Advances in Optics and Photonics

Fundamentals of automated human gesture recognition using 3D integral imaging: a tutorial

Bahram Javidi,[1,*] Filiberto Pla,[2] José M. Sotoca,[2] Xin Shen,[3] Pedro Latorre-Carmona,[4] Manuel Martínez-Corral,[5] Rubén Fernández-Beltrán,[2] and Gokul Krishnan[1]

[1]University of Connecticut, Electrical & Computer Engineering Department, 371 Fairfield Way, Storrs, Connecticut 06269, USA
[2]Institute of New Imaging Technologies, Universitat Jaume I. Campus Riu Sec s/n, 12071 Castelló de la Plana, Spain
[3]Massachusetts College of Liberal Arts, Computer Science Department, 375 Church Street, North Adams, Massachusetts 01247, USA
[4]University of Burgos, Department of Computer Science, Avda. Cantabria s/n. 09006 Burgos, Spain
[5]3D Imaging and Display Laboratory, University of Valencia, Department of Optics, 46100 Burjassot, Spain
*Corresponding author: Bahram.Javidi@uconn.edu

Received February 18, 2020; revised October 20, 2020; accepted October 20, 2020; published December 14, 2020 (Doc. ID 390929)

Automated human gesture recognition is receiving significant research interest, with applications ranging from novel acquisition techniques to algorithms, data processing, and classification methodologies. This tutorial presents an overview of the fundamental components and basics of the current 3D optical image acquisition technologies for gesture recognition, including the most promising algorithms. Experimental results illustrate some examples of 3D integral imaging, which are compared to conventional 2D optical imaging. Examples of classifying human gestures under normal and degraded conditions, such as low illumination and the presence of partial occlusions, are provided. This tutorial is aimed at an audience who may or may not be familiar with gesture recognition approaches, current 3D optical image acquisition techniques, and classification algorithms and methodologies applied to human gesture recognition.

https://doi.org/10.1364/AOP.390929

Fundamentals of automated human gesture recognition using 3D integral imaging: a tutorial

Bahram Javidi, Filiberto Pla, José M. Sotoca, Xin Shen, Pedro Latorre-Carmona, Manuel Martínez-Corral, Rubén Fernández-Beltrán, and Gokul Krishnan

1. INTRODUCTION

The ability to identify human activities is one of the most studied areas in computer vision and machine learning today. Human activity identification aims to detect and analyze human activities using information acquired by sensors in an automated manner. These sensors can be RGB cameras, range sensors, or other sensing modalities.

Research on human gesture recognition has a direct influence on various research fields, e.g., sign language recognition [1]. The lack of widespread sign language knowledge is a global issue. As a result, there is a large demand to realize efficient sign language recognition systems, with computers set to play a vital role in this regard. In the field of tele-robotics [2], robots that can imitate movements and gestures made by humans are in demand. Moreover, there is an urgent need to be able to operate such robots remotely through gestures via environments such as games, simulations, and virtual reality applications [3]. Human action and gesture recognition are important when *actors* (game characters) need to move in a way that seems real and natural to the player. Human–computer interaction (HCI) [4] has applications in fields including military, medicine, graphical and data processing, and document annotation and editing.

Three-dimensional (3D) optical image acquisition and processing is a promising technique for acquiring 3D information from a scene. It has shown good performance when object(s) of interest are occluded by obstacles or are under low or even photon-starved illumination conditions.

As with any expanding research field, tutorials are valuable educational tools for improving the understanding of the fundamental components and basics of the field. This paper presents a tutorial overview of the current 3D optical image acquisition technologies for gesture recognition, including the most promising human gesture recognition algorithms. In addition, we present examples illustrating the performance of gesture recognition systems in degraded environments.

Taking the aforementioned diverse applications into account, this tutorial paper discusses human gesture recognition using 3D optical imaging, structured around the following three pillars: (1) the main characteristics of 3D image acquisition systems based on camera arrays, such as integral imaging, with a particular focus on their properties and advantages when compared to other imaging systems within the framework of human gesture recognition; examples might include inferring the depth of a 3D scene or extracting information under degraded environmental conditions such as very low illumination or when the objects to be viewed are obscured; (2) current methodologies that use 3D information for human gesture/action classification; and (3) a series of real gesture recognition experiments and examples where the

advantages of 3D optical imaging systems are illustrated in terms of human gesture recognition performance metrics. We consider this structure useful to meet the main objectives of this tutorial: (a) presenting a wide variety of sensing, acquisition, and processing techniques that are currently being employed in gesture recognition; (b) understanding the main components of automated gesture recognition using optical imaging; and (c) proposing an optimal acquisition methodology for human gesture recognition that works not only under "normal conditions" but also under sub-optimal or degraded conditions.

The study is supported by a wealth of citations to assist readers and provide additional details regarding the topics covered in this tutorial [1–114]. We may have overlooked some relevant works, as it is not possible to present an exhaustive list of related studies in a single tutorial paper, for which we apologize in advance.

The rest of this paper is organized as follows. Section 2 provides a summary of the current data acquisition technologies and recognition methodologies. Section 3 discusses the main characteristics and properties of the different types of 3D optical technologies and configurations. In addition, details regarding the advantages of 3D optical imaging under low illumination and degraded conditions, e.g., in the presence of occlusions, and low illumination environments are provided. Section 4 presents an overview of human gesture recognition methodologies and algorithms from a pattern

Table 1. List of Acronyms and Initials Used in this Tutorial

Acronym	Definition
AUC	Area under the curve
BoW	Bag of words
CCD	Charge-coupled device
CMOS	Complementary metal-oxide semiconductor
CNNs	Convolutional neural networks
DBF	Depth-based filtering
DOF	Depth of field
EMG	Electromyography
FN	False negative
FOV	Field of view
FP	False positive
FT	Fourier transform
HCI	Human–computer interaction
HOF	Histogram of optical flow
HOG	Histogram of oriented gradients
GHOG	Global histogram of oriented gradients
IMUs	Inertial measurement units
LMCs	Leap motion controllers
LSTM	Long short-term memory
LWIR	Long-wave infrared
MCC	Matthews's correlation coefficient
ML	Maximum-likelihood
MLA	Microlens array
PCA	Principal component analysis
PMLEM	Penalized maximum-likelihood expectation maximization
POE	Peak-to-output energy
RBF	Radial basis function
ReLU	Rectified linear unit
RGB-D	Red green blue-depth
RNN	Recurrent neural network
ROC	Receiver operating characteristic
ROI	Region of interest
SL	Structured light
STIPs	Spatio-temporal interest points
SVM	Support vector machine
TN	True negative
ToF	Time-of-flight
TP	True positive
TV	Total variation

recognition perspective, including the main performance metrics used to assess the capability of the gesture recognition systems. Section 5 presents real experiments conducted on human gesture recognition under normal conditions, as well as under occlusions and degraded environmental conditions. Section 6 concludes the paper. The acronyms and initials used in this tutorial are listed in Table 1.

