Chapter 3
Pre-processing

Computer vision is aimed at simulating the human visual system in order to extract useful information for machines to make decisions. A visual camera is usually used for this purpose which detects brightness, colour, texture and dimensions of an object in focus. When a camera captures scenery, it contains both ‘wanted’ as well as ‘unwanted’ information. If the camera is focussed on a person’s hand looking for a possible gesture, then the ‘unwanted’ objects in the scenery would be the background which may contain the person’s body, clothing, other people, pets, walls, windows, curtains or any other equipment. Since the system is developed to respond to gestures, the system would try to extract only the ‘wanted’ information. However, as the system would not have the level of intelligence as a human, it relies on ‘clues’ to extract only the ‘wanted’ objects.

Recognizing the ‘wanted’ information poses many challenges in computer vision. In the case of hand gesture, how a machine would identify a hand with various gestures that it could produce with different looking skin tones from around the world is difficult problem. This problem is even more compounded when hand gestures are captured in varying lighting conditions as the same hand would look different under different lighting conditions. Yet, the amount of knowledge that has been gathered in the past few decades will offer potential solutions to sift ‘wanted’ information from ‘unwanted’ clutter. This chapter will discuss many concepts of skin segmentation, morphological filtering, noise removal, and depth measurements of objects in order to identify the ‘wanted’ information reliably in the context of hand gesture recognition.

The next section will detail the approach that a machine would take to look for human hand called ‘skin segmentation’. Once an object resembling human skin is detected, the system would expect to extract further information from this skin-tone region. However, due to poor lighting and other imperfections in the camera sensor, the extracted skin looking region may turn out to be ‘noisy’ resembling rough edges and missing parts in a skin region. These imperfections would be removed using a process called morphological filtering as would be discussed later. Finally, recent developments in the camera technology that derives depth information together with visual information provides opportunities to remove unwanted areas in an image using depth information would be discussed at the end of this chapter.

P. Premaratne, *Human Computer Interaction Using Hand Gestures,*
Cognitive Science and Technology, DOI 10.1007/978-981-4585-69-9_3,

3.1 Skin Segmentation

Skin segmentation is the process of looking for skin-like regions or skin tone in a visual image. The purpose of skin segmentation lies in the applications of computer vision such as people detection and tracking, face detection and tracking and gesture recognition and tracking. Once detected, this information will lead to applications in door access control, crowd counting, robotic control and human computer interaction, removing pornographic content using internet filters and many other video applications. There have been other instances of applications in automatic video annotations where newscasters were detected using skin color present in face and hand regions [1] and in image retrieval from image archives. There are many similar applications where background is controlled or unlikely to contain skin color tones, skin color detection is used to detect human faces and hands in face recognition in controlled environments [2].

Human skin is relatively easy to detect in controlled environments. However, detection in uncontrolled settings such as in consumer digital photographs is generally difficult. The appearance of skin in an image depends on illumination, geometry and color when the image was captured [3]. The humans are known to be adept at recognizing color of objects in different illumination conditions known as color constancy. This is however, is not trivial for a machine to achieve with our present level of understanding of imaging. Algorithms need to be robust enough to deal with variations in lighting or illumination, color resolution, and imaging noise. There are also other issues where skin-tone colors are found in wood, leather, certain clothing, hair, sand, paints, etc. These materials cause the classifiers to record false positives when looking for skin-tones.

3.1.1 The Problem of Skin Detection

Skin detection problem is recognized as a classification problem in many computer vision problems. In many common approaches, skin tones belonging to many ethnicities around the world under different lighting conditions are used to build databases to develop algorithms to classify them effectively. As would be discussed in the following sections, it has been found that the standard RGB color space is not the optimum color space for skin detection. Researchers have used experimental data to conclude that different color spaces have varying capabilities at extracting features or learning parameters to have better performance when extracting information to classify skin tones. As shown in Fig. 3.1, it is logical to select a color space where skin tones are represented more compactly. In this graph, Asian, African and Caucasian skin colors in *R* and *G* color spaces occupy different regions (sub images (a), (b) and (c)). However, these apparent different skin colors are confined to smaller area where they cluster together in UV, C_bC_r regions. This fact highlights why UV (from YUV color space) or C_bC_r (from YC_bC_r color space) is better than RGB color space in detecting skin tones.

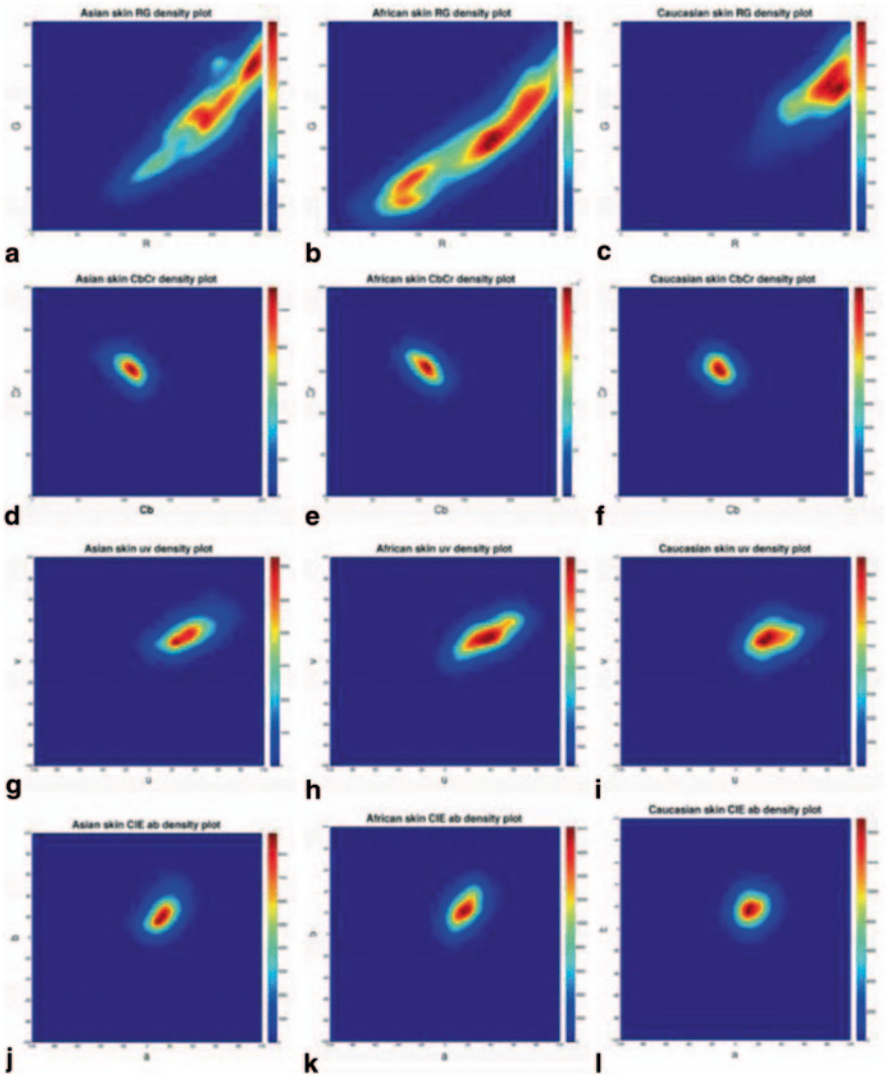


Fig. 3.1 Skin color tones do differ dramatically with ethnicity from different parts of the world when compared in RGB color space but is more stable in C_BC_R, CIE Lab and UV spaces. (Courtesy of [3])

Forsyth and Fleck [4] have reasoned why human skin color has limited range of hues despite the appearance of different skin tones from different parts of the world. The skin tone of any ethnicity is simply formed by combination of blood which is red and melanin which is brown. Therefore, despite the appearance, human skin color clusters in a small area in any color space. Researchers have experimented with different color spaces in order to find a color space which is invariant to illumination conditions [3].

There are two prominent approaches to skin segmentation practiced by researchers in this discipline; pixel based and region based. Pixel-based method classifies each pixel as skin or non-skin individually, independently from its neighbors. Methods utilizing color falls into this category. The region based method tries to take the spatial arrangement of skin pixels into account during the detection stage to enhance the performance. Region-based methods rely on additional knowledge such as texture of the color being investigated.

Skin color segmentation can be defined as the process of discrimination between skin and non-skin pixels. However, ambient light, shadows and the non-uniformity of imaging sensors in different cameras produce different tones that would result in different skin tones for the same person at different times. This makes it important that skin color determination is carried out in an appropriate color space where illumination or lighting conditions does not affect the decision making. Furthermore, due to variety of different skin colors from different parts of the world, it would be intriguing to see whether skin segmentation could be effectively carried at all using machine vision. The following section on color spaces will answer these questions.

3.1.2 Appropriate Color Space for Skin Segmentation

RGB is the most prominent additive color space consisting of Red (R), Green (G) and Blue (B) channels. These channels are highly correlated and contain luminance or brightness information along with the chrominance value. However, due to the presence of luminance information in each channel, any color observed does not linearly correspond to human perception. In other words, due to presence of luminance, two slightly different colors (R, G, B combined) with different luminance may appear to be the same. As was shown in Fig. 3.1, RGB color space skin color for different ethnicities would spread so widely that its use in skin segmentation in the presence of other objects would be questionable.

There are other classes of color spaces in existence because of Television transmission. The YUV contains Y luminance and U and V chrominance components. Unlike RGB, Y contains the entire luminance component making U and V independent or invariant to illumination. YIQ is a similar color space which is used in NTSC Television format. YC_bC_r carries similar information to that of YUV and is used in JPEG based image compression standard. Figure 3.1 shows the benefit of using these color spaces opposed to RGB as they provide compact clusters invariant to ethnic background that would facilitate simpler classification approach [5–9].

Perceptual color spaces which have been developed the way how artists describe color, and its properties have also been used for skin segmentation research. Color spaces such as HSI, HSV and HSL are commonly used as they are much closer to human perception than the television broadcast related color spaces. Hue (H) has been described as the color and Saturation (S) which describes how ‘pure’ the color and brightness (I, V or L). HSV can be mapped from RGB using nonlinear mapping. Similar to YUV approach, H and S values are used for skin segmentation where intensity or the brightness value is disregarded to remove the sensitivity of the illumination on skin segmentation results [5, 9].

Such complexities can only be overcome if an approach can be devised where skin segmentation is invariant to most of these variables yet resulting in an acceptable discriminatory power of skin vs non-skin regions in an image. The answer lies in some color spaces other than the most common RGB. Red, Green and Blue (RGB) color space is the most common color space used to represent images. RGB is an additive color space with Red, Green and Blue components carrying highly correlated information. John and Rehg [10] and Brand and Mason [11] have demonstrated that skin segmentation is possible in RGB space. However, there is overwhelming evidence that suggests RGB color space is not effective for skin segmentation for variety of skin color from different parts of the world. Researchers had proposed using normalized RGB to obtain chromaticity information to classify skin pixels effectively. However, normalized RGB is plagued by uneven illumination [12–15]. The skin segmentation thresholds for RGB are given by Kovac et al. [16]:

For uniform daylight illumination:

$$\begin{gathered} R > 95, G > 40, B > 20 \\ Max\{R, G, B\} - min\{R, G, B\} < 15 \\ |R - G| > 15, R > G, R > B \end{gathered}$$

Flashlight or daylight lateral illumination:

$$\begin{gathered} R > 220,\ G > 210,\ B > 170 \\ |R \text{ - } G| \leq 15,\ B < R,\ \ B < G. \end{gathered}$$

3.1.2.1 Normalized RGB

There have been efforts to remove discrepancies observed when different color combinations with varying intensity appearing similar in RGB space. One such suggestions is normalized color space given by following expressions:

$$r = \frac{R}{R+G+B}, g = \frac{G}{R+G+B}, b = \frac{B}{R+G+B} \tag{3.1}$$

Here $r = 1 - g - b$ due to normalization. Hence, determining any two normalized colors will completely define the color space. Gomez and Morales used a constructive induction approach to determine the skin map [17, 18]. Using the normalized RGB values they determined that the following thresholds resulted in best skin segmentation performance:

$$\frac{r}{g} > 1.185, \frac{r \cdot b}{(r+g+b)^2} > 0.107 \text{ and } \frac{r \cdot g}{(r+g+b)^2} > 0.112$$

3.1.2.2 YCbCr, YUV and YIQ

Due to the linear nature of transformation between RGB and YC_bC_r, YC_bC_r is often used in people surveillance and skin color segmentation [19–22]. The YC_bC_r values are less computationally intensive to achieve compared to the HSV values and are computed as follows:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 65.481 & 128.553 & 24.966 \\ -37.797 & -74.203 & 112 \\ 112 & -93.786 & -18.214 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{3.2}$$

The Y, U, V and YIQ values are similarly calculated from RGB using a linear conversions:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.14713 & -0.28886 & 0.436 \\ 0.615 & -0.51499 & -0.10001 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}, \tag{3.3}$$

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.595716 & -0.274453 & -0.321263 \\ 0.211456 & -0.522591 & 0.311135 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \tag{3.4}$$

As shown if Fig. 3.2, a hand gesture looks different in different color spaces. Yet, YC_bC_r offers the ability to separate skin tone from non-skin regions as shown in Fig. 3.3.

