the Kinect™ case is an IR image called I_K and it is defined on the lattice Λ_K associated to the C sensor. The axes that identify Λ_K coincide with u_C and v_C of Figure 1.7. The values of I_K belong to interval $[0,1]$. Image I_K can be considered a realization of a random field $\mathscr{I}_K$ defined on Λ_K, with values in $[0,1]$.
- The estimated disparity map, called $\hat{D}_K$ is defined on the lattice Λ_K associated to the C sensor. The values of $\hat{D}_K$ belong to interval $[d_{min}, d_{max}]$, where d_{min} and d_{max} are the minimum and maximum allowed disparity values. Disparity map $\hat{D}_K$ can be considered a realization of a random field $\mathscr{D}_K$ defined on Λ_K, with values in $[d_{min}, d_{max}]$.
- The estimated depth map computed by applying (1.14) to $\hat{D}_K$, called $\hat{Z}_K$ is defined on the lattice Λ_K associated to the C sensor. The values of $\hat{Z}_K$ belong to the interval $[z_{min}, z_{max}]$, where $z_{min} = \frac{bf}{d_{max}}$ and $z_{max} = \frac{bf}{d_{min}}$ are the minimum and maximum allowed depth values respectively. Depth map $\hat{Z}_K$ can be considered as a realization of a random field $\mathscr{Z}_K$ defined on Λ_K, with values in $[z_{min}, z_{max}]$.

The spatial resolution of I_K, $\hat{D}_K$ and $\hat{Z}_K$ is 640×480. The minimum measurable depth is $0.5[m]$ and the nominal maximum depth is$15[m]$. According to [3], the values of b and f are $75[mm]$ and $585.6[pxl]$ respectively. Therefore the minimum allowed disparity is $2[pxl]$ and the maximum is $88[pxl]$. Figure 3.3 shows an example of I_K, $\hat{D}_K$ and $\hat{Z}_K$ acquired by the Kinect™ range camera.

Section 1.2 introduced active triangulation applied to single pixels p_C of the $N_R \times N_C$ images I_K acquired by camera C. The Kinect™, instead, simultaneously

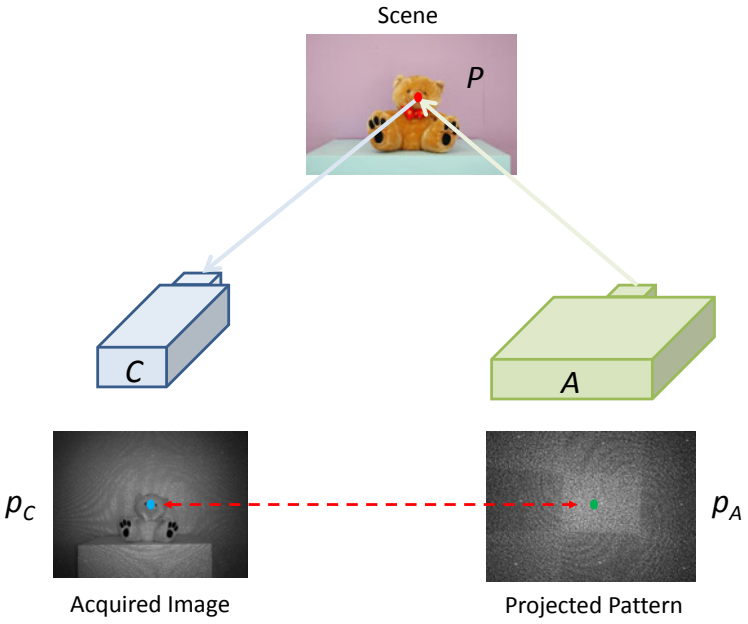


Fig. 3.2 Matricial active triangulation flow: pixel p_A (green dot) is coded in the pattern. The pattern is projected to the scene and acquired by C. The 3D point associated to p_A is P and the conjugate point of p_A (green dot) in I_K is p_C (blue dot). The correspondence estimation algorithm (red dashed arrow) estimates the conjugate points.

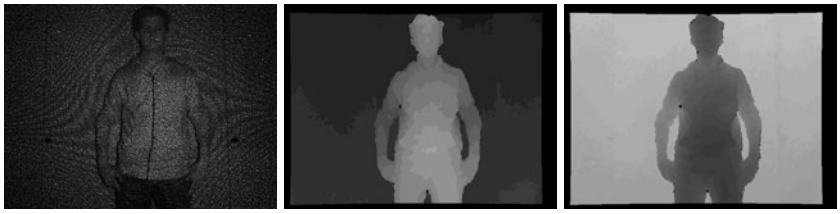
applies active triangulation to all the $N_R \times N_C$ pixels of I_K, a procedure called *matricial active triangulation*, which requires a number of special provisions discussed in the next subsection.