2. FUNDAMENTALS OF HUMAN GESTURE RECOGNITION

Gestures are an intrinsic part of most of our daily actions and activities and are a crucial component of human communication, either helping with speech cognition, in some cases for people with impaired hearing function, or replacing spoken language in circumstances in which the communication conditions are degraded (underwater, noisy environments, secret communications, etc.). Gestures should not be confused with other hand movements. In general, gestures start slowly, then have a quicker part, and finish by returning to a resting position. These features could be used to help define a classification strategy.

The hand is the body part involved most frequently in defining a gesture. Consequently, the hand is the body part best adapted for communication and, therefore, best suited for integration with HCI. According to [5,6], approximately 21% of all gestures involve just the hands, 7% the hand and the head, and 7% the body.

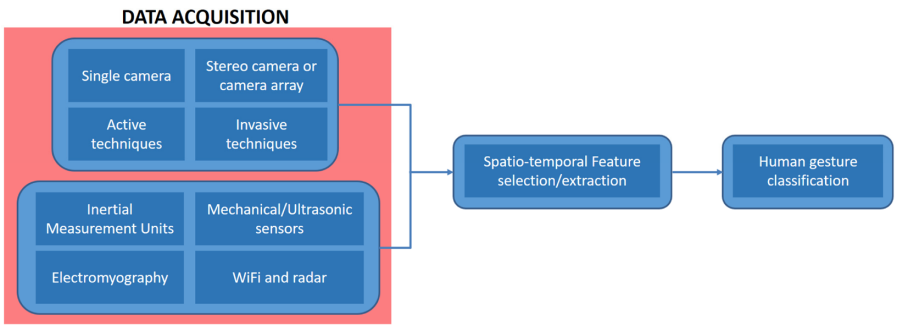
Applications of human gesture recognition are numerous and include HCI, human–machine interaction, virtual reality, communications, entertainment, security, and autonomous driving, to name just a few. It is an important component of technologies aimed at assisting the handicapped and the elderly, helping them to deliver a significant impact.

Human gesture recognition involves the following processing steps: (1) data acquisition, (2) feature characterization, including extraction and/or reduction, and (3) classification based on these features (Fig. 1).

Chronologically, the first effort towards the acquisition of data useful for gesture recognition came from the use of gloves that incorporated sensors. These first devices were both uncomfortable and considerably expensive. Thereafter, cameras started to be used as a cheap and easy to use (and manage) data acquisition alternative. Today, there are a myriad of sensors that can be used for gesture recognition. We can characterize gestures (from an acquisition viewpoint) using vision-based or sensor-based approaches.

Vision-based approaches are based on the acquisition of human gestures through a remote camera (or camera setup). Common image acquisition setups include:

Figure 1



Human gesture recognition workflow.

- Single camera. From a cost-efficiency perspective, and with the technological evolution of cameras, single cameras (a video camera, smartphone, or webcam) are a competitive solution, although they may be limited in terms of performance.
- Stereo-camera or a camera array. The use of two or more cameras is a promising solution for gesture recognition because these systems can capture features that are helpful for classification, particularly those related to 3D scene information, such as depth, body shape, location, and orientation [7,8].
- Active sensing techniques. These techniques are based on the projection of structured light on a scene [9]. Examples include RGB-D sensors such as Kinect and/or leap motion controllers (LMCs) [10].
- Techniques based on the use of body markers [11]. Information about the body or hand position in 3D space can be obtained using LED lights or elements such as colored gloves.

In the case of sensor-based approaches, there are many devices that can acquire information about the position, motion, and trajectory (including the speed vector) through the following [12]:

- Inertial measurement units (IMUs). These sensors measure the acceleration, position, and several other features associated with the fingers.
- Electromyography (EMG). This measures how the electrical signals evolve through the muscles in the human body, thereby inferring the movements made by the hands and fingers.
- WiFi and radar. These techniques use radio waves to detect the signal changes as propagating waves interact with the human body/gesture movement. These techniques are receiving increasing interest among the research community because of the low sensor costs, their recognition accuracy, and their almost ubiquitous presence, particularly with respect to WiFi signals [13–16].

Stereo vision is a passive 3D imaging technology. It is used to obtain a 3D scene from two calibrated imaging sensors; it extracts the depth information of the 3D scene via triangulation between the imaging sensors based on image correspondences. In stereo vision, it is important to find the matched features and image correspondences. The accuracy of the extracted depth map is limited by the ambiguities of feature matching.

- Alternative approaches. Flexible sensing technologies can be adapted to the hand structure. Other approaches include mechanical and ultrasonic sensors/technologies.

Among these techniques, human gesture recognition using 3D information obtained from RGB-D sensors and camera-array setups has seen a significant increase in research interest. RGB-D sensors are usually based on time-of-flight (ToF) technologies, or structured light (SL) approaches. In contrast, the 3D information obtained from a camera-array setup is inferred using the different positions of the cameras. It is a passive sensing acquisition technology, applied in cases where active sensing technologies are inadequate, mainly in outdoor environments and for non-controlled illumination conditions.

Each type of sensor has its own advantages and drawbacks. As summarized elsewhere [17], gloves are precise but uncomfortable, ToF sensors depend heavily on the scene geometry, and RGB-D sensors perform best indoors. For single cameras, durability is an issue, and for stereo and multiple cameras, the computational complexity and calibration can create a source of uncertainty. Alternative acquisition systems and technologies related to human gesture recognition and analysis are described elsewhere [18–21]. In some cases, the acquisition occurs under relatively controlled environmental acquisition conditions. In other cases, specific and simple-to-identify gestures are considered, using an RGB-D sensor.

Regarding the second processing step (feature characterization), we should stress that, depending on the type of information, we can generate a gesture representation from two different perspectives [22]:

- Three-dimensional model-based strategies. These approaches help describe the main features of hand gesture in a 2D or 3D space. These methodologies can be sub-divided into (a) volumetric and (b) skeletal models. The former takes the space occupied by the hand gestures and its dynamic behavior into account, whereas the latter interprets the hand gestures as a group of angles (and parameters derived from them) and segment lengths.
- Appearance-based approaches. Features are derived directly from images or videos and compared with those obtained from gesture templates.

An effective way to characterize gestures is using 3D information. Three-dimensional data, including depth data, provide a rich description of a scene, particularly when compared to the analysis of a 2D image, because 3D data contain additional information pertaining to the third (z) space coordinate. Stereo cameras and multi-array systems can be used to extract or infer 3D information, including depth information. Range cameras can be used to combine depth information with 2D intensity cues.

Human action recognition and, in particular, gesture recognition presents a series of challenges, including (1) the presence of occlusions; (2) illumination conditions, such as illumination regimes and the presence of non-homogeneities; (3) complex backgrounds; (4) intra- and inter-class similarity and the variability of actions, for example, each action performed by an individual is unique; and (5) dynamic features of the movement. All of these conditions affect the classification performance of any type of gesture recognition system substantially.