YC_bC_r thresholds for skin segmentation are:

$$77 \le C_b \le 127 \text{ and } 133 \le C_r \le 173.$$

3.1.2.3 HSV, HIS, HSL—Hue, Saturation and Intensity (Value, Lightness)

Researchers have devised HSV (Hue Saturation and Value) and YC_bC_r color space to separate luminance and chrominance information. This separation of brightness information from chrominance leads to reduction in uneven illumination [23]. The HSV values are derived using the following expressions using RGB components:

$$H = \arccos \frac{\frac{1}{2}((R-G)+(R-B))}{\sqrt{(R-G)^2 + (R-B)(G-B)}} \tag{3.5}$$

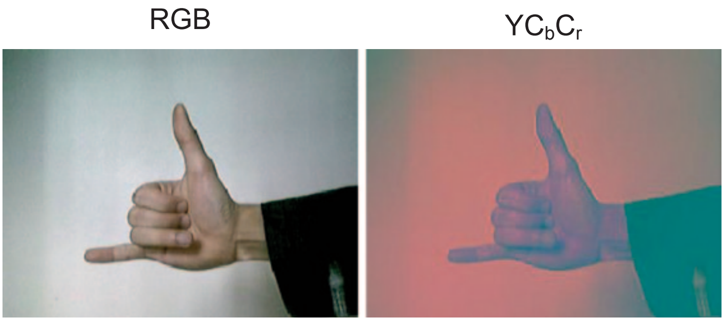


Fig. 3.2 Hand gesture in RGB and YC_bC_r color spaces

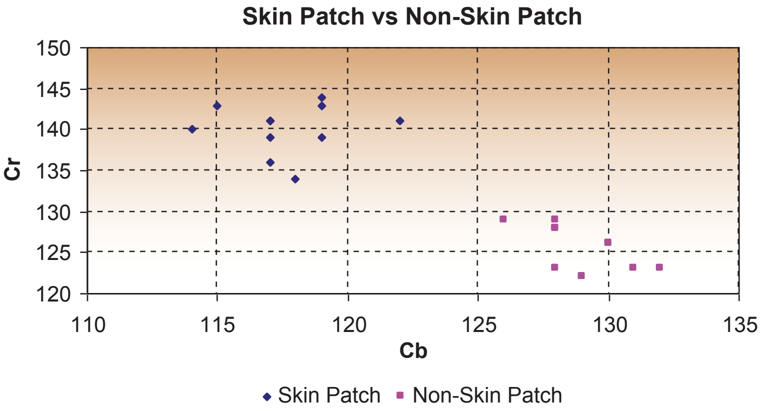


Fig. 3.3 Correlation between C_r and C_b for Skin Patch and Non-Skin patch pixels

$$S = 1 - 3\frac{\min(R,G,B)}{R+G+B} \tag{3.6}$$

$$V = \frac{1}{3}(R+G+B). \tag{3.7}$$

Tsekeridou and Pitas [18, 24], have obtained thresholds for skin segmentation using the following thresholds:

$$V \geq 40;$$
$$0.2 < S < 0.6;$$
$$0^{\circ} < H < 25^{\circ} \text{ or } 335^{\circ} < H < 360^{\circ}.$$

Starting from a training data set composed of skin color samples, Garcia and Tziritas computed the color histogram in HSV color space, and estimated the shape of the skin color cluster [18, 25]. They found a set of planes by successive adjustments depending on segmentation results, developing the thresholds shown below which define six bounding planes found in the HSV color space case, where $H \in \left[-180^{\circ}\ 180^{\circ}\right]$:

$$\begin{aligned}
&V \geq 40; \\
&H \leq (-0.4V + 75) \\
&10 \leq S \leq (-H - 0.1V + 110) \\
&\mathit{if}\ H \geq 0 \quad S \leq (0.08(100 - V)H + 0.5V) \\
&\mathit{if}\ H < 0 \quad S \leq (0.5H + 35).
\end{aligned}$$

Hue-saturation based color spaces stems from the humans desire to numerically specify the notions of tint, saturation and tone. Hue represents the dominant color (as in dominant wavelength) whereas saturation defines the 'colorfulness' of an area with respect to its brightness [26]. The amount of light or luminance, historically measured in lux, has lead to the notions of 'intensity', 'lightness' or 'value'. The user is directed to the following references for deeper notions of color spaces in skin segmentation [27–31].

There are direct relationships among the brightness and the chrominance values which attempt to conceal the chrominance information. In 1999, Fleck et al. developed an alternative way of hue and saturation computation using log opponent values to reduce the dependence of chrominance on the illumination levels [32].

The polar coordinate system of Hue-Saturation spaces, as shown in Eq. 3.5, results in a cyclic form. This is inconvenient color space for parametric skin color models that need tight cluster of skin colors for best performance. A different representation of Hue-saturation using Cartesian coordinates can be used [19, 33]:

$$X = S\cos H, \qquad Y = S\sin H$$

HSL and HSV are the two most common cylindrical-coordinate representations of points in an RGB color model. The two representations rearrange the geometry of RGB in an attempt to be more intuitive and perceptually relevant than the cartesian (cube) representation. Developed in the 1970s for computer graphics applications, HSL and HSV are used today in color pickers, in image editing software, and less commonly in image analysis and computer vision [34]. The relationship between RGB and HSL, and HSV are as follows:

$$\begin{aligned}
M &= \max(R, G, B) \\
m &= \min(R, G, B) \\
C &= M - m
\end{aligned} \tag{3.8}$$

$$H' = \begin{cases} \mathit{undefined}, & \mathit{if}\, C=0 \\ \dfrac{G-B}{C} \bmod 6, & \mathit{if}\, M=R \\ \dfrac{B-R}{C}+2, & \mathit{if}\, M=G \\ \dfrac{R-G}{C}+4, & \mathit{if}\, M=B \end{cases} \tag{3.9}$$

$$H = H' \times 60^{o} \tag{3.10}$$

$$\begin{aligned} I &= \frac{1}{3}(R+G+B) \\ V &= M \\ L &= \frac{1}{2}(M+m) \end{aligned} \tag{3.11}$$

$$\begin{aligned} S_{HSV} &= \begin{cases} 0, & \mathit{if}\ C=0 \\ \dfrac{C}{V}, & \mathit{otherwise} \end{cases} \\ &\mathit{or} \\ S_{HSV} &= \begin{cases} 0, & \mathit{if}\ C=0 \\ 1-\dfrac{m}{I}, & \mathit{otherwise} \end{cases} \end{aligned} \tag{3.12}$$

$$S_{HSL} = \begin{cases} 0, & \mathit{if}\ C=0 \\ \dfrac{C}{1-|2L-1|}, & \mathit{otherwise} \end{cases} \tag{3.13}$$

3.1.2.4 TSL—Tint, Saturation and Lightness

A normalized chrominance-luminance TSL space is a transformation of the normalized RGB into more intuitive values, close to hue and saturation in their meaning [19].

$$\begin{aligned} S &= \left[9/5(r'^2+g'^2)\right]^{1/2} \\ T &= \begin{cases} \arctan(r'/g')/2\pi+1/4, & g'>0 \\ \arctan(r'/g')/2\pi+3/4, & g'>0 \\ 0, & g'=0 \end{cases} \\ L &= 0.299R+0.587G+0.114B \end{aligned} \tag{3.14}$$

where $r'=r-1/3$, $g'=g-1/3$ and r, g are defined as in Eq. 3.1 [19]. Terrillon et al. [35] have compared nine different color spaces for skin modelling with a unimodal Gaussian joint probability density functions (only chrominance components of the color spaces were used). They argue that normalized TSL space is superior to other color spaces for this task.

3.1.2.5 CIELAB Color Space

CIELAB color space has been devised to be perceptually uniform color space. According to Poynton et al., perceptual uniformity refers to "*Digital image representation is perceptually uniform if a small perturbation of a component value—such as the digital code value used to represent red, green, blue, or luminance—produces a change in light output at a display that is approximately equally perceptible across the range of that value*" [36]. Hence uniform color spaces were defined in such way that all the colors are arranged by the perceptual difference of the colors. However, the perceptual uniformity in these color spaces is obtained at the expense of heavy computational transformations. As shown in Eqs. 3.15, 3.16 and 3.17, the computation of the luminance (L) and the chroma (a, b) is obtained through a non-linear mapping of the XYZ coordinates [37]. CIE (Commission International d'Eclairage) specifies three: CIE*XYZ, CIE*Lab, and CIE*Luv. In CIE*Lab or CIELab, the three components represent luma or luminance (or illumination) component and *ab* represent the chroma or color information [38]. The relationship between RGB, and XYZ and a, b components are:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4125 & 0.3576 & 0.1804 \\ 0.2127 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9502 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{3.15}$$

$$L = \begin{cases} 116\left(\frac{Y}{Y_n}\right)^{\frac{1}{3}} - 16 & if\ \frac{Y}{Y_n} > 0.008856 \\ 903.3 & otherwise \end{cases} \tag{3.16}$$

$$a = 500\left[\frac{X^{\frac{1}{2}}}{X_n} - \frac{Y^{\frac{1}{3}}}{Y_n}\right], \qquad b = 200\left[\frac{Y^{\frac{1}{2}}}{Y_n} - \frac{Z^{\frac{1}{3}}}{Z_n}\right]. \tag{3.17}$$

The threshold values for skin segmentation under CIE LAB are: [39]

$a_{\max}=14$, $a_{\min}=2$, $b_{\max}=18$, $b_{\min}=0.7$. Figure 3.4 depicts the results of skin segmentation under different color spaces.

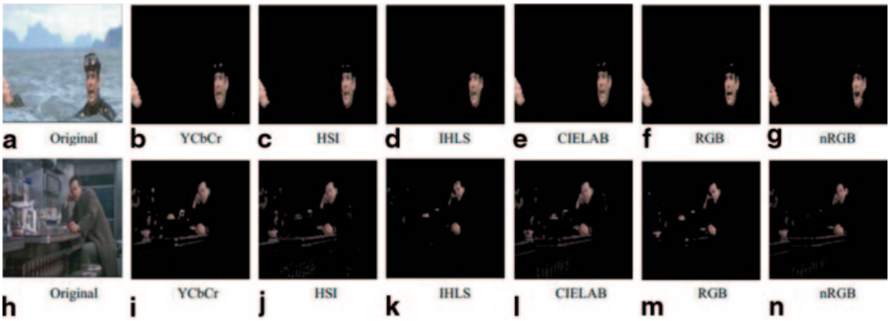


Fig. 3.4 Example results of skin detection using static skin filters in different color spaces. *Black* shows non-skin. (Courtesy of [39])

The goal of skin segmentation is the rapid decision making of skin vs non-skin regions. This can be accomplished by a set of rules which would define valid regions for skin in different color spaces. In the previous sections, for each color space, skin color thresholds were presented that were developed using extensive research over the years.

Pixel Based Skin Classification Using Non-parametric Skin Modelling The features used in skin classification are the values from color spaces. The problem then reduces to identifying a test pixel falls into the compact boundary or outside. Brand and Mason [40] constructed a simple one-dimensional skin classifier which would asses if the ratio between R and G channels falls in between particular upper and a lower bound. There are other approaches where the skin color region in a two dimensional color space (U, V or C_b, C_r, etc.) is modelled using an elliptical boundary model [41]. The model parameters are estimated with the help of a large skin patch database.

There are other classification strategies using Bayesian probabilistic approaches using the knowledge of statistics. The classification relies on finding the $P(skin|color)$ which is the probability of any *color* pixel being skin. This information is almost impossible to be determined given that any color space having extremely high number of colors. However, by rewriting this expression using the following way simplifies the problem:

$$P(skin \mid color) = \frac{P(color \mid skin)P(skin)}{P(color \mid skin)P(skin) + P(color \mid nonskin)P(nonskin)}.$$

Since finding information of $P(\text{color}|skin)$ can be achieved using information gathered by recording human skin color from every part of the globe. Also the denominator signifies the total probability of observing color which does not affect the classification as it is a constant. Therefore the problem reduces to finding $P(skin|color)$ which can be estimated using histograms [13, 20, 28, 42–44], mixture of Gaussian models [30, 45] to approximate probability density functions.

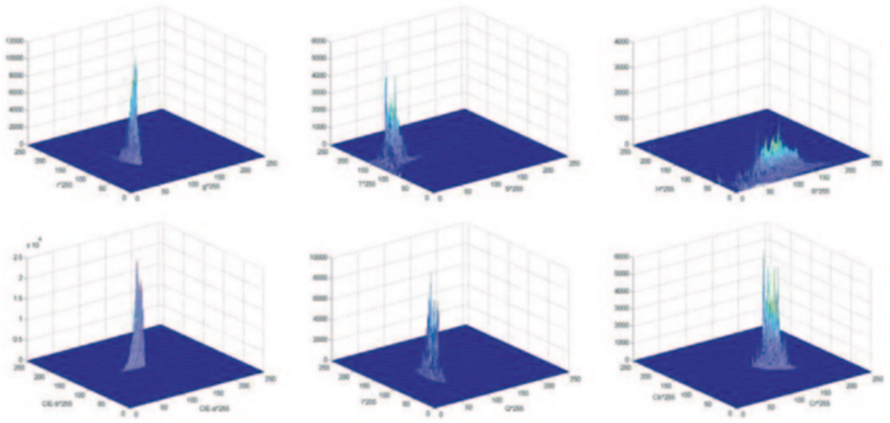


Fig. 3.5 Cumulative histograms of the training skin color pixels in different chrominance spaces: normalized r-g, T-S, H-S, CIE-ab, I-Q, Cb-Cr [47]

3.1.2.6 Region Based Skin Segmentation

Scientists overwhelmingly agree that for effective skin segmentation, it is natural to treat skin or non-skin as regions instead of individual pixels [45]. This would reduce the amount of noise that is present when isolated skin-tone ‘patches’ are erroneously classified as skin. Some of the early work on region based skin segmentation was reported by Yang and Ahuja on Gaussian mixture model for skin classification opposed to the predominantly simple thresholding or a single Gaussian distribution to characterize the properties of skin color [45]. They used multiscale segmentations to find elliptical regions for face detection. Hence, their model is biased toward elliptical objects. Kruppa et al. proposed a simple generative skin patch model combining shape and color information [46]. Their model was parametric and represented the spatial arrangement of skin pixels as compact elliptical regions. Those parameters were estimated by maximizing the mutual information between the model-generated skin pixel distribution and the distribution of skin color as observed in the image.

As shown in Fig. 3.5, histograms can be developed for different color spaces using variety of skin tones representing variety of human races from the world for an effective classifier [47]. Such knowledge can then be used effectively for skin segmentation as shown in Fig. 3.6 [47].

Poudel et al. proposed a segmentation technique based on the notion of superpixel [48–50], to group similar color pixels together. Each superpixel was classified as skin or non-skin by aggregating pixel-based evidence obtained using a histogram based Bayesian classifier similar to [11].

The result was further improved with Conditional Random Field (CRF), which operate over superpixels instead of pixels. Even though the segmentation cost is an overhead over the pixel-based approach, it greatly reduces the processing cost further down the line, such as smoothing with CRF. Furthermore, aggregation of pixels into regions helps to reduce local redundancy and the probability of merging

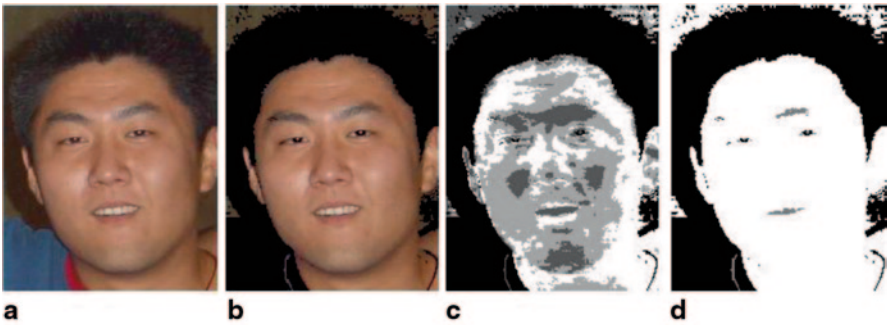


Fig. 3.6 An example of image segmentation. (**a**) The original image, (**b**) the result after pre-processing, (**c**) the result of the original FCM, (**d**) the result of the improved FCM. (Courtesy of [47])

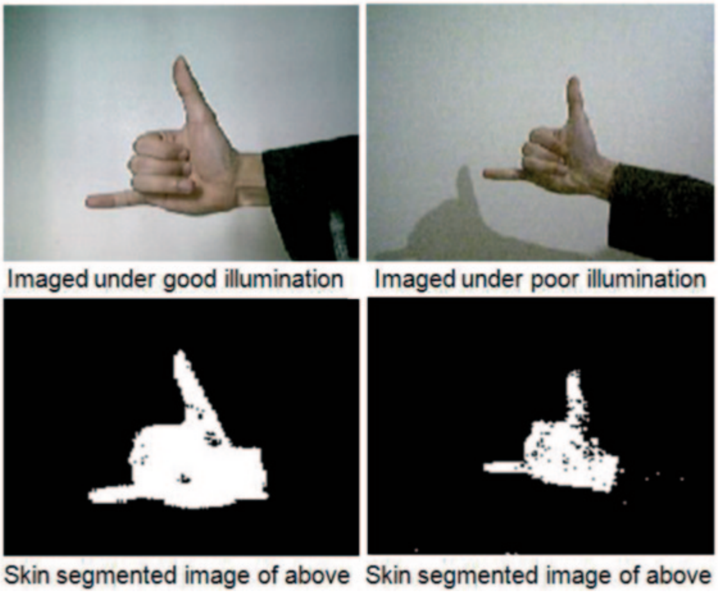
unrelated pixels [51]. Since superpixels preserve the boundary of the objects, it helps to achieve very accurate object segmentations [52]. Their method not only outperformed the current state-of-the-art pixel-based skin color detection methods but also extracted larger skin regions while still keeping the false-positive rate lower, providing semantically more meaningful results. This could in turn benefit higher-level vision tasks, such as face or hand detection.