The major difficulty with matricial active triangulation is keeping the correspondence problem as simple as in the single point case seen in Section 1.2. This issue can be handled by designing the patterns projected by A by the *light coding* methods described next. The specific design of the light-coded pattern is the actual core of the Kinect^{TM} range camera. The next two subsections present current light coding techniques and give some hints about the specific Kinect^{TM} operation.

3.1.1 Light Coding Techniques

Let us assume that the projected pattern has $N_R^A \times N_C^A$ pixels p_A^i, $i = 1,...,N_R^A \times N_C^A$ where N_R^A and N_C^A are the number of rows and columns of the projected pattern respectively. In order to apply active triangulation, each pixel needs to be associated to a *code-word*, i.e., a specific local configuration of the projected pattern. The pattern undergoes projection by A, reflection by the scene and capture by C (see Figure 3.2). A correspondence estimation algorithm analyzes the received code-words in the acquired images I_K in order to compute the conjugate of each pixel of the projected pattern. The goal of pattern design (i.e., code-words selection) is to adopt code-words effectively decodable even in presence of non-idealities of the pattern projection/acquisition process, as pictorially indicated in Fig. 3.2 .

Let us first consider what makes a code-word highly decodable. It is intuitive that



a) b) c)

Fig. 3.3 Example of I_K, $\hat{D}_K$ and $\hat{Z}_K$ acquired by the Kinect™ range camera.

the more the code-words are different the more robust is the coding against disturbances and self-interferences. For a given cardinality of the total number of possible code-words, the smaller is the number of used code-words, the greater become the differences between the various code-words and the more robust is the code. Since for a calibrated and rectified setup conjugate points lie on horizontal lines, the coding problem can be independently formulated for each row in order to keep as low as possible the cardinality of the total number of possible code-words. Assume that for each row of the projected pattern there are $N = N_C^A$ pixels $p_A^1, p_A^2, ..., p_A^N$ to be encoded with N code-words $w_1, w_2, ..., w_N$. Each code-word is represented by a specific local pattern distribution. Clearly, the more the local pattern distribution of a single pixel differs from the local pattern distribution of the other pixels of the same row, the more robust will be the coding.

A code-words alphabet can be implemented by a light projector considering that it can produce n_P different illumination values called *pattern primitives* (e.g., $n_P = 2$ for a binary black-and-white projector, $n_P = 2^8$ for a $8-bit$ gray-scale projector and $n_P = 2^{24}$ for a RGB projector with $8-bit$ color channels). The local distribution of a pattern for a pixel p_A is given by the illumination values of the pixels in a window around p_A. If the window has n_W pixels, there are $n_P^{n_W}$ possible pattern configurations on it. From the set of all possible configurations, N configurations need to be chosen as code-words. What is projected to the scene and acquired by C is the pat-

tern resulting from the code-words relative to all the pixels of the projected pattern.

Let us recall that, because of the geometrical properties of a calibrated and rectified system described in Section 3.1, a pixel p_A of the pattern, with coordinates $\mathbf{p_A} = [u_A, v_A]^T$, is projected to the scene point P, with coordinates $\mathbf{P} = [x, y, z]^T$, and acquired by C in p_C, with coordinates $\mathbf{p_C} = [u_A + d, v_A]^T$. The projection/acquisition process introduces an horizontal shift d proportional to the inverse of the depth z of P according to (1.14). Disparity shift d is the most important quantity in the active triangulation process, and it needs to be accurately estimated since it carries the 3D geometry information relative to the considered scene point P (obtainable from (1.15)). The disparity values in matricial active triangulation are handled as arrays, like the disparity map $\hat{D}_K$ of Kinect$^{\text{TM}}$ introduced in Section 3.1.

In the pattern projection/acquisition process, there are a number of factors transforming the projected pattern and introducing artifacts which must be taken into consideration:

a) *Perspective distortion.* Since the scene points may have different depth value z, neighboring pixels of the projected pattern may not be mapped to neighboring pixels of I_C. In this case the local distribution of the acquired pattern becomes a distorted version of the relative local distribution of the projected pattern.
b) *Color or gray-level distortion due to scene color distribution and reflectivity properties of the acquired objects.* The projected pattern undergoes reflection (and absorption) by the scene surfaces. The ratio between incident and reflected radiant power is given by the scene reflectance, which is generally related to the scene color distribution. In particular in the case of IR light, used by the Kinect$^{\text{TM}}$ projector, the appearance of the pixel p_C on the camera depends on the reflectance of the scene surface at the IR frequency used by the projector. An high intensity pixel of the projected pattern at p_A for instance may undergo a strong absorption because of the low reflectance value of the scene point to which it is projected, and the values of its conjugate pixel p_C on I_K may consequently appear much darker. This is a very important issue, since it might completely distort the projected code-words. The second row of Figure 3.4 shows how the radiometric power of the projected pattern is reflected by surfaces of different color.
c) *External illumination.* The color acquired by the color camera depends on the light falling on the scene surfaces, which is the sum of the projected pattern and of the scene illumination (i.e. sunlight, artificial light sources, etc..). This second contribution with respect to code-word detection acts as a noise source added to the information signal of the projected light.
d) *Occlusions.* Because of occlusions, not all the pattern pixels are projected to 3D points seen by C. Depending on the 3D scene geometry, in general between the pattern pixels and the pixels of I_K (the image acquired by C), there may not be a biunivocal association. It is important therefore to correctly identify the pixels of I_K that do not have a conjugate point in the pattern, in order to discard erroneous correspondences.