A ToF 3D imaging system consists of a passive CMOS image sensor (camera) with an active modulated light source such as a laser or light-emitting diodes (LEDs). The active light source is used to illuminate the scene, and the light reflected from the scene is acquired using a dedicated sensor. The phase difference between the emitted and reflected light is used to estimate the distance and depth information for the objects in the scene. In order to estimate the corresponding depth information, ToF requires highly accurate wave delay measurements. It is used typically for indoor and short-range applications.

Current 3D optical imaging technologies can solve some of the abovementioned problems in gesture recognition. They outperform systems based on 2D information when dealing with partial actions, gesture obstructions, and low illumination conditions.

3. BASIC THEORY OF 3D OPTICAL CAPTURE AND IMAGING SYSTEMS

In Section 2, we presented a succinct overview of the acquisition technologies currently used in human gesture recognition. In Section 3, we focus our attention on acquisition technologies that can provide 3D human gesture data using optical and RGB-D information.

The first of these techniques is integral imaging, which is a 3D technique designed specifically for acquiring the spatio-angular information of the rays emitted by 3D scenes. This is facilitated by capturing multiple perspectives, with both horizontal and vertical parallaxes, in ambient light, i.e., under polychromatic and spatially incoherent illumination [23–38].

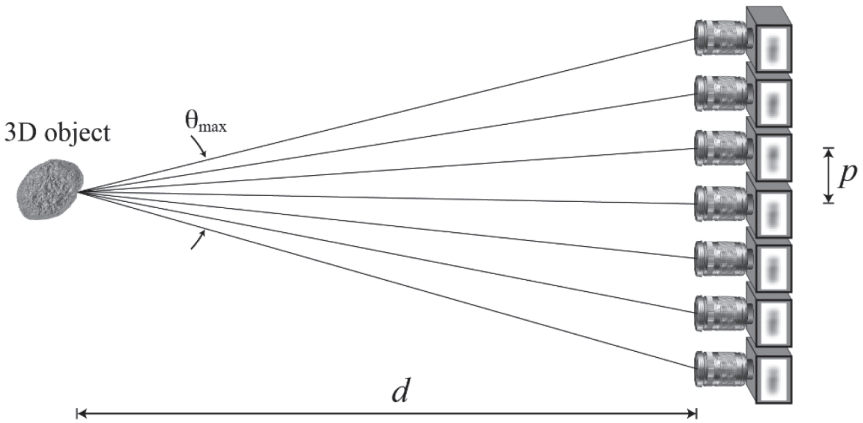
3.1. Array of Digital Cameras

At present there are three architectures for implementing the Lippmann concept for the capture of 3D information of macroscopic scenes. First, we analyze the case in which the Lippmann concept is implemented using arrays of digital cameras [39,40] (see Fig. 2).

The camera-array solution has many advantages, the first being the high lateral resolution of any elemental image, which is determined by the sensor resolution. The second is the amount of parallax, which is higher (than for the case of the lenslet array) and can be fixed at will (within certain limits). Additionally, one can utilize all the capabilities of digital cameras, for instance, changing the f-number, the focus plane, or even the focal length of the objective. Figure 3 shows an example of the multi-perspective information that can be captured with this type of system [41].

From this collection of elemental images, it is possible to reconstruct the 3D structure of the scene computationally by using a simple algorithm in which, in the first step, all the elemental images are superposed over the central one. Then, all the pixels with the same lateral coordinate are added to give the value of the refocused pixel at infinity. In the next step, all the elemental images except the central one are shifted by one pixel and the same summation is performed to obtain the refocused image at a closer distance. The iterative application of this process provides the complete refocused stack. An example is shown in Fig. 4, where we show three refocused images from the stack. Note from the figure that not only is a bokeh effect obtained, but also some occlusions are overcome.

Figure 2



Scheme of an integral imaging capture device comprising a set of equidistant digital cameras. p is the camera pitch, d is the range, and θ_{max} is the field of view.

The actual depth of refocused planes depends on the parameters of the capture architecture according to the following formula:

$$z_R = f\frac{N}{n_S}, \tag{1}$$

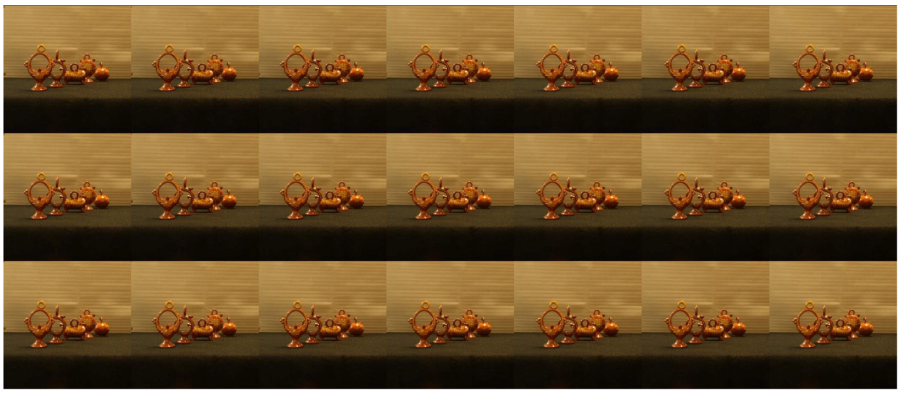
where the distance z_R is measured from the plane containing the objectives of the digital cameras, f is the focal length of the photographic objectives, N is the number of pixels of the CCD, and n_S is the number of pixels that the elemental images have been shifted by in the calculation of the refocused image ($0 \leq n_S \leq N$). By calculating the first derivative of Eq. (1), it is easy to find the distance between consecutive refocused planes:

$$\Delta z_R = \frac{z_R^2}{fN}. \tag{2}$$

Finally, the depth of field of any refocused image can be calculated as described in [34]:

$$\mathrm{DoF}_R = 2\frac{z_R^2}{fNN_H^2}, \tag{3}$$

Figure 3



2D elemental images with 3(V) × 7(H) perspectives obtained from a 3D scene by Martínez-Corral and Javidi [41] using a camera array. Each 2D elemental image shows a slightly different perspective of the 3D scene.

Figure 4

3D reconstructed images at different depth planes obtained from the study by Martínez-Corral and Javidi [41].

where N_H is the number of elemental images along, for example, the horizontal direction.

As an example, we can consider the case in which a 7×7 array of cameras, with $f = 25$ mm and $N = 960$ pixels of 4.5 μm, are used to capture a 3D scene extended from $z_1 = 1.00$ m to $z_2 = 2.00$ m. In this case, the number of refocused images within this interval is equal to 8, the lateral resolution ranges from 0.3 mm to 0.5 mm, and the DoF_R ranges from 2 mm to 4 mm.