3.2 Morphological Filtering

Computer vision relies on identifying shapes and structures in image acquisition. As was discussed in the section of skin segmentation, once a shape is isolated as a binary image with numerous imperfections, morphological filtering is commonly used to remove imperfections in shapes to understand the image content. In particular, the binary regions produced by simple thresholding are distorted by noise and texture. Morphological image processing pursues the goals of removing these imperfections by accounting for the form and structure of the image.

Morphological filtering is a broad set of non-linear image processing operations that can be used to process images based on shapes. These operations apply a structuring element of different shapes to an input image. The output image usually retains its original size. The structural element denotes the size of the window that would operate on a neighbourhood of a pixel to create the output. The size and shape of the neighbourhood can be chosen to construct a morphological operation that is sensitive to specific shape(s) in the input image.

Before the detailed theory of morphological filtering is discussed, it would be useful to see an example of a computer vision application in the context of computer human interaction to ascertain the usefulness of this process. Figure 3.7 shows that under good lighting conditions, the skin segmented hand gesture contain few noise patches. When the lighting deteriorates, the resulting thresholded image contains more noise patches as shown in Fig. 3.7 (right bottom). In order for computer vision



Imaged under good illumination Imaged under poor illumination

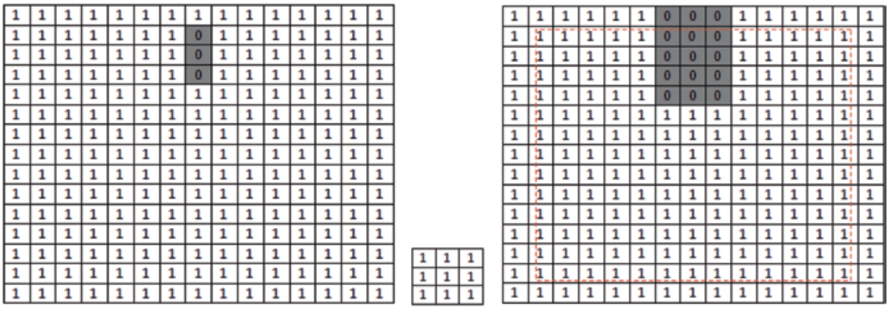
Skin segmented image of above Skin segmented image of above

Fig. 3.7 The above images show that under poor illumination, skin segmentation results in multiple undesired artefacts. Even the well-lit images produce undesirable regions as shown in images of *left*

system to be effective, the skin segmented extracted gesture should be solid white for further processing. Also, the noise spots shown in Fig. 3.7 (left bottom and right bottom) should be removed. The only operation that facilitates this requirement is morphological filtering as would be discussed next.

3.2.1 Basic Operations; Erosion and Dilation

Dilation and erosion are considered to be the most basic morphological operations. Dilation adds pixels to the boundaries of objects in an image, while erosion removes pixels on object boundaries. The size of the structuring element (SE) determines the number of pixels added or removed from the objects in an image. In dilation and erosion, the state of any given pixel in the output image is determined by applying a rule to the corresponding pixel and its neighbors in the input image [53, 54]. There are set rules that define the process either as dilation or erosion. The morphological filtering process is mostly binary in nature however; these operations can also be used on gray scale images. These operations can be applied on gray scale images when their light transfer functions are unknown and therefore their absolute pixel values are of no or minor interest. In binary operation, the outcome is either 1 (which is the highest intensity value) or 0 (which is lowest intensity possible). In dilation, the value of the output pixel is the maximum value of all the pixels in



1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

1	1	1
1	1	1
1	1	1

1	1	1	1	1	1	0	0	0	1	1	1	1	1	1
1	1	1	1	1	1	0	0	0	1	1	1	1	1	1
1	1	1	1	1	1	0	0	0	1	1	1	1	1	1
1	1	1	1	1	1	0	0	0	1	1	1	1	1	1
1	1	1	1	1	1	0	0	0	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Fig. 3.8 Binary image of size 15 × 15 is operated on with a structuring element which performs 'erosion' and the result is shown on the *right*. Only a 13 × 13 sized area contain the valid signal after erosion marked with *red broken line*

the input pixel's neighborhood. In a binary image, if any of the pixels is set to the value 1, the output pixel is set to 1. The erosion rule states that the value of the output pixel is the minimum value of all the pixels in the input pixel's neighborhood. In a binary image, if any of the pixels is set to 0, the output pixel is also set to 0. Figure 3.8 shows the operation of a structuring element of size 3 × 3 on a binary image of size 15 × 15. The outcome of this is shown in the right hand size matrix of Fig. 3.8. The 'red' broken line marks the boundary of the valid signal after the operation as outside of this region is considered invalid due to the size of the structuring element. Figure 3.9 shows the outcome using a 3 × 3 structural element. As can be seen, this leaves skin tone regions intact. Hence the size of the structural element is very important. The size of the structural element depends on the size of disconnected or noisy artefacts that remains after skin segmentation.

Figure 3.10 shows that the outcome of any morphological filtering is sensitive to the size of the structuring element as an inappropriate size would simply result in a more complicated image that a computer vision system is unable to utilize. Fairly large structural elements erode the information contained in the useful object such as skin segmented hand gesture. Only close observation of the objects to be preserved and removed would justify the size of the structural element.

3.2.1.1 Mathematical Definition of Morphological Filtering—Erosion and Dilation

Mathematically, erosion is defined for an Image *I* by a structural element *S* as follows:

$$I \ominus S = \left\{ I \middle| S_I \subseteq I \right\}$$

Where S_I refers to *S* translated with *I*.

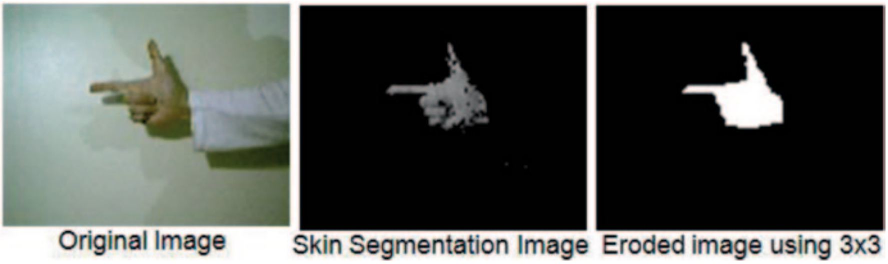


Fig. 3.9 Result of erosion using a 3 × 3 structural element

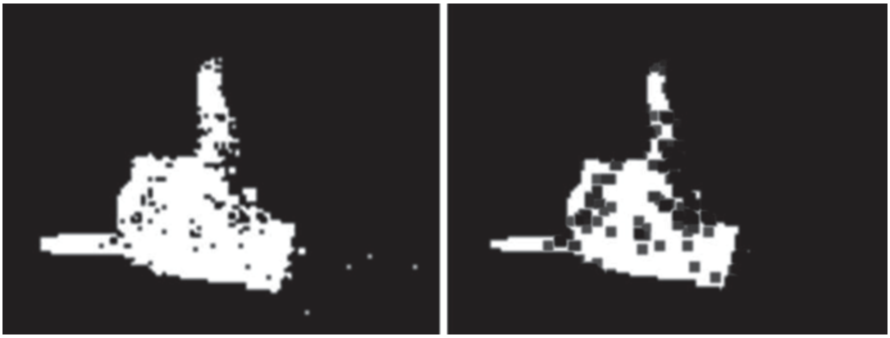
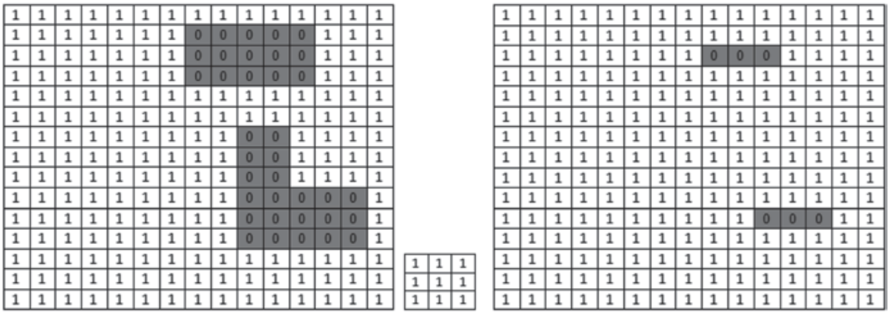


Fig. 3.10 Erosion of a noisy hand gesture using a structural element of size 7 × 7. Here the result shows large *square holes* in the resulting image signalling that the size of the structuring element is not appropriate for this operation

A complementary operation to that of erosion is dilation. It is defined simply as the erosion of the complement of a set. If I^c denotes the complement of I, then the dilation of a set I by a set S is denoted by $I \oplus S = (I^c \ominus S)^c$. This helps us to easily understand dilation in terms of erosion. Figure 3.11 shows the effect of dilation on a structure using a 3 × 3 structural element.

Figure 3.12 shows the outcome of 'filling' when dilating an eroded image. However, this process outline that dilation with larger structural elements will not necessarily fill image gaps. Morphological operations such as erosion and dilation can be performed on gray scale images as shown in Fig. 3.13 and 3.14. In Fig. 3.13, the result of erosion using a structural element of size 6 × 6 square results in disfiguring the letters and darkening the image. On the contrary, dilation result in similar disfigurement of lettering yet, lightening the image as shown in Fig. 3.14.

Processing Pixels at Image Borders (Padding Behavior) In morphological filtering, origin of the structuring element is centred over the pixel of interest in the input image. For pixels at the edge of an image, parts of the neighborhood is defined by the amount that structuring element can extend beyond the border of the image.



1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	0	0	0	0	0	1	1	1
1	1	1	1	1	1	1	0	0	0	0	0	1	1	1
1	1	1	1	1	1	1	0	0	0	0	0	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	0	0	1	1	1	1
1	1	1	1	1	1	1	1	1	0	0	1	1	1	1
1	1	1	1	1	1	1	1	1	0	0	1	1	1	1
1	1	1	1	1	1	1	1	1	0	0	0	0	0	1
1	1	1	1	1	1	1	1	1	0	0	0	0	0	1
1	1	1	1	1	1	1	1	1	0	0	0	0	0	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

1	1	1
1	1	1
1	1	1

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	0	0	0	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	0	0	0	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Fig. 3.11 Results of dilation using a 3 × 3 element. See that the *vertical line* has completely disappeared as its width was less than the width of the structural element

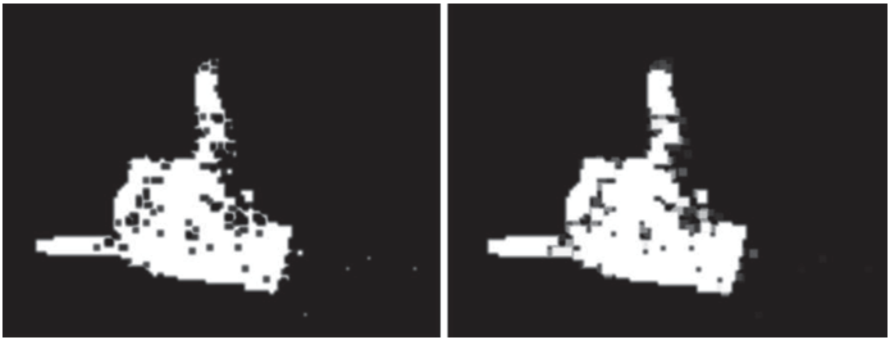


Fig. 3.12 Dilation of an eroded image with a structural element of size 5 × 5 (*left*) and 7 × 7 (*right*)

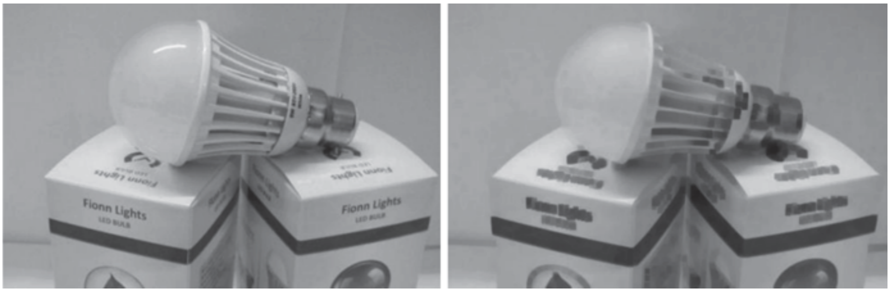


Fig. 3.13 Erosion of a gray scale image by a 6 × 6 structural element. Original (*left*), eroded (*right*). The image content is much darker after erosion

To process border pixels, the morphological functions assign a value to these undefined pixels as if the functions had padded the image with additional rows and columns. The value of these padding pixels varies for dilation and erosion operations. Pixels beyond the image border are assigned the minimum value afforded by

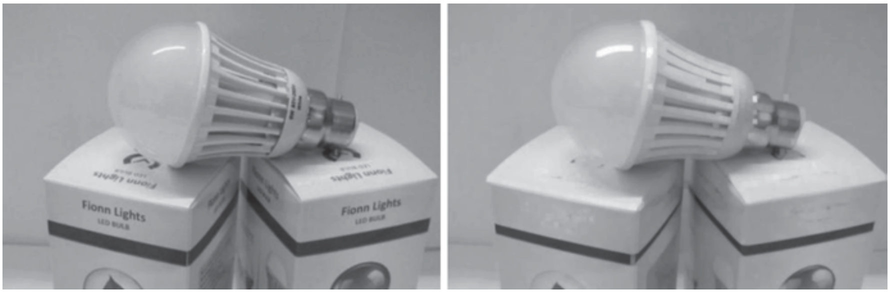


Fig. 3.14 Dilation of grayscale image by a 6 × 6 structural element. The image is lighter than before after dilation. Original (*left*) and the dilated image (*right*)

the data type. For binary images, these pixels are assumed to be set to 1. For gray scale images, the maximum value for uint8 images is 255. For dilation of binary images, these pixels (padding pixels) are assumed to be set to 0 whereas for gray scale images, the minimum value for uint8 images is 0.

3.2.2 Opening and Closing

Erosion and dilation are used in many other morphological filtering to achieve different outcomes for computer vision applications. Hand gesture recognition in its binary representation usually result in many holes and noisy unconnected artefacts. These areas need to be filled up producing solid gestures while removing the artefacts without affecting the gesture.