e) *Projector and camera non-idealities.* Both projector and camera are not ideal imaging systems. In particular, they generally do not behave linearly with respect to the projected and the acquired colors or gray-levels.
f) *Projector and camera noise.* The presence of random noise in the projection and acquisition processes is typically modeled as Gaussian additive noise in the acquired images I_K.

Figure 3.4 shows some examples of the considered transformations/distortions. As result of the above transformations/distortions, an acquired code-word may be very different from the projected one and due to occlusions some pixels of I_K may even not correspond to any pixel of the projected pattern.

In order to understand how to possibly mitigate such potentially disruptive effects for the correspondence problem by comparison methods assisted by suitable patterns, let us make two considerations. The first one is that the correspondences estimation process is characterized by two fundamental decisions, namely:

- what code-word assign to each pixel p_A of the projected pattern. The code-word corresponds to a pattern to be projected on a window centered at p_A (all the local pattern distributions of neighboring pixels are fused together in a single projected pattern);
- what code-word assign to each pixel p_C of I_K, or equivalently how to detect the code-word most similar to the local pattern distribution around p_C in order to correctly identify the conjugate pixel p_C of the projected pattern pixel p_A.

The second consideration is that there are mainly three possible coding schemes, pictorially exemplified in Figure 3.5:

a) *Direct coding*: the code-word associated to each pixel p_A is represented by the pattern value at the pixel itself (i.e., the gray-level or the color of the pattern at p_A). In this case there may be up to n_P code-words, since $n_W = 1$. Therefore the maximum number of pattern columns to decode is $N_C = n_P$.
b) *Time-multiplexing coding*: a sequence of T patterns is projected to the surface to be measured at T subsequent times. The code-word associated to each pixel p_A is the sequence of the T pattern values (i.e., of gray-level or color values) at pixel p_A. In this case there may be up to n_P^T code-words and the maximum number of pattern columns to decode is $N_C = n_P^T$.
c) *Spatial-multiplexing coding*: the code-word associated to each pixel p_A is the spatial pattern distribution in a window of n_W pixels centered around p_A. In this case there might be up to $n_P^{n_w}$ code-words. For instance if the window has 9 rows and 9 columns, which is a common choice, $n_W = 81$. It is important to note how in this case neighboring pixels share parts of their code-words, thus making their coding interdependent.

Each one of the considered coding strategies have different advantages and disadvantages as described in [5], which gives an exhaustive review of all these techniques. In particular, direct coding methods are the easiest to implement and allow