Integral imaging is a passive auto-stereoscopic 3D sensing and imaging technique that can provide 3D images with full parallax and continuous viewing angles. In the image capture process, the intensity and the directional information of a scene are recorded by using an array of cameras. Thus, the multiple perspectives of the 3D scene are acquired during the camera pick-up process. The integral imaging reconstruction is the reverse of the capture process; the voxels are reprojected into the 3D space at specific depths in order to reconstruct the 3D scene.

3.2. Stereo Cameras

Stereo cameras can be considered as a case of the Lippmann setup. Thus, the same formulae apply for the calculation of z_R, Δz_R, and DoF_R. The main difference is that in this case there is no vertical parallax, with the result that the blurring of refocused images spreads only in the horizontal direction (in the form of duplicated images). Another consequence is that the DoF_R is much larger (it increases from 24 mm to 48 mm using the data from the previous example). As an example, we have extracted two views from Fig. 4 (the upper left and the upper right corners) and calculated the refocused image at three depth planes (shown in Fig. 5).

Note, however, that stereo images are not often used for the calculation of refocused images but, typically, for the calculation of 3D point clouds. To achieve this, a convolution is performed usually over a window of 3×3 or 5×5 pixels. This gives rise to depth maps with a lateral resolution of one-third (or one-fifth) of that of the stereo images and an axial resolution provided by Δz_R [defined in Eq. (2)].

Thus, relative to integral imaging, stereo images have the advantage of being much less bulky and faster for calculation, but have the drawback of providing 3D information with significantly lower axial and lateral resolution.

Figure 5

Refocused images at three depth planes from a stereo pair.

3.3. Plenoptic Camera

Another method for implementing the Lippmann concept is the so-called plenoptic camera [31–33], which is realized by placing an array of microlenses at the image plane of a conventional photographic camera. As the schematic in Fig. 6 depicts, all the rays emitted by the central point of the reference object plane intersect after passing through the camera objective at the optical center of the central microlens. Then, they propagate to the pixelated sensor, where each pixel collects the rays within a given inclination-angle range. A similar process occurs in the other microlenses of the array. Note that any pixel within a microimage has information corresponding to a specific ray inclination.

The main advantages of plenoptic cameras are their simplicity and low cost, and the possibility of acquiring the plenoptic frame in a single CCD with a single shot. A drawback is that they provide much lower parallax and resolution. However, modern technologies used for producing microlens arrays (MLAs) with small pitch and sensors with tiny pixels allow plenoptic cameras to produce elemental images with satisfactory resolution. Therefore, the main current drawback of these cameras is the low parallax when imaging objects that are far away.

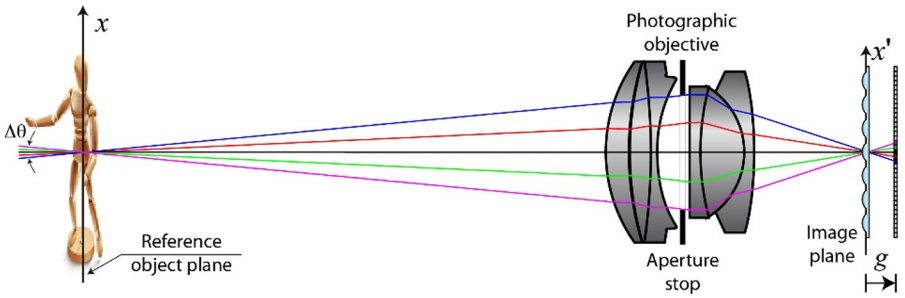
The collection of microimages of a given 3D scene is nothing but a sampled map of the spatio-angular information of the light field coming from the 3D scene. This image collection is typically known as the integral image, but it is also known as the plenoptic image or light field image. Next, in Fig. 7 (left), we show an example of a plenoptic frame.

The plenoptic frame has the appearance of a low-resolution image of the 3D scene. The "pixels" of such an image are circular microimages. However, looking deeper into the frame [see the magnified inset in Fig. 7 (left)], we find that the microimages are not homogeneous inside, but have a structure. Sharp images are not found within the microimages, possibly because they are far from being conjugated with the scene. Therefore, the plenoptic frame does not provide a collection of multi-perspective elemental images directly. Nevertheless, it contains this information, which can be assessed easily by applying the following pixel mapping:

$$\mathbf{CEI}_{i,j}(p, q) = \mu \boldsymbol{I}_{p,q}(i, j). \tag{4}$$

In this equation, $\mathbf{CEI}_{i,j}$ denotes the (i, j) calculated elemental image, and $\mu \boldsymbol{I}_{p,q}$ denotes the (p, q) captured microimage. Using a collection of $J \times J$ microimages,

Figure 6



Schematic of a plenoptic camera. The curved side of the lens array is conjugated with the reference object plane, whereas the sensor is conjugated with the aperture stop.

each with $K \times K$ pixels (usually $J \gg K$), one can calculate a collection of $K \times K$ elemental images, each with $J \times J$ pixels. As an application of the pixel-mapping procedure, in Fig. 7 (center), we show the elemental images calculated from the microimages shown in Fig. 7 (left).

As the calculated elemental images (CEIs) obtained from a plenoptic frame are the result of a hybrid procedure that includes both optical capture and computational pixel mapping, the features of such images are influenced by both optical and computational factors, such as the angle of maximum parallax, which can be evaluated in terms of the f-number of the lenslets as

$$\theta_{\max} = \frac{|M_{\mathrm{ob}}|}{f_{\#}^{ML}}, \tag{5}$$

where M_{ob} is the lateral magnification of the host camera.

Therefore, we can conclude that the smaller the $f_{\#}^{ML}$ value, the higher the field of view (FOV) and the parallax of the plenoptic camera. However, this is not a free parameter, as it is linked to the f-number of the camera objective.

As in the case of the camera array, the resolution limit of CEIs is determined by the Nyquist theorem. Here, Nyquist establishes that two points in the object are resolved in the CEI provided that their images obtained through the plenoptic camera correspond to different microlenses, leaving one empty in between. Thus,

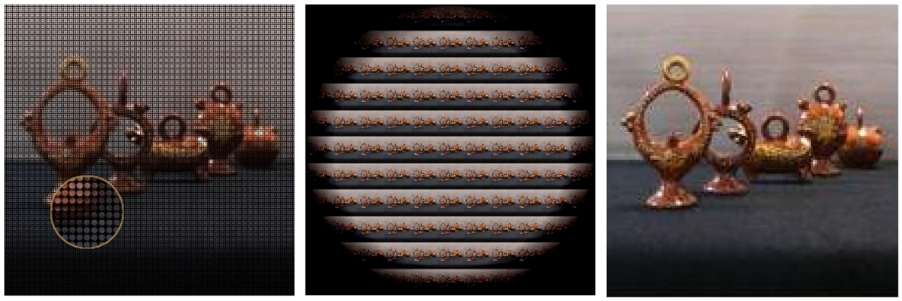
$$\rho_{C\mathrm{pix}} = 2\frac{p}{|M_{\mathrm{ob}}|}, \tag{6}$$

where p is the pitch. However, to avoid the diffraction effects that are detrimental to the ray-optics nature of integral imaging (or light field) technology, the size of the diffraction spot over the MLA ($1.22\lambda_0 f_{\#}^{ML}$) should be of the order of (or lower than) one-fifth of p [41–43]. Then, the spatial resolution of the CEIs would be approximately 10 times worse than that of the host camera.