Opening can be described using more fundamental operations. Opening is so called because it can open up a gap between objects connected by a thin bridge of pixels. In this case, the dilation and erosion should be performed with a structuring element that has been rotated by 180°. Typically, the structural elements are symmetrical, so that the rotated and initial versions of it do not differ. Any regions that have survived the erosion are restored to their original size by the dilation. All pixels which can be covered by the SE with the SE being entirely within the foreground region will be preserved. All foreground pixels which cannot be reached by the structuring element without lapping over the edge of the foreground object will be eroded away. Opening is idempotent which refers to the fact that repeated application has no further effects.

Closing is the operation of filling holes in the regions while keeping the initial region sizes. In other words, closing (opening) of a binary image can be performed by taking the complement of that image, opening (closing) with the structuring element, and taking the complement of the result. The formal mathematical definitions of opening and closing are defined next.

Opening Opening is performed by erosion followed by dilation resulting in eliminating protrusions and smoothing contours. Both of these operations are attempted

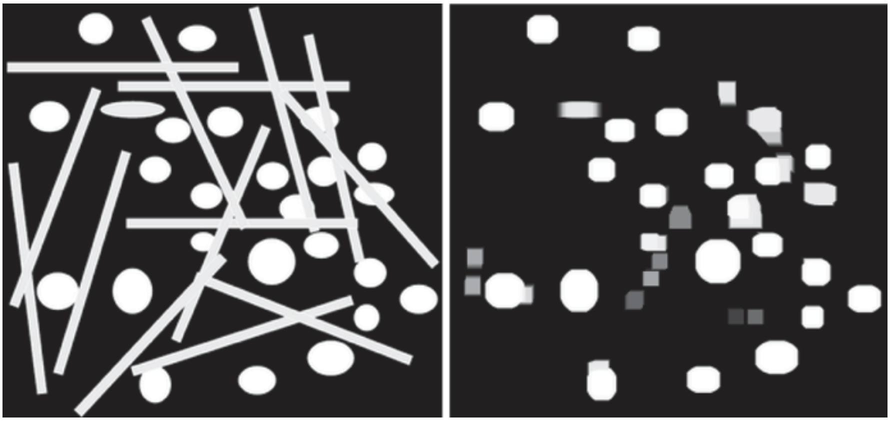


Fig. 3.15 Opening with a 10 × 10 square structuring element

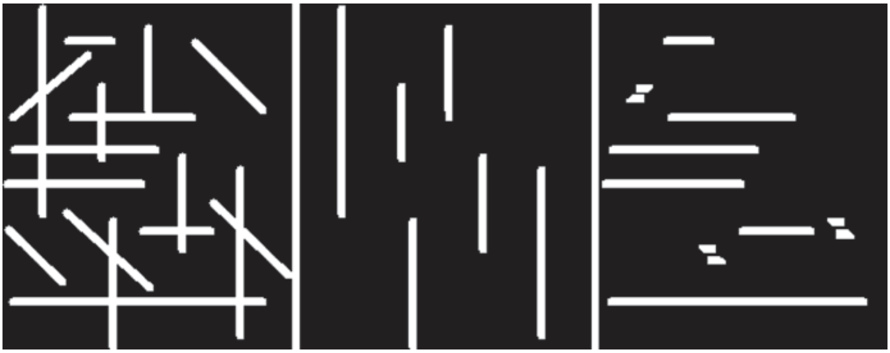
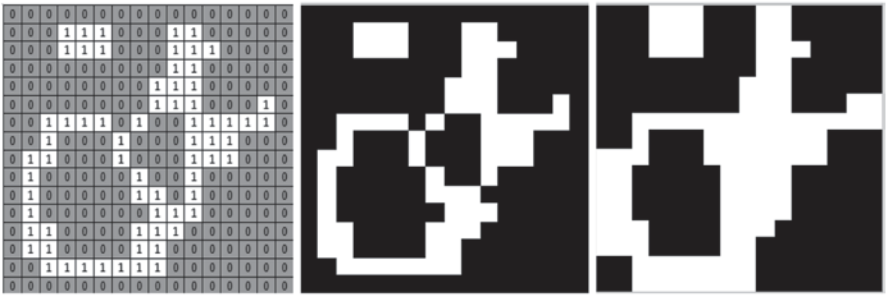


Fig. 3.16 Opening the image on the *left* with a 3 × 9 structuring element (result shown in the *middle*), opening with 9 × 3 structuring element (*right*)

using the same structural element. The mathematical symbol of opening is '∘' and the definition opening using erosion and dilation is given by:

$$I \circ S = (I \ominus S) \oplus S$$

Opening is known as a filtering mechanism to remove clutter to enhance image intelligibility especially for computer vision. As shown in Fig. 3.15, using a specific type of structuring element with specific size, the long thin objects are removed from the image. This would be advantageous for removing clutter for medical diagnosis or counting certain type of objects removing unnecessary ones. The effect of the choice of the structuring element size is illustrated in Fig. 3.16. A SE of size 3 × 9 will result in leaving vertical bars intact and the 9 × 3 will remove the vertical bars leaving only the horizontal ones.



0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0	1	1	0	0	0	0	0
0	0	0	1	1	1	0	0	0	1	1	1	0	0	0	0
0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	0	0	0	1	0
0	0	1	1	1	1	0	1	0	0	1	1	1	1	1	0
0	0	1	0	0	0	1	0	0	0	1	1	1	0	0	0
0	1	1	0	0	0	1	0	0	0	1	1	1	0	0	0
0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0
0	1	0	0	0	0	0	1	1	0	1	0	0	0	0	0
0	1	0	0	0	0	0	0	1	1	1	0	0	0	0	0
0	1	1	0	0	0	0	1	1	1	0	0	0	0	0	0
0	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0
0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 3.17 Closing a 16 × 16 image with a 3 × 3 square structuring element. The figure on the *left* shows an image as matrix with '1' associated with *white* and '0' associated with *black*. The operations are performed on the host image in the *middle* with the results shown on the *right*

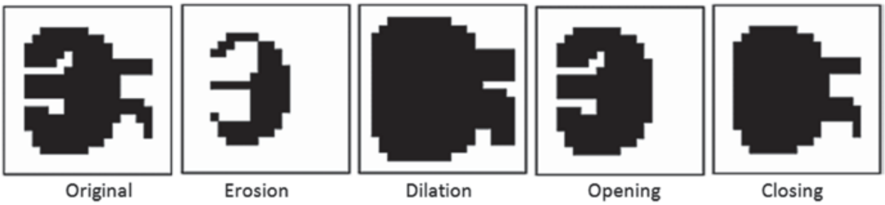


Fig. 3.18 Comparison of different processes of fundamental morphological filtering with an illustration of their use on a binary image, courtesy of [55]

Closing Closing is performed using dilation followed by erosion resulting in smoothing contours and fusing narrow breaks and long thin gulfs. This eliminates small holes and fills gaps in contours. As in opening, same structural element is used for both dilation and erosion. It would be interesting to understand the closing process as a structural element operates on the host image. For the initial dilation, the SE slides around outside each foreground region. All background pixels which can be covered by the SE with the SE being entirely within the background region will be preserved. All background pixels which cannot be reached by the structuring element without lapping over the edge of the foreground object will be turned into foreground. This scenario is illustrated in Fig. 3.17 when operated on by a 3 × 3 square structuring element. Opening is also known to idempotent as Opening. The symbol of closing is '$\bullet$' and is defined using dilation and erosion as follows:

$$I \bullet S = (I \oplus S) \ominus S.$$

The morphological operations described so far can be compared with each other based on their effect on the host image as shown in Fig. 3.18. Hand gesture recognition research relies heavily on these fundamental operations when using computer vision to register gestures. This chapter will further discuss other morphological operations such as hit and miss transform, thickening, thinning followed by skeletonization as they are commonly used in hand gesture recognition research.

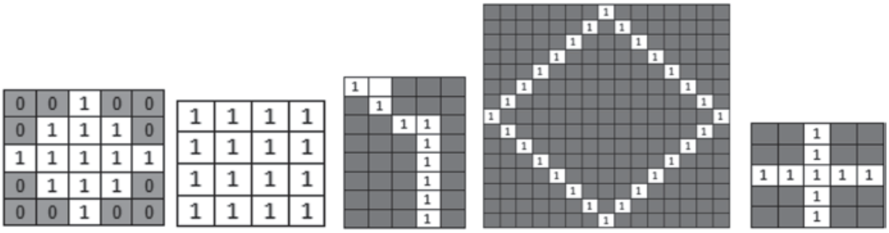


Fig. 3.19 Variety of structuring elements; disc, Square, irregular and asymmetric, very large structuring element and a cross. The *darkened squares* contain zero

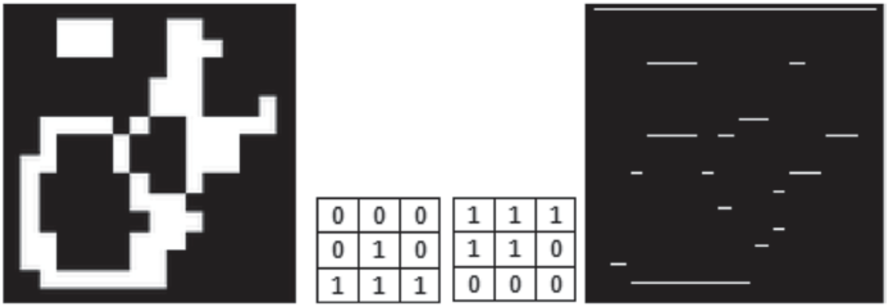
3.2.3 Structuring Element (SE)

A structuring element is a matrix consisting of only 0's and 1's that can have any arbitrary shape and size. The pixels with values of 1 define the neighborhood. One dimensional or two dimensional structuring elements are typically much smaller than the image being processed. The center pixel of the structuring element is known as the origin which identifies the pixel being processed. The pixels in the structuring element containing 1's define the neighborhood of the structuring element. 3D structuring elements use 0's and 1's to define the extent of the structuring element in the x- and y- axes with z signifying height values to define the third dimension. The operation of morphological filtering on binary images can be better understood by considering compound operations like opening and closing as filters. Their resemblance to filters of shape, opening with a disc shaped structuring element which smooths corners from the inside and closing with a disc results in smoothing corners from the outside. They also can filter out any image details that are smaller in size than the structuring element (e.g. opening is filtering the binary image at a scale defined by the size of the structuring element). Only those portions of the image that fit the structuring element are passed by the filter; smaller structures are blocked and excluded from the output image. The size of the structuring element is most important to eliminate noisy details but not to damage objects of interest.

The structuring elements do not have much restriction apart from the fact that they should not increase the energy of the resulting process. Any shape and size can be selected for structuring element. However, it would be advantageous to select a shape that would easily achieve the purported purpose in the morphological process. Some of the different shapes used are shown in Fig. 3.19.

3.2.4 Hit-and-miss Transform

Hit-and-miss Transform is used to look for particular patterns of foreground and background pixels for very simple object recognition. It is well-known that all other morphological operations can be derived from it [57–59]. The transform operates by assessing whether the foreground and background pixels in the structuring



0	0	0
0	1	0
1	1	1

1	1	1
1	1	0
0	0	0

Fig. 3.20 Binary image developed in Fig. 3.17 operated on by two transforms to achieve the result shown on *right*

1	1	1
1	1	0
0	0	0

x	1	X
0	1	1
0	0	x

x	1	0
1	1	0
x	0	0

0	0	x
0	1	1
x	1	x

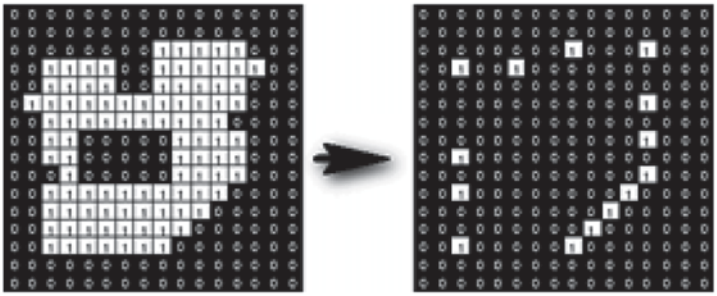
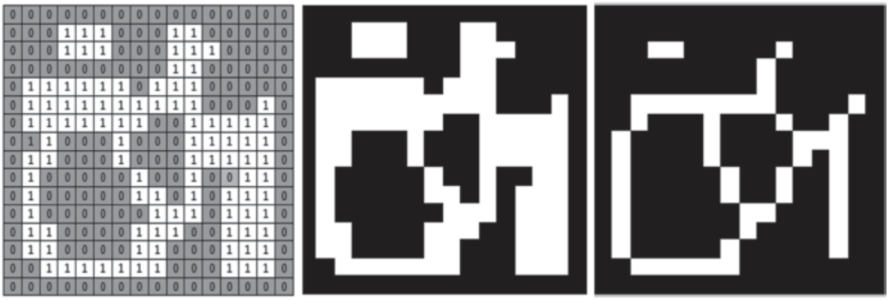


Fig. 3.21 Corner detection using hit-and-miss transform. The four transforms are shown on the *top* row with 'x' marking 'don't care' states. *Bottom* images show the original image transforming to corner detection where only the corner pixels remain

element exactly matches the foreground and background pixels in the image. If they match, then the pixel underneath the origin of the structuring element is set to the foreground color. The transform consists of 0's and 1's with usually a 1 at the origin. The transform matrix could also contain 'don't care' values which refers to either '0' or '1' which are not going to affect the outcome of the result significantly. An image can be operated on by more than one structural element one after the other. Figure 3.20 shows a binary image operated on by two structuring elements shown in the middle and the result on the right. Figure 3.21 shows how four miss-and-hit transforms can be used for corner detection on a binary image. In this, each transform operate on the input image and the results are 'OR'ed using logical processing to arrive at the final corner detection.



0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0	1	1	0	0	0	0	0
0	0	0	1	1	1	0	0	0	1	1	1	0	0	0	0
0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
0	1	1	1	1	1	1	0	1	1	1	0	0	0	0	0
0	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0
0	1	1	1	1	1	1	1	0	0	1	1	1	1	1	0
0	1	1	0	0	0	1	0	0	0	1	1	1	1	1	0
0	1	1	0	0	0	1	0	0	0	1	1	1	1	1	0
0	1	0	0	0	0	0	1	0	0	1	0	0	1	1	0
0	1	0	0	0	0	0	1	1	0	1	0	1	1	1	0
0	1	0	0	0	0	0	0	1	1	1	0	1	1	1	0
0	1	1	0	0	0	0	1	1	1	0	0	1	1	1	0
0	1	1	0	0	0	0	1	1	0	0	0	1	1	1	0
0	0	1	1	1	1	1	1	1	0	0	0	1	1	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 3.22 Thinning of a binary image. The image on the *left* shows the pixel arrangement where some regions are 4 pixels wide. The sections with 1 pixel width remain unchanged

3.2.5 *Thinning*

Morphological thinning is used to remove selected foreground pixels from binary images after edge detection where lines are often thicker than one pixel in width. Thinning will result in lines only one pixel wide. Hit-and-miss Transform can be used to perform thinning operation. In this approach, the effectiveness of thinning is determined by the structuring element [60, 61]. The mathematical definition of the thinning is given by the following relationship when using hit-and-miss transform:

$$Thin(I \text{ by } S) = I - HitandMiss(I, S)$$

Where logical subtraction is defined by A − B = A ∩ *NOT* B. The thinning of a binary image is shown in Fig. 3.22.