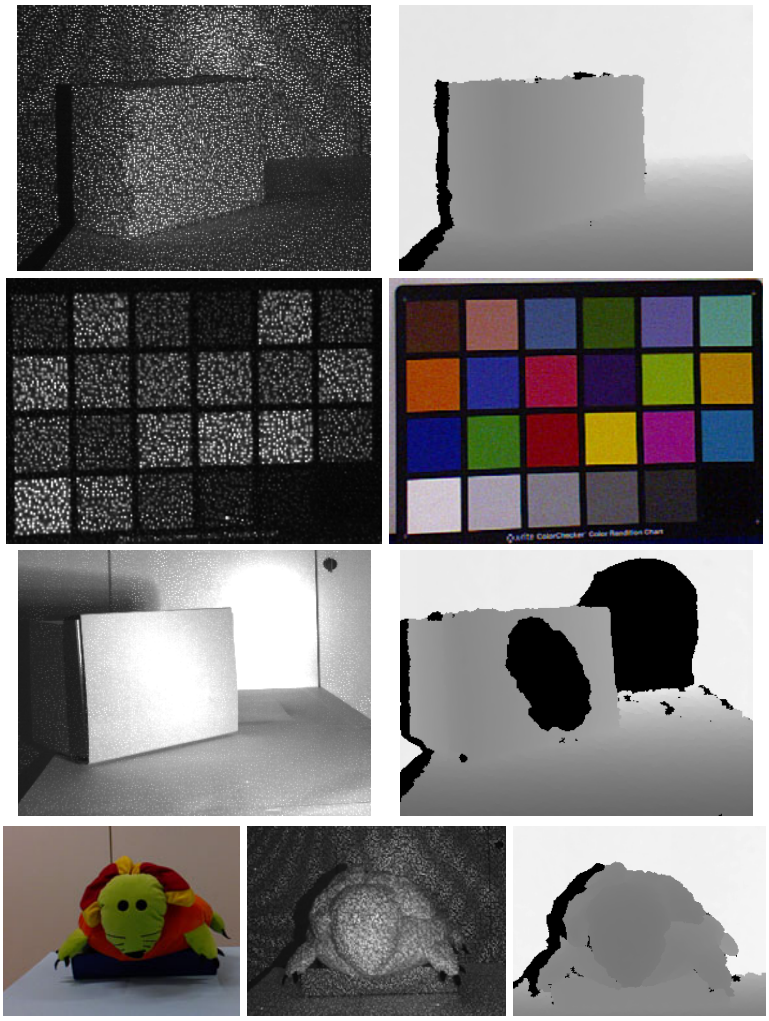


Fig. 3.4 Examples of different artifacts affecting the projected pattern (*in the depth maps black pixels correspond to locations without a valid depth measurement*). *First row*: projection of the IR pattern on a slanted surface and corresponding depth map; observe how the pattern is shifted when the depth values change and how perspective distortion affects the pattern on the slanted surfaces. *Second row*: Kinect^{TM} pattern projected on a color checker and corresponding depth-map; observe how the pattern appearance depends also on the surface color. *Third row*: a strong external illumination affects the acquired scene; the acquired IR image saturates in correspondence of the strongest reflections and the Kinect^{TM} is not able to acquire the depth of those regions. *Fourth row*: the occluded area behind the ear of the teddy bear is visible from the camera but not from the projector viewpoint; consequently the depth of this region can not be computed.

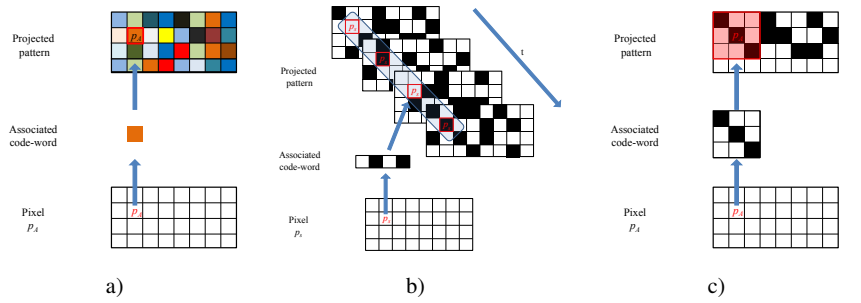


Fig. 3.5 Examples of coding strategies: a) direct coding; b) time-multiplexing coding; c) spatial-multiplexing coding.

for the capture of dynamic scenes, since they require the projection of a single pattern. They have the advantage to be potentially able to deal with occlusions as well as with projective distortion, and the disadvantage to be extremely sensitive to color or gray-level distortion due to scene color distribution, reflectivity properties and external illumination. Furthermore they are also very sensitive to projector and camera noise and non-idealities. Examples of these approaches are [6, 7].

Time-multiplexing coding allows to adopt a very small set of pattern primitives (e.g., a binary set), in order to create arbitrarily different code-words for each pixel. Such code-words are therefore robust with respect to occlusions, projective distortion, projector and cameras non-idealities and noise. They are also able to deal with color and gray-level distortion due to scene color distribution, reflectivity properties and external illumination. Their major disadvantage is that they require the projection of a time sequence of T patterns for a single depth measurement, hence their application is not suited to dynamic scenes. Examples of time-multiplexing approaches are the ones based on binary coding [8] and gray coding with phase-shifting [9].

The spatial-multiplexing techniques are the most interesting for the acquisition of dynamic scenes since they require the projection of a single pattern, like direct coding methods. They are generally robust with respect to projector and cameras non-idealities. They can also handle noise and color or gray-level distortion due to scene color distribution, reflectivity properties and external illumination. Difficulties come with occlusions and perspective distortion, because of the finite size of the pattern window associated to each code-word. The choice of the window size n_W associated to each code-word is crucial. Indeed, on one hand, the smaller is n_W the more robust is the coding with respect to perspective distortion, since it is more likely that all the scene points on which the pattern window is projected share the same disparity. On the other hand, the greater is n_W the greater is the robustness with respect to projector and cameras non-idealities and noise or color distortion. Examples of spatial-multiplexing coding are the ones based on non-formal codification [10], the ones based on De Bruijns sequences [11] and the ones based on M-arrays [12].