To calculate the position of the refocused images, we need to adapt Eqs. (1) and (2) to this architecture. Then,

$$z_R = \frac{f^2}{f_{ML}}\frac{n_S}{J}, \tag{7}$$

Figure 7



(Left) Plenoptic image of a 3D scene. The image is composed of an array of tangent microimages; (center) elemental images calculated from the plenoptic image; and (right) central elemental image. These images were obtained from the study by Martínez-Corral and Javidi in [41].

where the distance z_R is measured from the front focal plane of the objective of the plenoptic cameras, f is the focal length of the plenoptic objective, J is the number of pixels behind any microlens, and n_S is the number of shifted pixels ($-J \leq n_S \leq J$). Note that although the number of refocused planes is now much smaller, they are equidistant.

We can conclude that the advantages of plenoptic cameras are that they are compact and portable. However, compared with the array of digital cameras, they provide refocused images with significantly worse resolution, parallax, and depth density.

3.4. Structured IR Patterns

An alternative device for recording the information of a 3D scene was launched initially by Microsoft under the name of Kinect in 2010, as an add-on accessory for the Xbox game console. The distinctive hallmark of this device is its capability to record the RGB image and the depth information simultaneously in real time. This is possible because the Kinect has two different cameras that operate with the same resolution [44]: an RGB camera and an infrared (IR) camera. The principle behind the Kinect technology is based on depth mapping obtained from projected structured IR patterns. The Kinect's IR emitter projects a fixed pattern onto the target, and both the depth distance and the 3D reconstructed map are obtained from the reflected pattern recorded by the IR camera [45,46]. The practical depth information provided by the Kinect ranges between 1000 mm and 3000 mm. The resulting depth map has a lateral resolution of 320 × 240 pixels, which, when projected onto the object space, results in an angular resolution limit of approximately 0.4°, which translates to a resolution limit ranging between 7.0 mm and 21.0 mm. On the other hand, and in accordance with the expected quadratic depth resolution associated with triangulation-based devices, the quantization step, q [mm], is related to the target distance, z [mm], through the function [47]

$$q = 2.73z^2 + 0.74z - 0.58. \tag{8}$$

Thus, for the practical axial range, the depth resolution ranges from 3 mm to 25 mm.

We can conclude here that the main advantage of these setups is that they provide RGB-D maps in real time. However, their axial and lateral resolutions are far from being comparable with those provided by the integral imaging camera arrays.

3.5. Time-of-Flight Cameras

Time-of-flight (ToF) cameras are light and compact 3D image sensors that provide depth maps with high frame rates. Originally proposed in 1977 [48], the technique only became widespread after the appearance of the first prototype of a CCD-based ToF camera, in 1999 [49]. ToF cameras are based on measuring the phase shift between IR flashes emitted at high frequency by a modulated source and the signal received after reflection from the surface of the 3D scene.

If we focus our attention, for example, on the ToF camera associated with the Kinect v2, we find that the provided depth map has a lateral resolution of 512 × 424 pixels. Taking into account the focal length of the IR objective, $f = 3.67$ mm, an angular resolution of around 0.15° is obtained [50]. However, as expected in a technology based on wave propagation, the depth resolution falls proportionally to the squared distance. According to [51],

$$q = 0.542z^2 - 0.885z + 2.708. \tag{9}$$

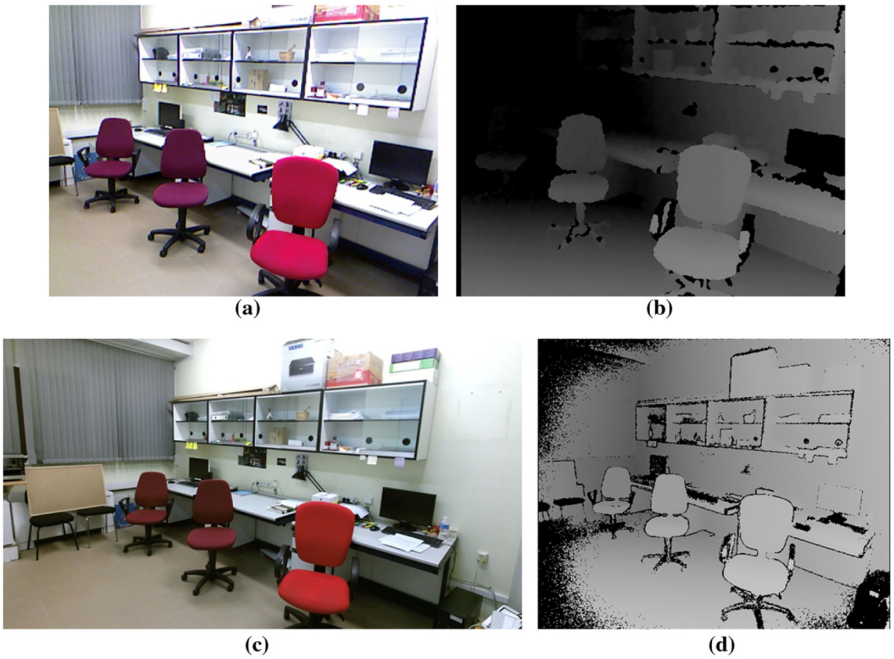
Thus, for the same axial range as in the previous section, 1.0 m to 3.0 m, the depth resolution is 2.5 mm to 5 mm.

We should consider, however, that Eq. (9) is not a universal law, but rather a general trend, as the depth precision has been demonstrated to depend on several factors, including the lighting conditions, object color, object reflectance, and type of scene [52]. Finally, Fig. 8 shows an example of the different depth maps that the two versions of Kinect can provide.

We can conclude that the system's performance depends on the type of scene. However, in general terms, the ToF technology offers better possibilities for human gesture recognition than the IR structured dots technology. When compared with integral imaging implemented via a camera array, ToF has the advantage of offering impressive results using a compact light setup. However, integral imaging has the advantages of being a passive technique and offering better 3D resolution, albeit using a bulkier setup.

Table 2 compares different types of imaging systems that can be used for human gesture recognition, in terms of properties that are considered the most distinctive, in order to provide high-quality gesture data acquisition.

Figure 8



Captured images from two versions of Kinect: (a), (b) Kinect v1 and (c), (d) Kinect v2. Both pairs of images are captured from the same standpoint. (Reprinted from [50].)