3.2.6 *Thickening*

Thickening is a morphological operation that is used to grow selected regions of foreground pixels in binary images similar to dilation or closing. It has several applications, including determining the approximate convex hull of a shape, and determining the skeleton by zone of influence [57–61]. Thickening is normally only applied to binary images, and it produces another binary image as output [58]. The definition of the Thickening can be given by the following relationship using Hit-and-Miss Transform:

$$Thicken(I \text{ by } S) = I \cup HitandMiss(I, S).$$

Thus the thickened image consists of the original image and any additional foreground pixels switched on by the hit-and-miss transform. Figure 3.23 shows the application of Thickening on a binary image.

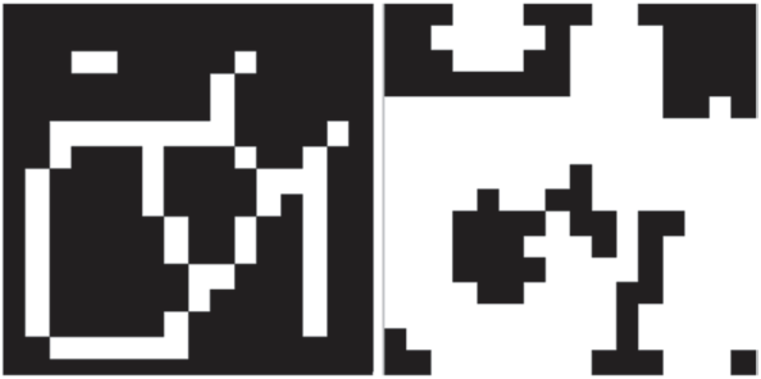


Fig. 3.23 Thickening of one pixel thick object (on the *left*). The result is shown on the *right*

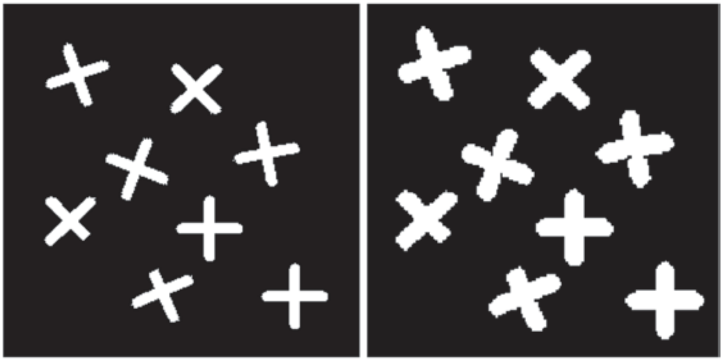


Fig. 3.24 Some objects retains shape if they are located adequately apart during the transformation

As was depicted in Fig. 3.23, the shape of the original object is somewhat obscured after thickening. However, this may not always be the case if the shapes are adequately located far apart and the SE is of specific size as shown in Fig. 3.24.

3.2.7 Skeletonization

Skeletonization is the process for reducing foreground regions in a binary image to a skeletal remnant that largely preserves the extent and connectivity of the original region. This in essence throws away most of the original foreground pixels. The skeleton is useful because it provides a simple and compact representation of a shape that preserves many of the topological and size characteristics of the original shape. This results in providing an approximate length of a shape by considering just the end points of the skeleton and finding the maximally separated pair of end

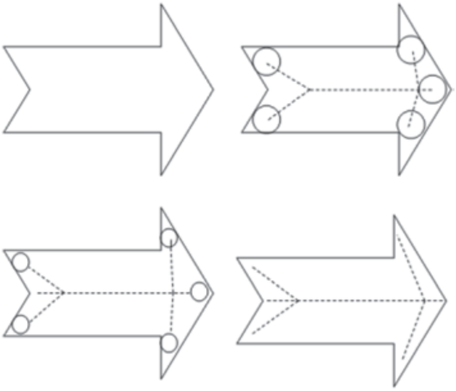


Fig. 3.25 Skeletonization is the process of continuously eroding of a structure (object) with ever decreasing structural element until it can be carried no further

points on the skeleton. Similarly, this will also lead to distinguishing many qualitatively different shapes from one another on the basis of how many ‘triple points’ there are (i.e. points where at least three branches of the skeleton meet).

Using the previous definition of erosion, skeletonization can be defined as the process where an object is eroded multiple times with ever decreasing size of structural element as follows:

$$Skeleton(I,S) = \bigcup_{k=0}^{K} I_k \ominus S_k$$

This process is illustrated in Fig. 3.25. Skeletonization is often used in text scanning to prune the thick edges so that optical character recognition and hand written recognition can be implemented in machines.

As with thinning, slight irregularities in a boundary will lead to spurious spurs in the final image which may interfere with recognition processes based on the topological properties of the skeleton. Figure 3.26 clearly illustrates this. Despurring or pruning can be carried out to remove spurs of less than a certain length but this is not always effective since small perturbations in the boundary of an image can lead to large spurs in the skeleton.

Skeletonization can result in a remarkable gesture identity if any gesture captured by camera can be turned into an accurate model. However, as was seen Fig. 3.26, skeletonization can result in much more complicated unintelligible realizations which offer no value for hand gesture recognition. However, recently, there have been few reported cases of research which were based on skeletonization of hand gestures for gesture recognition.

Ionescu and Coquin reported a hand gesture recognition method based on the 2D skeleton representation of the hand [61]. They represented each gesture with a hand skeleton and this skeleton was compared with a skeleton in a database for a match. They used Baddeley’s distance [62], as a measure of dissimilarities between model parameters. Even though the results were promising yet, they suffered from

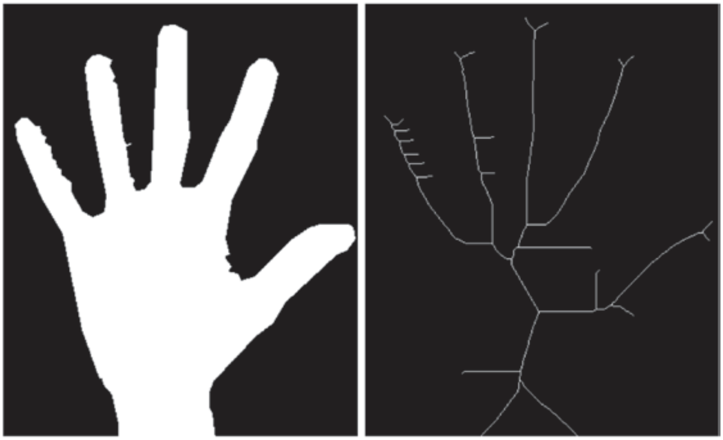


Fig. 3.26 Skeletonization in hand gesture recognition can sometimes lead to unforseen scenarios where even a slight imperfection on a binarized gesture can result in completely unintelligible results

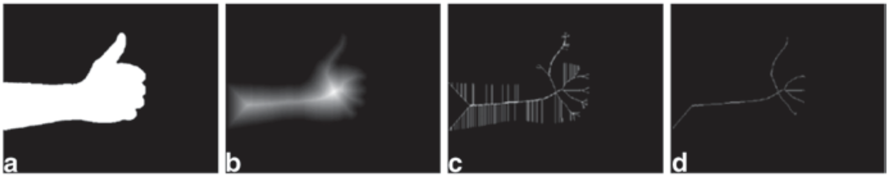


Fig. 3.27 Skeleton extraction: (**a**) hand region (binary image), (**b**) chamfer distance image (*white* corresponds to the greatest distance), (**c**) the skeleton obtained after connecting the centers of maximal discs, and (**d**) the skeleton obtained after spurious hole filling, pruning, and beautifying the previous skeleton. (Courtesy of [61])

occlusion and was limited to very few hand gestures. The directions of the camera were unconventional as it captured images from side which was unnatural for computer human interface as shown in Fig. 3.27.

Reddy et al. proposed an approach for calculating local orientation histograms of skeleton of the hand by using distance transformation techniques [63]. They relied on the local histograms as features due to their invariance to translation, rotation and scaling. Skeleton was computed for each and every hand posture in the entire hand motion and superimposed on a single image called as Dynamic Signature of the particular gesture type. Then the gesture was recognized by matching the image signature (features of local orientation) against the entries in the gesture alphabet. They used Image Euclidean distance measure as the metric to determine image similarities.

There are compelling reasons for using skeleton of the hand for gesture recognition. Skeletons provide compact representation of an object and preserve the topology of the object. Skeleton is robust against translation rotation and scaling

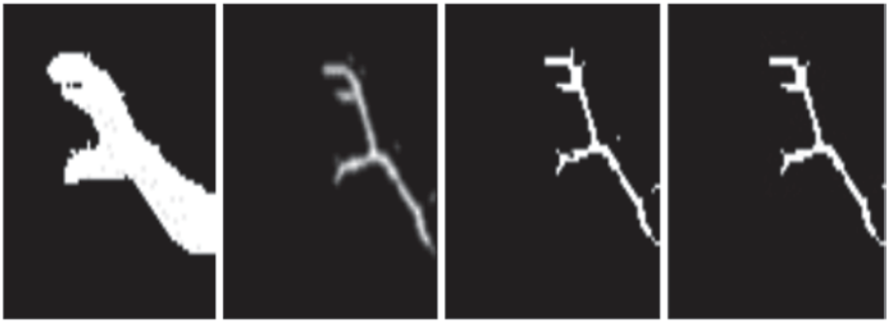


Fig. 3.28 Hand gesture skeletons for gesture recognition. (Courtesy of [63])

[64]. Skeleton is also extracted by using several methods such as chamfer distance transform [65], and morphological thinning [66] (Fig. 3.28).

Wu et al. in 2012 presented research using the skeleton of hand using depth information for hand gesture recognition [64]. They presented a method of recognizing hand gestures in the form of point clouds recorded by Kinect sensor. Initially, through Laplacian-based contraction and further processing, they extracted skeleton points from point clouds of hands. Then a novel partition-based descriptor and correspondence algorithm was applied to classify these skeletons and therefore to recognize gestures. In the process of recognition, the issue of scale variance and rotation variance were solved. They used 3D models downloaded from Princeton 3D Model Search Engine to be standard gestures, then record gestures using Kinect sensor. The recognition accuracy for 12 gestures was about 85 % on average. They finally verified their claims using performance analysis where the results proved both its accuracy and robustness. They demonstrated that skeleton-based method of recognition has great potential for further exploration. Figure 3.29 shows the stages of gesture skeletonization and their 12 gestures in Fig. 3.30.

3.3 Gesture Extraction Using Color and Depth Information

One of the major challenges in gesture recognition is to reliably capture the gesture alone from the clutter of the background. This is a non-trivial task as it has been shown over the years [68–74]. As was discussed in the previous section of skin segmentation, skin detection tries to separate the gesture from the background. However, this problem is compounded when the background contains skin-tone regions. Since the cameras are essentially 2D devices unlike the human eye, there is no information a camera can supply to separate hand gesture from another person in the background. However, if a stereo vision or another setup that detects

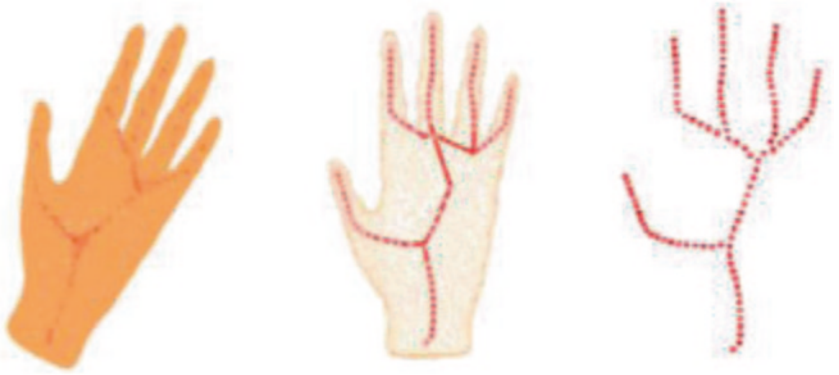


Fig. 3.29 Key points achieved from Laplacian-based contraction and index-based compaction. (Courtesy of [64])

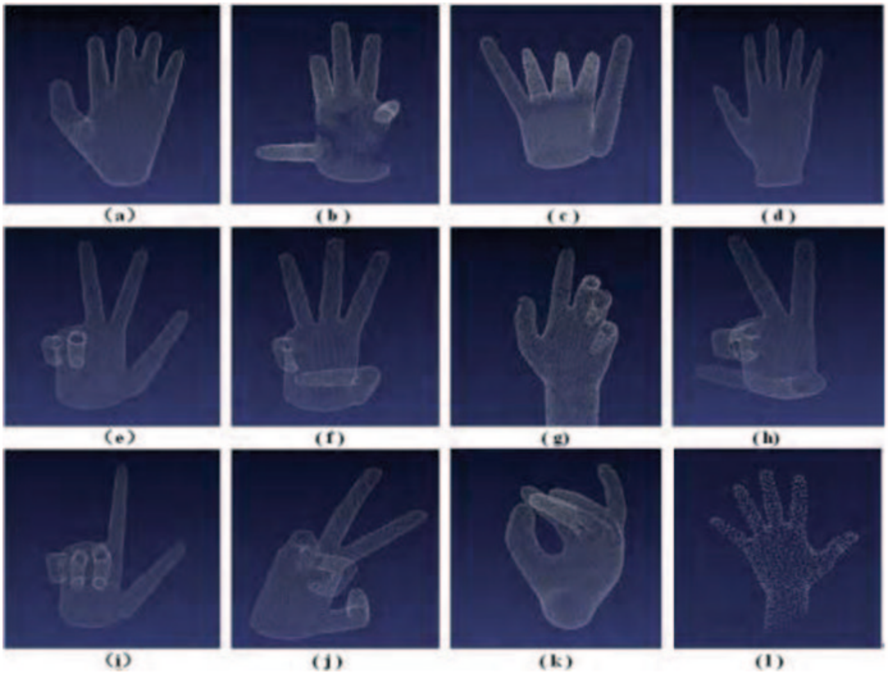


Fig. 3.30 Twelve cloud gestures used by Wu et al. (Courtesy of [67])

depth can be used, the complexity of the problem unravels as depth to the objects becomes available.

Yet, this section will discuss why this depth information alone is not reliable for background-foreground separation based on research carried out over the past 15 years. Very recently, there has been a glimmer of hope due to new breed of in-

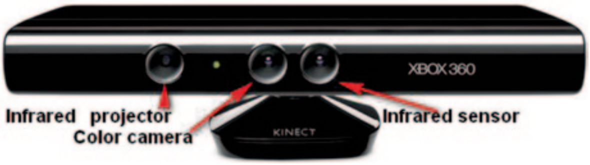


Fig. 3.31 Kinect Camera developed for Microsoft Xbox

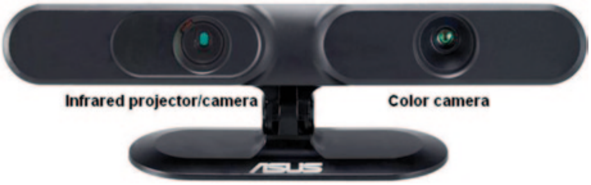


Fig. 3.32 Asus Xtion camera which has identical imaging capabilities to that of Kinect but with a personal computer compatible USB interface

expensive consumer grade cameras which are increasingly being used in an effort to retrieve depth information. Some of these devices are Kinect from Microsoft an Asus Xtion (both are manufactured by the same Taiwanese company with similar capabilities). Instead of stereo vision setup, these camera relies on infradred structured lighting projection and image capture through both infrared and color camera. The technology behind these cameras differs from the traditional depth camera; stereo vision. The novel technology is based on structured lighting which a well-understood phenomenon that is used in stereoscopy [75, 76]. The distortion pattern of the projected infrared structured light pattern is observed by the infrared camera to detect the depth to the objects and this information is fused with color image information so that every pixel has a depth parameter. Previously, non-stereoscopy systems relied on Time of Flight (TOF) cameras which have been confined to high end research due to their prohibitive cost. Currently, the next generation of Kinect camera is going to be released at the end of 2013 and is equipped with TOF technology opposed to Infrared light projection and the switch has been due to some of the limitations especially in resolution of infrared sensors compared to the CMOS imaging sensor.

Figure 3.31 depicts the Kinect camera with its onboard infrared projector, infrared receiver and color CMOS camera. The CMOS and the Infrared sensor both have a resolution of 640 × 480 at 30 fps. However, its depth perception is confined to 320 × 240. This results in many visual pixels not having proper depth information leading to edge anomalies in depth-color view. The Asus Xtion also has the same resolution in its sensors which is shown in Fig. 3.32. However, their physical appearance differ due to the Kinect having panning capability where as Xtion is simply has a front facing configuration.

Recently, there has been increased interest in applications of computer vision to traffic monitoring on highways to security surveillance in restricted areas. One of the preliminary tasks in such applications is to extract the foreground or objects from the background. Many early works relied on *background subtraction* which would simply look for the image difference before and after objects have

Fig. 3.33 Swiss Ranger 4000 by Mesa Imaging

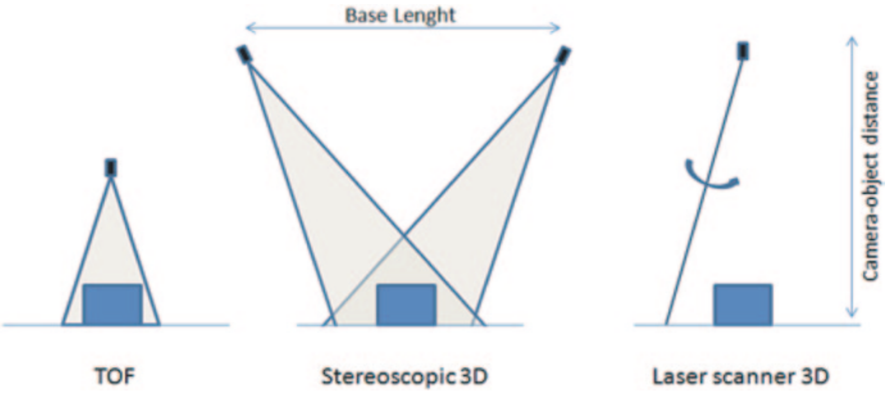


Fig. 3.34 TOF camera uses only one camera and needs a lesser distance from the camera to the object as shown on the *left*. Stereoscopy and laser scanners need more camera-object distance to be effective

been observed. As it turns out, same image sensor would produce slightly different picture with incrementally small color variations and noise when imaging an object few seconds apart. This problem is also compounded when natural lighting changes in day and temperature differences due to wind at night. Hence, simple subtraction of two images will not result in the foreground being revealed. It would contain undesirable sections of the background that would lead to false positives if decisions are made immediately without further elaborate processing. Such discrepancies in imaging sensors and technologies have called for more advanced hardware that would tackle some of the issues mentioned above.

Swiss high tech company Mesa Imaging had developed a TOF camera which dominated the market for many years for commercial imaging equipment that provided basic hardware as shown in Fig. 3.33. As shown in Fig. 3.34, TOF camera technology stands out from stereoscopy and laser scanning technology. Laser scanning technology has never been used for human surveillance as it is objectionable as a safe mode of information gathering due to high intensity lasers being used that

Fig. 3.35 Kinect II with its TOF technology

would cause eye damage. Stereoscopy devices need extensive special arrangement which is not suitable for the above applications. TOF technology stands out from these technologies however, they offer limited resolution opposed to the massive visual resolution offered by modern camera sensors. Therefore, the alignment of depth information with their visual counterpart usually results in more error prone low resolution scenario.

In 2009, ZCam, a company which developed TOF technology to develop a camera gesture interface to use human gestures to engage with gaming activities was taken over by Microsoft. It is rumoured that ZCam technology has enabled the Microsoft to develop a more advanced Kinect to use TOF camera technology at an extremely low cost compared to what has been commercially available from vendors such as Mesa Imaging. Kinect II released in the second half of 2013 is shown in Fig. 3.35. This is a very positive move for gaming enthusiasts as well as researchers in computer vision as Microsoft has a tendency to develop technology for mass market at reasonable costs. Its depth perception will increase from current 320 × 240 to 512 × 424 pixels which would be very valuable for emerging applications.

3.3.1 Image Registration

Image registration is the process of aligning two-dimensional images to a different three-dimensional space. In the case of a 3D camera, the registration process aligns the depth and colour streams together so that operations on either stream can then be related to the other stream. When the distance between the two perspectives of each camera (IR and Colour) is known, an approximation between pixels in each frame is determined. That is, if an object is closer to the camera (as known by the Depth component), the offset of that pixel to a pixel in the colour image array is larger than an object further away. There are numerous techniques for completing this operation, as listed below.

3.3.1.1 Edge/Key-Point Detection

A major option for image registration is the selection of key points, edges, surfaces or objects, then transferring those into another reference point. There are a number

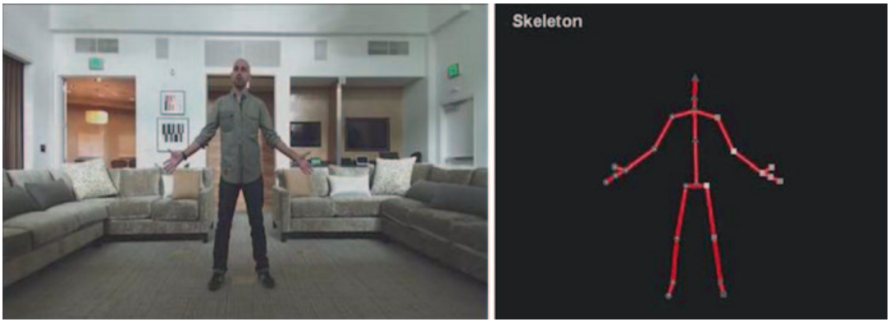


Fig. 3.36 Kinect markerless motion capture produced by BerkelTools

of methods for segmenting images into various objects which differ in complexity and accuracy [77]. Some methods prefer to isolate the different objects by locating edges or distinct rapid changes in the image [78]. Other methods search for continuous regions of consistent surfaces and segment within that section [79]. For greater accuracy, both methods can be combined for a hybrid-style algorithm. This method of key-point identification is found in ORB [80] and BRISK [81]. Another consideration of key-point locating involves searching for corners of objects in a scene, as these represent the orientation and boundary of an object, as covered by Rosten's work on image mapping [82]. All three options are available as part of the OpenCV computer vision library [83].

While often considered in the process of stitching together separate two-dimensional images to infer depth information of the scene, this method can also be used with a depth stream. The intention has been to improve the accuracy of the registration between an object in the colour stream and the comparative depth stream [84]. The key-point referencing method used was comparable to the results produced by the PrimeSense method coordinated by the camera. Some of the capabilities of Kinect II combined with BerkelTools offer new mode of gaming environment as shown in Fig. 3.36.

3.3.2 Stereo Triangulation for Depth Estimation—Passive Stereo Vision

Stereo vision is the concept of determining the lengths and sizes of objects in 3D space as done by humans and many other animals. It is a process which has been inspired by natural world where most of the fauna having two eyes. Mimicking such a system, engineers use two cameras which are few centimetres apart to create two slightly different views of the same scene (or object(s)). As shown in Fig. 3.37, in the ideal case of epipolar geometry which describes the mechanism of stereo vision, the *dot* in the diagram produces A_1 and A_2 in two camera views. However,

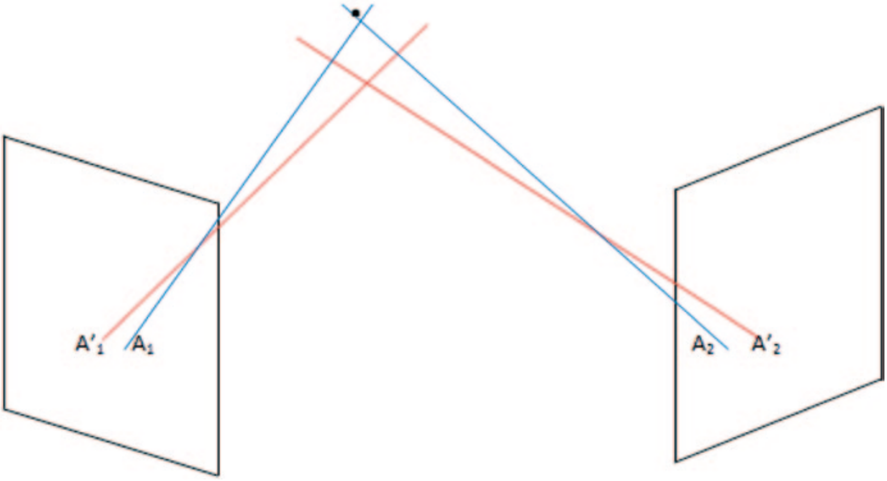


Fig. 3.37 Stereo vision disparity as seen by two cameras

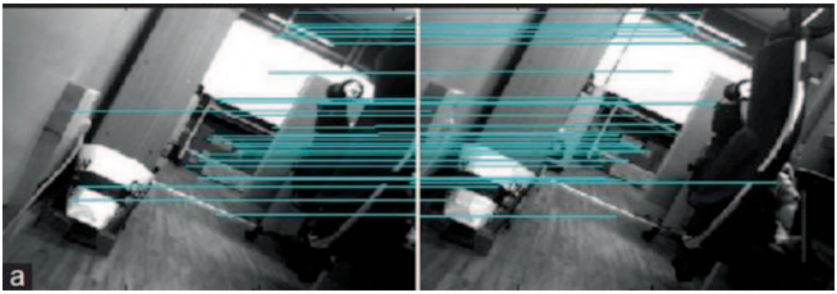


Fig. 3.38 Stereo disparity and correspondence. The *green* line connects the features identified in the *left* camera with the matching feature on the *right*

due to many imperfections in this system such as camera focussing issues, instead of the exact intersection of the two blue lines where the *dot* is, only measurements of A'$_1$, A'$_2$ are mostly available and used for triangulation. If the focal lengths and the camera and the distance between cameras are known (these are well-known for any stereo setup) the distance of the camera viewing plane to the *dot* (object) can be estimated using basic algebra. However, since ordinary images have complex scenery opposed to well-defined points, many other factors come into play reducing the applicability of simple stereovision in many imaging application. Figure 3.38 shows an image pair used in stereovision based robotic navigation attempt. These two images, even though are almost identical have been captured by a stereo camera pair with a slight change in point of view. Unlike Fig. 3.37 (triangulation diagram), the scenery has many points of interest. This issue now leads to identifying correspondence between each point, seen on the left camera with that of the right camera,

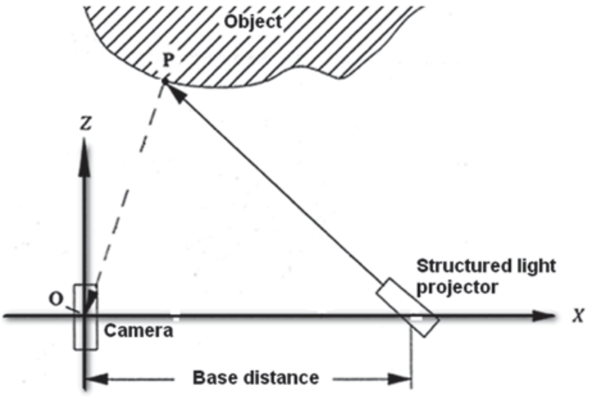


Fig. 3.39 Depth measurement using structured lighting with a single camera and a projector

known as the correspondence problem. The most prominent approach to find the correspondence relies on correlation between identical rows and constraint known as disparity that regulates mismatches [85–88].

Stereo vision often fails when used in featureless or textureless surroundings such as snow or highly repetitive patterns or uniform surroundings. Furthermore, if the forward looking cameras would not find any nearby imaging surfaces, it would also fail. Due to these limitations, stereovision along is not used for navigation especially in outdoor surroundings. However, stereovision does offer viable solutions for computer human interaction which usually takes place in indoors. Yet, the amount of processing power needed to solve the correspondence problem has dissuaded the commercial applications as seen by the investments of Microsoft on alternative technology that would be discussed next.

3.3.3 Active Stereo Vision-Coded Structured Light

Active stereo vision refers to the set up where instead of two cameras are setup to acquire images, a light pattern projector replaces one camera as shown in Fig. 3.39. In structured light imaging, a predefined light pattern is projected onto an object and simultaneously observed by a camera. The appearance of the light pattern in a certain region of the camera image varies with the camera-object distance. This effect is utilized to generate a distance image of the acquired scene. The predefined light patterns can be generated using many approaches as would be discussed in the next section. Some setups may use two cameras or multiple cameras to reduce the likelihood of occlusion by the object being imaged. Since this light pattern is visible to the human eye, such stereo systems are objectionable when used in public places such as airports for 3D face recognition or other types of surveillance [76].

In coded structured light, a light pattern is coded so that correspondences between image points and points of the projected pattern can be easily found. Therefore, coded structured light is considered to one of the most reliable ways for re-

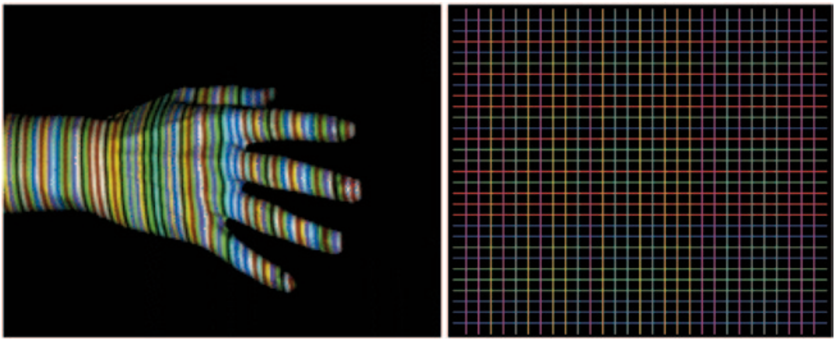


Fig. 3.40 125 slits encoded with a De Bruijn sequence of 8 colors and window size of 3 slits (*left*), courtesy of [99]. De Bruijn series spatial codification pattern (shown on *right*) [100]

covering the surface of objects [89]. Once the code pattern on the object is captured and decoded, the decoded points can be triangulated for 3D coordinates to recover the structure.