Table 2. Imaging Systems Used for Human Gesture Acquisition

	Compactness	Depth-Map Lateral Resolution	Depth-Map Axial Resolution	Depth Refocusing	Overcome Occlusions	Active/ Passive	Feature Dependence
Integral imaging camera array	Low	High	High	High	High	Passive	Medium
Stereo camera	High	Low	Low	Low	Low	Passive	High
Plenoptic camera	High	Low	Low	Medium	Medium	Passive	High
IR structured dots	High	Medium	Medium	Medium	No	Active	Medium
IR time-of-flight	High	High	High	Medium	No	Active	Low

3.6. 3D Integral Imaging in Degraded Environments

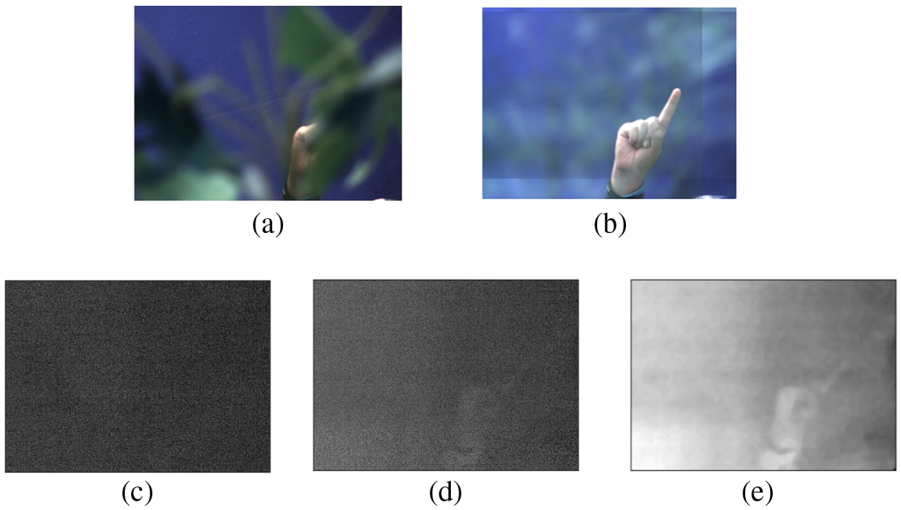
The integral imaging-based methods discussed in Subsections 3.1–3.3 have some advantages over other acquisition technologies in cases where imaging is carried out in poor or degraded conditions. In fact, the information extraction under degraded conditions is a complex process. Examples of degraded conditions considered herein include the existence of partial occlusion and imaging in low illumination environments. Such situations may occur when imaging through biological tissues, e.g., in night vision imaging, single photon emission tomography, visualization in dense smog or fog, remote sensing in geographic science, and sensing in fire and rescue operations [53–56]. To provide high-quality imaging solutions for such scenarios, a variety of methods have been proposed [57–60]. Among them, integral imaging has unique advantages. In this section, we provide a brief overview of integral imaging-based 3D sensing and visualization in situations where partial occlusion and low lighting present imaging challenges. The corresponding applications and experimental results for human gesture recognition under degraded conditions will be discussed in Subsection 4.2c and Subsections 5.1 and 5.2, respectively.

3.6a. Depth Extraction and Occlusion Removal

The 2D imaging sensors can only capture the intensity information of a scene; the depth information is not recorded. With an integral imaging-based system, the depth information relating to a real-world scene can be recorded from multiple perspectives as elemental images, so that we can extract the 3D information by reconstructing the captured multi-perspective pixels at the corresponding in-focus planes.

In addition, when the object of interest in a 3D scene is partially occluded, other 3D sensing systems, such as structured IR and ToF cameras (see Subsections 3.4 and 3.5), may not be able to provide accurate depth maps. In contrast, the reconstruction algorithms utilized in integral imaging provide a sharp image at the in-focus depth, while the out-of-focus objects show strong blurring. Thus, the occluded object can be extracted when there is partial occlusion.

Figure 9



Under regular illumination: (a) 2D elemental image of a human gesture with partial occlusion and (b) corresponding 3D reconstructed image with occlusion removal. Under low illumination: (c) 2D elemental image of a human gesture with partial occlusion, (d) corresponding 3D reconstructed image, and (e) corresponding 3D reconstructed image modified using the total variation (TV) denoising algorithm. (Reprinted from [83].)

3.6b. Imaging in a Low Illumination Environment

Conventional 2D imaging sensors designed for the visible wavelength range may not exhibit high-quality performance in photon-starved environments. One potential explanation for this is the low incident power (flux) coming from the scene—while the performance may also be compromised by different noise sources from the camera, e.g., read noise, dark current, and photon noise. Specific imaging systems have been developed for sensing and visualization at night such as night vision imaging and long-wave infrared (LWIR) imaging systems. However, the imaging systems mentioned above may suffer from low image resolution and low sharpness. In addition, they can be bulky and expensive. In [61], the potential of integral imaging-based 3D visualization of objects under extreme photon-starved conditions was demonstrated using conventional imaging sensors, such as cooled CCD. It has been further applied to target recognition in low light environments [62].

Under extremely low illumination conditions, the camera noise level can be even greater than the signal, leading to a low signal-to-noise ratio (SNR). To reduce the effects of noise and reconstruct the original object simultaneously, total variation (TV) denoising algorithms are applied. The total variation penalized maximum-likelihood expectation maximization (TV-PMLEM) algorithm has been reported for compressed imaging with extended space-bandwidth, and it has been applied for integral imaging reconstruction [63,64].

Figure 9 shows the advantage of integral imaging-based 3D visualization for human gestures in degraded environments. Figure 9(a) illustrates a 2D elemental image of a partially occluded human gesture under regular illumination. The 3D reconstructed image focused at the gesture depth is shown in Fig. 9(b). With integral imaging sensing and reconstruction, the occlusion in the foreground can be removed. However, under low illumination conditions, both the 2D image [see Fig. 9(c)] and the 3D reconstructed image [see Fig. 9(d)] have very low viewing quality due to the noise level. By applying the TV algorithm to the 3D reconstructed image, as shown in Fig. 9(e), the image quality is enhanced significantly. Figure 9 indicates that integral imaging has the potential for human gesture recognition in degraded environments, such as those where there is partial occlusion and low illumination. A detailed discussion of the corresponding algorithms and experiments can be found in the following sections.

4. OVERVIEW AND EXPLANATION OF ALGORITHMS FOR HUMAN GESTURE RECOGNITION

Human action recognition [65] is a broad research field including several topics. Human gesture recognition is a specific type of human action recognition process (Fig. 10), for which we can distinguish two main methods whereby a person performs gestures: body gestures and hand gestures. Body gesture recognition is related to human body part gestures that encode some information, such as using arm positions to communicate messages. In the case of hand gesture recognition, the information is encoded by means of certain positions and configurations of the hand and fingers.

This tutorial is aimed at explaining hand gesture recognition, which is one of the most relevant types of human action recognition used to communicate information. More specifically, we address hand gesture recognition using 3D image acquisition techniques, which provide data sources for this recognition task.