There are three prominent approaches for light pattern coding in practice. Time-multiplexing which is commonly used, is based on temporal coding. In this approach, a set of patterns are successively projected on the surface being imaged and captured at the same time. The codeword for any pixel is the result of multiplexing of the sequence of projected patterns on to that pixel. The codeword generation can be realized using, binary codes, *n-ary* codes, gray code combined with phase shifting and using hybrid techniques which are combination of time-multiplexing and neighbourhood strategies [90–93]. Time-multiplexing results in dense 3D points and high accuracy compared to other approaches. It is also suitable for objects with color as binary or *n-ary* codes are resilient against color objects. However, due to its reliance on multiple projections, the approach is limited to static objects.

The second techniques for light pattern coding are the approach based on spatial codification. The techniques used in this category generate a unique global pattern. The codeword for a single pixel can be determined by observing the pattern for its neighbors [94]. However, occlusions or non-neutral colors can lead to errors as not all neighborhood pixel patterns can be reliably retrieved. Some of these spatial neighbourhood strategies include De Bruiin sequences, *M-arrays* and strategies based on non-formal codification [95–98]. The technique is applicable to dynamic objects. Figure 3.40 shows a De Bruijn series coding pattern and how it is used for 3D depth measurements.

The third technique in light pattern generation is *direct codification*. In this approach, each pixel gets its own color (color intensity) to represent the pattern [101–106]. However, the observed color from any pixel does not solely depend on the projected color. It also depends on the color of the surface. Since different color objects reacts to colors differently, this strategy is only suitable for objects with neutral color object or objects with pale colors [94]. Some implementations

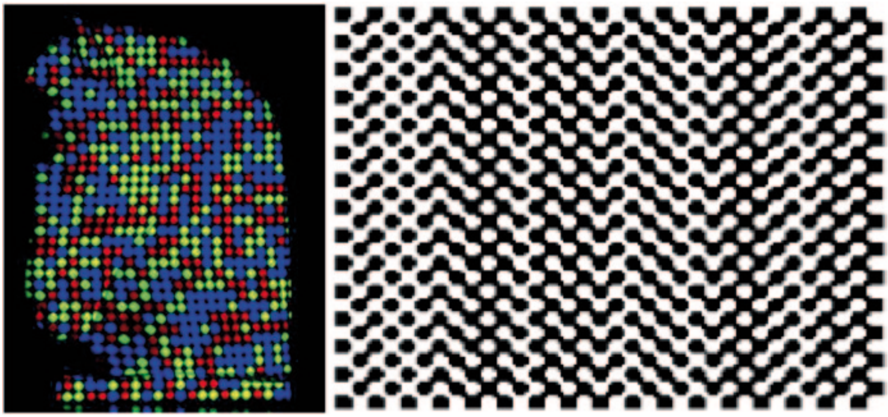


Fig. 3.41 M-arry represented with an array of *coloured dots* (*left*), courtesy of Morano et al. [107]. M-array proposed by Vuylsteke et al. represented with shape primitives [100]

of this direct codification relies on capturing many reference frames with change of colors. Therefore this is not generally used for dynamic scenes.

Even though objectionable to the human user, structured light based stereo vision can be very effective for 3D scanning in indoors. Many researchers have used different types of grid patterns and coded color schemes to determine the depth of the objects using the observed distortion due to object depth and shape of the predefined pattern. Such color schemes and code patterns as shown in Fig. 3.41 can help in reducing the correspondence problem if more than two cameras are used. However, since the light diffracts much more than a laser beam, the resolution from visible light based stereo vision is limited.

3.3.4 Infrared Structured Light for Active Stereo Vision

Recently, researchers have developed invisible infrared lighting patterns or structured light to obtain depth information without any public backlash. This has resulted in the successful Microsoft Kinect using infrared structured light patterns to develop low cost, yet, effective gaming apparatus. The light pattern used in Kinect is known as a speckle pattern which resembles speckle noise. However, it is a well structured pattern, projected using an infrared laser through a plastic material which houses the pattern. The distortion of this pattern is compared with the original speckle pattern to determine the depth of the objects. One of the problems of this approach which is also common to the visible structured light is the shadows created by occlusion of objects. In addition to the shadows, the depth-images contain noise due to limited resolution of the IR camera. Since the vision camera is of much higher resolution, a single infrared point of the captured pattern may not be

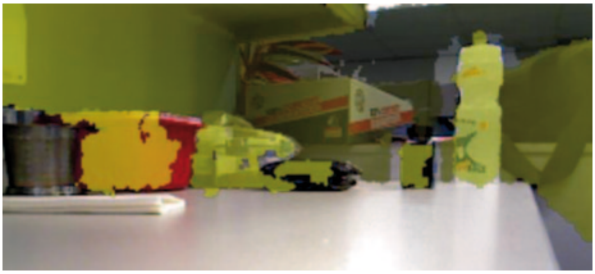


Fig. 3.42 Color Camera view with depth information fused using Kinect

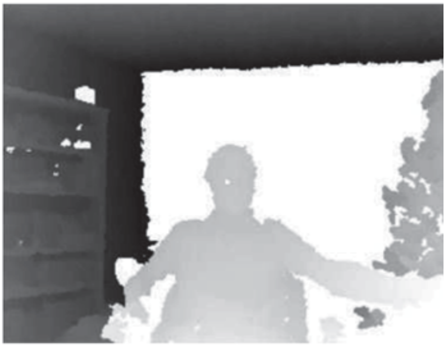


Fig. 3.43 Depth information is easy to ascertain in a gray-scale image fused with depth information using kinect

assigned to a single pixel. Therefore the position must be interpolated by the portion of luminosity of the two adjacent pixels. This interpolation is sensitive to external infrared radiation such as direct sun light. For many devices such as Kinect, increasing distance from the camera leads to poor resolution and errors due to misalignment of infrared pixels with CMOS pixels. There are other errors due to offset of the CMOS and the Infrared receiver. There are methods for calibrating the camera to modify the image, such that the depth map from the infrared camera and CMOS input are aligned [108–110]. The scene reconstruction then follows a process used in 3D game design known as texturing. The model generated from the depth map data is "painted" with a texture, in this case, the video input from the CMOS camera as shown in Figs. 3.42 and 3.43. This can be completed in real-time through the use of modern graphics processors.

Kinect uses an infrared speckle as shown in Fig. 3.44. The known pattern is compared with its offset when the pattern falls on a surface and distorted. This disparity measure results in triangulation to reveal the depth map. In Kincet, the color visual image sensor has much higher pixel density than its infrared sensor. This results in false depth map that is not really an issue in Kinects applications of gaming. However, since it is difficult to improve the resolution of the depth map using infrared speckle patterns, Microsoft has resorted to TOF camera technology for their upcoming Kinect II as was discussed before.

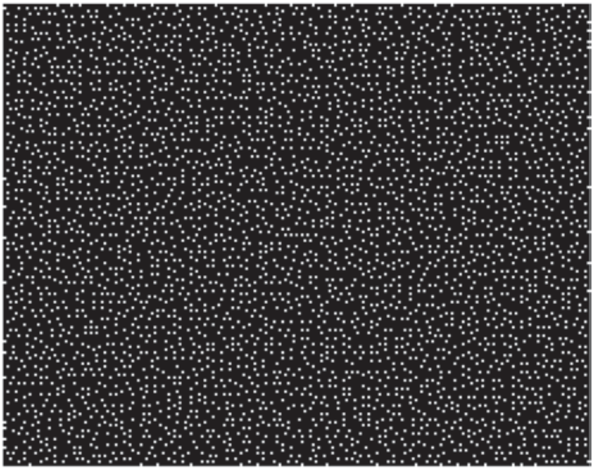


Fig. 3.44 Speckle pattern used in Kinect. (Courtesy of [111])

3.3.5 Time of Flight (TOF) Camera for Depth Information

The advantage of TOF compared to triangulation methods in passive or active stereo vision is that the whole system is very compact where the illumination (pattern projector) is placed just next to the camera lens, whereas the other systems need a certain minimum base line. In contrast to laser scanning systems, no moving parts are present in the system.

The working principle of the TOF camera can be understood as using extremely short or narrow light pulses to illuminate a target (any object in its flying path) and record the return of the pulse on every pixel on the camera sensor with its time of arrival. This is done extremely precisely so that even 1 cm depth differences of any part of the object being scanned can be differentiated. The pulse width of the illumination determines the maximum range the camera can handle. In case of pulse width of 50 ns, the scanning range is restricted to 7.5 m. These short times show that the illumination unit is a critical part of the system. These short pulses can currently be generated with special LEDs or lasers.

When infrared structured light is used in the presence of background light, the CMOS camera sensor receives an additional part of the signal. This results in disturbing the distance measurement. In order to eliminate the background contribution of the signal, the whole measurement can be performed a second time with the infrared illumination switched off. If the objects moves and are further away than the distance range, the measurements result in error. Here, a second measurement with the control signals delayed by an additional pulse width helps to suppress such objects. Other systems work with a sinusoidally modulated light source instead of the pulse source.

As mentioned previously in this chapter, the ZCam's time-of-flight camera system features a near-infrared (NIR) pulse illumination component as well as an image sensor with a fast gating mechanism. Based on the known speed of light, ZCam coordinates the timing of NIR pulse wave emissions from the illuminator with the gating of the image sensor so that the signal reflected from within a desired depth range is captured exclusively. The amount of pulse signal collected for each pixel corresponds to where within the depth range the pulse was reflected from, and can thus be used to calculate the distance to a corresponding point on the captured subject [112, 113].

Due to the fast timing required for light-based time-of-flight, the ZCam uses custom hardware for illumination and gating. The illuminator is a series of NIR laser diodes around the lens barrel, switched by special high-speed driver circuits that produce pulses with a rise time and fall time of less than 1 ns [113]. The time-of-flight camera is optically matched with a corresponding video camera, allowing the RGB video and range imaging to integrate together.

This chapter methodically developed the required knowledge for preprocessing that is vital in understanding object shapes. When undesirable noise and artefacts are present, morphological filtering based processing can restore objects so that they can be understood by computer vision. With the use of depth information, cluttered backgrounds can be removed to reveal the foreground which typically contains the hand gestures for human computer intearaction.

References

1. Abdel-Mottaleb, M., Elgammal, A.: Face detection in complex environments from color lmages. Proceedings of the International Conference on Image Processing (ICIP), 622–626 (1999)
2. Alshebani, Q., Premaratne, P., Vial, P.: An Embedded Door Access Based on Face Recognition System: A Survey. To appear in (ICSPCS), 2013, Australia, (2013)
3. Ahmed, E., Crystal, M., Dunxu H.: Skin Detection-a short Tutorial. Encyclopedia of Biometrics by Springer-Verlag Berlin, Heidelberg, 1218–1224 (2009)
4. Forsyth, D.A., Fleck, M.M.: Identifying nude pictures. Proceeding of Third IEEE Workshop on Applications of Computer Vision, 103–108 (1996)
5. Albiol, A., Torres, L., Delp, E.: Optimum color spaces for skin detection. In: Proceedings of the International Conference on Image Processing (ICIP), 122–124 (2001)
6. Shin, M.C., Chang, K.I., Tsap, L.V.: Does colorspace transformation make any difference on skin detection? WACV '02: Proceedings of the Sixth IEEE Workshop on Applications of Computer Vision, 275 (2002)
7. Zheng, Q.F., Zhang, M.J., Wang, W.Q.: A hybrid approach to detect adult web images. PCM 2 3332, 609–616 (2004)
8. Lee, Y., Yoo, S.I.: An elliptical boundary model for skin color detection. In: Proceedings of the International Conference on Imaging Science, Systems, and Technology, (2002)
9. Senior, A., Hsu, R.L., Mottaleb, M.A., Jain, A.K.: Face detection in color images. IEEE Trans. PAMI 24(5), 696–706 (2002)
10. Menser, B., Wien, M.: Segmentation and tracking of facial regions in color image sequences. Proceeding of SPIE Visual Communications and Image Processing, 731–740 (2000)
11. Jones, M.J., Rehg, J.M.: Statistical color models with application to skin detection. In: Proceeding of CVPR'99 1, 274–280 (1999)
12. Beetz, M., Radig, B., Wimmer, M.: A person and context specific approach for skin color classification. 18th International Conference on Pattern Recognition (ICPR 2006), (2006)
13. Soriano, M., et al.: Skin detection in video under changing illumination conditions. 15th International Conference on Pattern Recognition, (2000)

14. Kawato, S., Ohya, J.: Automatic skin-color distribution extraction for face detection and tracking. 5th International Conference on Signal Processing Proceedings (WCCC-ICSP 2000), (2000)
15. Park, J., et al.: Detection of human faces using skin color and eyes, IEEE International Conference on Multimedia and Expo (ICME 2000), (2000)
16. Kovac, J., Peer, P., Solina, F.: 2D versus 3D color space face detection. 4th EURASIP Conference on Video/Image Processing and Multimedia Communications, 449–454 (2003)
17. Gomez, G., Morales, E.F.: Automatic feature construction and a simple rule induction algorithm for skin detection. Proceedings of ICML workshop on Machine Learning in Computer Vision, 31–38 (2002)
18. Gasparini, F., Schettini, R.: Skin Segmentation using Multiple Thresholding. Proceedings of SPIE 6061, 128–135 (2006)
19. Vezhnevets, V., Sazonov, V., Andreeva, A.: A Survey on Pixel-Based Skin Color Detection Techniques, In Proceedings of GRAPHICON-2003, (2003)
20. Zarit, B.D., Super, B.J., Quek, F.K.H.: Comparison of five color models in skin pixel classification. ICCV'99 Int'l Workshop on recognition, analysis and tracking of faces and gestures in Real-Time systems, 58–63 (1999)
21. Hsu, R.-L., Abdel-Motalleb, M., Jain, A. K.: Face detection in color images. IEEE Trans. PAMI **24**(5), 696–706 (2002)
22. Ahlberg, J.: A system for face localization and facial feature extraction. Technical Report no. LiTH-ISY-R-2172, Linkoping University, (1999)
23. Sebastian, P., Yap, V.V., Comley, R.: The effect of colour space on tracking robustness. 3rd IEEE Conference on Industrial Electronics and Applications (ICIEA 2008), 2512–2516 (2008)
24. Tsekeridou, S., Pitas, I.: Facial feature extraction in frontal views using biometric analogies. Proceedings of IX European Signal Processing Conference 1, 315–318 (1998)
25. Garcia, C., Tziritas, G.: Face detection using quantized skin color regions merging and wavelet packet analysis. IEEE Transaction on Multimedia. **1**, 264–277 (1999)
26. Poynton, C.A..: Frequently Asked Questions About Colour. In ftp://www.inforamp.net/pub/users/poynton/doc/colour/ColorFAQ.ps.gz (1995)
27. Skarbek, W., Koschan, A.: Colour image segmentation—a survey. Technical Report, Institute for Technical Informatics, Technical University of Berlin, (1994)
28. Sigal, L., Sclaroff, S., Athitsos, V.: Estimation and prediction of evolving color distributions for skin segmentation under varying illumination. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition 2, 152–159 (2000)
29. Mckenna, S., Gong, S., Raja, Y.: Modelling facial colour and identity with gaussian mixtures. Pattern Recognit 31, **12**, 1883–1892 (1998)
30. Jordao, L., Perrone, M., Costeira, J., Santos-Victor, J.: Active face and feature tracking. In Proceedings of the 10th International Conference on Image Analysis and Processing, 572–577 (1999)
31. Fleck, M., Forsyth, D.A., Bregler, C.: Finding nacked people. In Proceedings of the ECCV 2, 592–602 (2002)
32. Brown, D., Craw, I., Lewthwaite, J.: A som based approach to skin detection with application in real time systems. In Proceedings of the British Machine Vision Conference, (2001)
33. http://en.wikipedia.org/wiki/HSL_and_HSV
34. Terrillon, J.-C., Shirazi, M.N., Fukamachi, H., Akamatsu, S.: Comparative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in color images. In Proceedings of the International Conference on Face and Gesture Recognition, 54–61 (2000)
35. Poynton, C., Funt, B.: Perceptual uniformity in digital image Representation and display. Color Research and Applications, (2013)
36. Kaur, A., Kranthi, B.V.: Comparison between YCbCr color space and CIELab color space for skin color segmentation. Int. J. Appl. Info. Syst. **3**(4), 30–33 (2012)
37. Singh, S.K., Chauhan, D.S., Mayank, V., Singh, R.: A robust skin color based face detection algorithm. Tamkang J. Sci. Engg. **6**(4), 227–234 (2003)