Techniques for 3D hand gesture recognition using depth information can be categorized into three groups [4]: (1) static, (2) trajectory, and (3) continuous. All of these categories can use additional 3D hand modeling information for finer hand

gesture recognition, for instance, to recognize hand gestures with finger movements. Furthermore, in continuous or dynamic hand gesture recognition, these techniques can determine when a gesture starts and when it ends from hand motion trajectories. Static gesture recognition uses only a selected frame or still image representing the gesture; therefore, this method is not adequate for complex hand gestures where temporal information is a key factor, such as certain hand part trajectories.

The most successful approaches are appearance-based techniques [4], which are more robust to sensing and context conditions as model-based methods are based on higher conceptual models and rely on an accurate extraction of the hand or human model; small errors in extracting the hand or body parts produce frequent recognition failures.

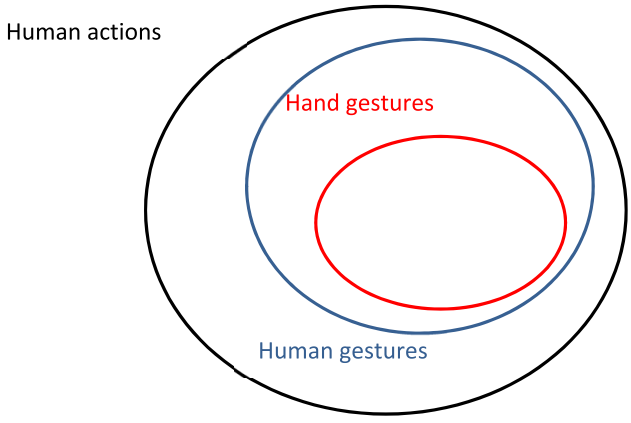
In recent years, image and video technologies have made the acquisition of 3D video in real time feasible, which has led to significant progress in research on human action recognition. The main 3D image and video acquisition technologies currently available are ToF, RGB, RGB-D, and integral imaging [4]. ToF can be included as part of RGB-D technologies, since the depth measured by the RGB-D sensors can also be obtained using ToF techniques. The other common depth sensing technique used in RGB-D technologies is structured light (SL). In addition, RGB-D includes RGB images associated with depth data.

Both ToF and SL are active sensing techniques, involving the projection of infrared light on scene objects to recover the 3D structure. In contrast, integral imaging is a passive sensing technique, which brings many advantages in applications where active sensing technologies are inadequate, such as outdoors and in non-controlled illumination scenarios.

Currently, RGB-D and integral imaging can be considered the principal state-of-the-art 3D image and video data acquisition methods used in human gesture recognition and, more specifically, in hand gesture recognition. These 3D image and video technologies have led to the development of different methods and techniques, depending on the 3D image acquisition technology and the way the 3D data are pre-processed.

The main stages of recognition are characterization and classification. There are two main approaches to achieve this. The classical approach is to use extracted features designed for the problem and then apply a general classification technique, for instance, *k*-Nearest Neighbors (kNN) or support vector machines (SVM). The other approach is an end-to-end process, such as those utilizing some types of artificial neural networks (ANNs), where the raw data measured by the sensor represents the

Figure 10



Scope and relationship between human gesture recognition domains.

system input and the output is the classification result. In these approaches, feature characterization and classification are learned by the system and no ad-hoc designs for feature extraction are needed.

Subsection 4.1 introduces some representative approaches for 3D image/video characterization techniques that have been applied successfully to hand gesture recognition, including general 3D video characterization features, such as 3D local spatio-temporal patterns, and more specific features, such as 3D silhouettes. Furthermore, some examples of classical, end-to-end, and specific classification techniques used predominantly in 3D hand gesture recognition analyses are described in Subsection 4.2, including SVMs, ANNs, and correlation filter approaches.

4.1. Three-Dimensional Image/Video Characterization

Given a 3D image/video data source, the first step in hand gesture recognition is the extraction of basic image and video features that will be used in the final recognition step. Some authors have proposed different taxonomies for 3D image/video characterization [4,65]. Table 3 lists a representative selection of 3D image/video characterization methods for 3D hand gesture recognition, based on the most common types of visual features chosen as input data.

The techniques based on *3D local occupancy patterns*, *3D silhouettes*, and *3D optical flow* use RGB-D video as the input information. In these types of techniques, RGB information is used only to segment the region of interest (ROI), with depth information used to characterize the gesture. These techniques offer different strategies to deal with the depth information, either describing the 3D spatial information of regions in voxels or analyzing the 3D optical flow that characterizes the action movement.

Local spatio-temporal interest points is a technique that allows the action recognition based on characterizing a monochrome or RGB video. This strategy is adapted to 3D optical video using integral imaging, which is used to reconstruct the image sequence at the depth of the ROI, obtaining a 3D volume along the spatial (x, y) and the temporal t axes, where spatio-temporal interest points are extracted and used as the input data for classification.

The final characterization method discussed here is an end-to-end learning technique and a hot topic in the field of deep learning (DL) that exploits *convolutional neural networks* (CNNs). Using this approach, it is possible to train complex learning systems, bypassing the intermediate stages typically associated with traditional recognition designs, such as feature extraction design and the type of classifier used. In this case, the input data are raw video RGB-D, with all channels used in the gesture recognition. The CNN approach can also be used as a feature extraction process, using only the auto-encoder part of the network as input data for other types of classifiers.

Table 3. Main 3D Image/Video Characterization Methods for Hand Gesture Recognition

Characterization Technique	Main Property
3D local occupancy patterns	Local features extracted in 3D point cloud sequences from random 4D (x, y, z, t) sub-volumes
Local spatio-temporal interest points	Local features extracted in 2D image videos from Harris 3D (x, y, t) interest point detector
3D silhouettes	2D features extracted from the 3D plane of the hand position
3D optical flow	Temporal features from optical flow of the video sequence
Self-learning CNN features	Features learned by the auto-encoder layers of a convolutional neural network

4.1a. Three-Dimensional Local Occupancy Patterns

Three-dimensional local occupancy patterns [113] are semi-local features extracted from 4D sub-volumes sampled at random (Fig. 11). Semi-local implies local extraction from the image, but with a global characterization of the object of interest. Each depth video sequence is treated as a 4D volume in space and time, containing each 3D pixel at a specific instant in time, which can be represented as $I(x, y, z, t)$. Sub-volumes can be defined from the entire video 4D volume, by fixing a pixel center and a sub-volume size around the center point (x, y, z, t). Each pixel in the 4D video volume $I(x, y, z, t)$ has a value of either 1 or 0, depending on whether there is a 3D object point at this location and time. The *occupancy pattern* of a sub-volume whose pixel center is located at (x, y, z, t) is expressed as

$$o(x, y, z, t) = \delta\left(\sum_{q \in \mathrm{SV}(x,y,z,t)} I(\mathbf{q})\right), \tag{10}$$