38. Khan, R., Khan, Z., Aamir, M., Sattar, S.Q.: Static filtered skin detection. IJCSI International Journal of Computer Science Issues. **9**(2), 257–261 (2012)
39. Poudel, R.P.K., Nait-Charif, H., Zhang, J.J., Liu, D.: Region-based skin color detection. VISAPP 1, 301–306 (2012)
40. Hikal, N.H., Kountchev, R.: Skin color segmentation using adaptive PCA and modified elliptic boundary model. ICACSIS. **2011**, 407–412 (2011)
41. Chen, Q., Wu, H., Yachida, M.: Face detection by fuzzy pattern matching. In Proceedings of the Fifth International Conference on Computer Vision, 591–597 (1995)
42. Schumeyer, R., Barner, K.: A color-based classifier for region identification in video. Vis. Commun. Image Process. SPIE. **3309**, 189–200 (1998)
43. Birchfield, S.: Elliptical head tracking using intensity gradients and color histograms. In Proceedings of CVPR '98, 232–237 (1998)
44. Yang, M.H., Ahuja, N.: Detecting human faces in color images. In International Conference on Image Processing 1, 127–130 (1998)
45. Kruppa, H., Bauer, M., Schiele, B.: Skin patch detection in real-world images. In: Van Gool, L. (ed.), Pattern Recognition, Lecture Notes in Computer Science **2449**, 109–116 (2002)
46. Chang, F., Ma, Z., Tian, W.: A region-based skin color detection algorithm advances in knowledge discovery and data mining. Lecture Notes in Computer Science 4426, 417–424 (2007)
47. Ren, X., Malik, J.: Learning a classification model for segmentation. In IEEE International Conference on Computer Vision 1, 10–17 (2003)
48. Moore, A.P., Prince, S., Warrell, J., Mohammed, U., Jones, G.: Superpixel lattices. In IEEE Conference on Computer Vision and Pattern Recognition, 1–8 (2008)
49. Soatto, S.: Actionable information in vision. In Proceedings of the International Conference on Computer Vision 25, 17–48 (2009)
50. Fulkerson, B., Vedaldi, A., Soatto, S.: Class segmentation and object localization with superpixel neighborhoods. In Proceedings of International Conference on Computer Vision 5, 670–677 (2009)
51. Brand, J., Mason, J.: A comparative assessment of three approaches to pixellevel human skin-detection. In Proceedings of the International Conference on Pattern Recognition 1, 1056–1059 (2000)
52. Soille, P.: Morphological Image Analysis Principles and Applications, 2nd ed., XVI, 391 (2003)
53. www.cs.princeton.edu/~pshilane/class/mosaic/
54. Smith, S.W.: The Scientist and Engineer's Guide to Digital Signal Processing, Chap. 25.
55. www.mmorph.com/html/morph/mmopen.html/
56. Gonzalez, R., Woods, R.: Digital Image Processing, Addison-Wesley Publishing Company, 518–548 (1992)
57. Davies, E.: Machine Vision: Theory, Algorithms and Practicalities, Academic Press, 149–161 (1990)
58. Haralick, R., Shapiro, L.: Computer and Robot Vision 1, Addison-Wesley Publishing Company, Chap. 5, 168–173 (1992)
59. Jain, A.: Fundamentals of Digital Image Processing, Prentice-Hall, Chap. 9. (1989)
60. Vernon, D.: Machine Vision, Prentice-Hall, Chap. 4 (1991)
61. Ionescu, B., Coquin, D.: Dynamic hand gesture recognition using the skeleton of the hand. EURASIP J. Appl. Signal Process. **13**, 2101–2109 (2005)
62. Coquin, D., Bolon, P.: Applications of Baddeley's distance to dissimilarity measurement between gray scale images. Pattern Recognit. Lett. **22**(14), 1483–1502 (2001)
63. Reddy, K.S., Latha, P.S., Babu, M.R.: Hand Gesture Recognition Using Skeleton of Hand and Distance Based Metric, D.C. Wyld et al. (eds.) ACITY 2011, CCIS, 198, 346–354 (2011)
64. Borgefors, G.: Distance transformations in digital images. Comp. Vis. Graphics Image Process. **34**(3), 344–371 (1986)
65. Chehadeh, Y., Coquin, D., Bolon, H.: A skeletonization algorithm using chamfer distance transformation adapted to rectangular grids. In: Proceedings of 13th IEEE International Conference on Pattern Recognition (ICPR 1996) 2, 131–135 (1996)

66. Hasthorpe, J., Mount, N.: The generation of river channel skeletons from binary images using raster thinning algorithms. School of Geography, University of Nottingham
67. Wu, S., Jiang, F., Zhao, D.: Hand Gesture Recognition based on Skeleton of Point Clouds. 2012 IEEE fifth International Conference on Advanced Computational Intelligence (ICACI), 566–569 (2012)
68. Premaratne, P., Ajaz, S., Premaratne, M.: Hand Gesture Tracking and Recognition System Using Lucas-Kanade Algorithm for Control of Consumer Electronics. Neurocomputing Journal, (2012)
69. Premaratne, P., Nguyen, Q.: Consumer electronics control system based on hand gesture moment invariants. IET Comp. Vis. **1**(1), 35–41 (2007)
70. Zou, Z., Premaratne, P., Premaratne, M., Monaragala, R., Bandara, N.: Dynamic hand gesture recognition system using moment invariants. 5th International Conference on Information and Automation for Sustainability, 108–113 (2010)
71. Herath, D.C., Kroos, C., Stevens, C.J., Cavedon, L., Premaratne, P.: Thinking head: Towards human centred robotics. 11th International Conference on Control, Automation, Robotics and Vision (ICARCV), 2042–2047 (2010)
72. Premaratne, P., Ajaz, S., Premaratne, M.: Hand Gesture Tracking and Recognition System for Control of Consumer Electronics. Springer Lecture Notes in Artificial Intelligence (LNAI) 6839, 588–593 (2011)
73. Premaratne, P., Nguyen, Q., Premaratne, M.: Human computer interaction using hand gestures. Adv. Intell. Comput. Theor. Appl. Commun. Comput. Info. Sci. **93**, 381–386 (2010)
74. Premaratne, P., Safaei, F., Nguyen, Q.: Moment invariant based control system using hand gestures Intelligent Computing in Signal Processing and Pattern recognition, Book Series Lecture Notes in Control and Information Sciences vol. 345, 322–333 (2006)
75. Premaratne, P., Safaei, F.: Feature based Stereo Correspondence using Moment Invariant. Proceedings of the IEEE International Conference on Information and Automation for Sustainability, 104–108 (2008)
76. McGuire, D., Premaratne, P.: A System for the 3D Reconstruction of the Human Face using the Structured Light Approach. The 5th Workshop on the Internet Telecommunications and Signal Processing, 1–7 (2006)
77. Ding, Y., Ping, X., Hu, M., Wang, D.: Range image segmentation using randomized Hough transform. In Information, Communications and Signal Processing, 2003 and Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint Conference of the Fourth International Conference on 2, 807–811 (2003)
78. Jiang, X., Bunke, H.: Edge Detection in Range Images Based on Scan Line Approximation. Comp. Vis. Image Underst. **73**(2), 183–199 (1999)
79. Besl, P.J., Jain, R.C.: Segmentation through Variable-Order Surface Fitting. IEEE Trans. Pattern Anal. Mach. Intell. **10**(2), 167–192 (1988)
80. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: Orb: An efficient alternative to sift or surf. International Conference on Computer Vision, (2011)
81. Leutenegger, S., Chli, M., Siegwart, R.: Brisk: Binary robust invariant scalable keypoints. In Dimitris N. Metaxas, Long Quan, Alberto Sanfeliu, Luc J. Van Gool (eds.) ICCV, 2548–2555 (2011)
82. Rosten, E., Drummond, T.: Machine learning for high-speed corner detection. In European Conference on Computer Vision 1, doi: 10.1007/11744023 34. http://edwardrosten.com/work/rosten_2006_machine.pdf., 430–443 (2006)
83. OpenCV, W.G.: Opencv 2.4.5.0 documentation. (2013)
84. Herrera, D.C., Kannala, J., Heikkil, J.: Joint depth and color camera calibration with distortion correction. IEEE Trans. Pattern Anal. Mach. Intell. **34**(10), 2058–2064 (2012)
85. Howard, I., Rogers, B.: Seeing in depth. (2002)
86. Coutant, B.E., Westheimer, J.: Population distribution of stereoscopic ability. Ophthal. Physiol. Optics. **13**(1), 3–7 (1993)
87. Liesbeth, I.N., Mazyn, Lenoir, M., Montagne, G., Geert, J., Savelsbergh, P.: The contribution of stereo vision to one-handed catching. Exp. Brain Res. **157**(3), 383–390 (2004)
88. Salas, J., Tomasi, C.: People detection using color and depth images. Pattern Recognition, Lecture Notes in Computer Science 6718, 27–135 (2011)

89. Payeur, P., Desjardins, D.: Structured light stereoscopic imaging with dynamic pseudo-random patterns. Image Analysis and Recognition. Lect. Notes Comput. Sci. 5627, 687–696 (2009)
90. Desjardins, D., Payeur, P.: Dense stereo range sensing with marching pseudo-random patterns. Fourth Canadian Conference on Computer and Robot Vision (CRV '07), 216–226 (2007)
91. Grin, P.M., Narasimhan, L.S., Yee, S.R.: Generation of uniquely encoded light patterns for range data acquisition. Pattern Recog. 25(6), 609–616 (1992)
92. Morita, H., Yajima, K., Sakata, S.: Reconstruction of surfaces of 3D objects by M-array pattern projection method. Second International Conference on Computer Vision, 468–473 (1998)
93. Salvi, J., Pagès, J., Batlle, J.: Pattern codification strategies in structured light systems. Pattern Recognit. 37(4), 827–849 (2004)
94. van Aardenne-Ehrenfest, T., de Bruijn, N.G.: Circuits and trees in oriented linear graphs. Simon Stevin. 28, 203–217 (1951)
95. Han, Y.K., Yang, K.: New M-ary power residue sequence families with low correlation. Proceedings of IEEE International Symposium on Information Theory (ISIT2007), 2616–2620 (2007)
96. Han, Y.K., Yang, K.: New M-ary sequence families with low correlation and large size. IEEE Trans. Inf. Theory 55(4), 1815–1823 (2009)
97. Kim, Y.-S., Chung, J.-S., No, J.-S.: and Chung, H.: New families of M-ary sequences with low correlation constructed from Sidel'nikov sequences. IEEE Trans. Inf. Theory 54(8), 3768–3774 (2008)
98. Zhang, L., Cudess, B., Seitz, M.: Rapid Shape Acquisition Using Color Structured Lightand Multi-pass Dynamic Programming. 1st IEEE International Symposium on 3D Data Processing, Visualization, and Transmission, 1–13 (2002)
99. Vuylsteke, P., Oosterlinck, A.: Range image acquisition with a single binary-encoded light pattern. Pattern Analy. Mach. Intell. 12(2), 148–163 (1990)
100. Carrihill, B., Hummel, R.: Experiments with the intensity ratio depth sensor. Comp. Vis. Graphics Image Process. **32**, 337–358 (1985)
101. Hung, D.: 3d scene modelling by sinusoid encoded illumination. Image Visi. Comp. **11**, 251–256 (1993)
102. Tajima, J., Iwakawa, M.: 3-D data acquisition by rainbow range finder. International Conference on Pattern Recognition, 309–313 (1990)
103. Geng, Z.J.: Rainbow 3-dimensional camera new concept of high-speed 3-dimensional vision systems. Opt. Eng. **35**(2), 376–383 (1996)
104. Wust, C., Capson, D.W.: Surface profile measurement using color fringe projection Mach. Vis. Appl. **4**, 193–203 (1991)
105. Sato, T.: Multispectral pattern projection range finder. Proceedings of the Conference on Three-Dimensional Image Capture and Applications II 3640, SPIE, 28–37 (1999)
106. Morano, R.A., Ozturk, C., Conn, C., Dubin, S., Zietz, S., Nissanov, J.: Structured light using pseudorandom codes. Pattern Anal. Mach. Intell. **20**(3), 322–327 (1998)
107. Sali, E., Avraham, A.: Three-Dimensional Mapping and Imaging. http=://www.faqs.org/patents/app/20100265316#ixzz299280m00 (2010). Accessed Oct 2010
108. Shpunt, A., Mor, Z.: Non-Uniform Spatial Resource Allocation for Depth Mapping. http=://www.faqs.org/patents/app/20110211044#ixzz299LnJhHM (2011). Accessed Sept 2011
109. Zalevsky, Z., Shpunt, A., Maizels, A., Garcia, J.: Method and System for Object Reconstruction. http://www.sumobrain.com/patents/WO2007043036.html (2007). Accessed April 2007
110. http://azttm.wordpress.com/2011/04/03/kinect-pattern-uncovered/
111. Katz, S.: Boxing with ZCam. Engineering TV. (2009)
112. Iddan, G.J., Yahav, G.: 3D imaging in the studio. Proceedings of SPIE 4298, (2003)
113. Iddan, G.J., Yahav, G.: 3D imaging in the studio.Three-Dimensional Image Capture and Applications IV, Brian D.C., Joseph H.N., Roy P.P. (eds.), Proceedings of SPIE 4298, 48–55 (2001)