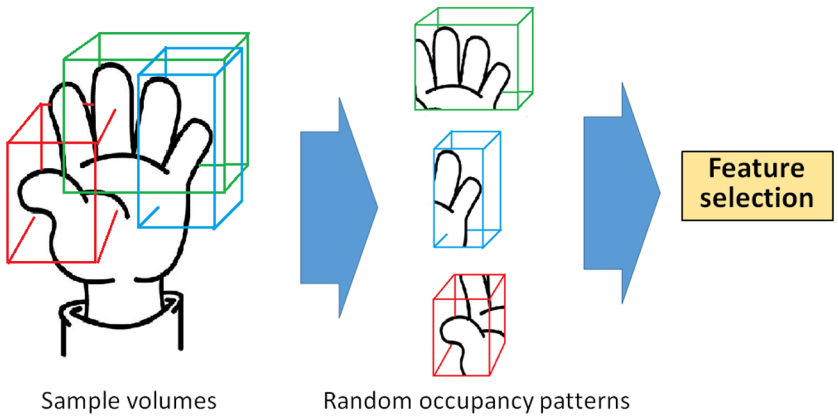
where $\mathrm{SV}(x, y, z, t)$ is the sub-volume centered at (x, y, z, t), q is the point in that sub-volume, and $I(\boldsymbol{q}) = 1$ if the point $\boldsymbol{q} = (x_q, y_q, z_q, t_q)$ is a 3D object point; otherwise, $I(\boldsymbol{q}) = 0$. $\delta(\cdot)$ is a sigmoid normalization function: $\delta(x) = 1/(1 + e^{-\beta x})$. Notably, this occupancy pattern can be computed efficiently using incremental algorithms for high-dimensional images [68].

To characterize the video sequence, dense sampling of the sub-volumes can be performed using uniform sampling strategies, i.e., uniform sampling of the 4D video volume and assigning a random sub-volume size to each point. A more efficient sampling method is based on a weighted sampling approach, which characterizes each sub-volume using a scatter class separability measure [68] that is based on the class scatter matrices defined as

$$\boldsymbol{S}_B = \sum_{i=1}^{C} (\boldsymbol{m}_i - \boldsymbol{m})\,(\boldsymbol{m}_i - \boldsymbol{m})^T, \tag{11}$$

$$\boldsymbol{S}_W = \sum_{i=1}^{C} \sum_{j=1}^{n_i} (\boldsymbol{b}_{ij} - \boldsymbol{m}_i)\,(\boldsymbol{b}_{ij} - \boldsymbol{m}_i)^T. \tag{12}$$

Figure 11



Gesture characterization by sampling 4D (x, y, z, t) sub-volumes at a given t. The rest of the gesture characterization process includes feature selection and a sparse coding phase prior to support vector machine (SVM) classification. This approach is presented in [113].

Here, C is the number of classes, n_i is the number of samples in class i, $\boldsymbol{h}_{ij}$ is the feature vector of the jth sample of the ith class, $\boldsymbol{m}_i$ is the mean vector of the features $\boldsymbol{h}_{ij}$ in the ith class, and $\boldsymbol{m}$ is the mean vector of the data samples. The class separability measure J can be defined as $J = tr(\boldsymbol{S}_W)/tr(\boldsymbol{S}_B)$, where $\boldsymbol{S}_W$ is the *within-class scatter matrix*, and $\boldsymbol{S}_B$ is the *between-class scatter matrix*. Each point p is characterized by means of eight Haar feature vectors $\boldsymbol{h}$ from a sub-volume centered at point p.

Let us define V as the 4D volume of the entire depth sequence. For each pixel $p \in V$, the class separability score J_p at the pixel p is computed. Thus, the probability for a sub-volume R to be sampled is defined as $P_R^{\text{sampled}} = P_R^{\text{uniform}} \cdot P_R^{\text{accept}}$, where

$$P_R^{\text{accept}} = \frac{|V|}{\sum_{p \in V} J_p} J_R. \tag{13}$$

Here, $|V|$ is the number of pixels in the 4D video sequence volume, and $J_R = 1/N_R \sum_{p \in N_R} J_p$, where N_R is the number of pixels in the R sub-volume. A sampled sub-volume R is accepted to characterize the video sequence if its sampled P_R is greater than a fixed threshold.

As a final step, a sparse subset of N_f features is extracted from the occupancy patterns that define each depth video sequence using an elastic-net regularization process, which is based on a linear regression algorithm with L_1 and L_2 regularization terms, i.e., minimizing the following objective function E for N video samples:

$$E = \sum_{i=1}^{N} \left(y_i - \sum_{j=1}^{N_f} \boldsymbol{w}_j \boldsymbol{h}_{ij} \right)^2 + \lambda_1 ||\boldsymbol{w}||_1 + \lambda_2 ||\boldsymbol{w}||_2^2 . \tag{14}$$

Here, λ_1 and λ_2 are the L_1 and L_2 regularization parameters that control the sparsity of the coefficients in vector w and the margin that defines the linear classifier, respectively. From the N_f features extracted, only those with a significant w_j value are selected.

Given N depth video sequences as training data, each new video is expressed as a set of α_i, $(i = 1, \ldots N)$ coefficients that reconstruct the new video sequence as a linear combination of the N training data videos expressed in the sparse feature domain that was extracted previously. The α vector will be the final feature vector representation of each depth video sequence, and it will be used as the input data for further classification or recognition stage(s).

4.1b. Local Spatio-Temporal Interest Points

Spatio-temporal interest points (STIPs) are points extracted from 2D video sequences, which are treated as 3D volumes (x, y, t). The points in this 3D volume are characterized using the histogram of oriented gradients (HOG) and/or the histogram of optical flow (HOF) in a local manner, by defining small 3D volumes around the points.

One approach to extend the use of STIPs involving 3D information involves using integral imaging and reconstructing the video sequence from the integral imaging video at the depth of interest, i.e., the depth where the hand (that characterizes the movement) is in focus. Therefore, the 3D information is embedded in the reconstructed image sequence, which is treated as a 2D video sequence, and the STIPs are extracted from this reconstructed video sequence [7].

The first step in extracting the STIPs involves smoothing the 3D video sequence $f(x, y, t)$ using an anisotropic 3D Gaussian kernel g, with variances that define

different scales for spatial σ_l^2 and temporal τ_l^2 dimensions:

$$L\left(x, y, t; \sigma_l^2, \tau_l^2\right) = g\left(\cdot, \sigma_l^2, \tau_l^2\right) * f\left(x, y, t\right),$$
$$g\left(x, y, t; \sigma_l^2, \tau_l^2\right) = \frac{1}{\sqrt{(2\pi)^3 \sigma_l^4 \tau_l^2}} \exp\left(\frac{x^2 + y^2}{2\sigma_l^2} + \frac{t^2}{2\tau_l^2}\right). \quad \textbf{(15)}$$

Note that the depth is assumed to be fixed where the object of interest (hand) is in the scene, and the integral imaging reconstruction is performed at the plane located at this depth, where the STIPs are being extracted. Given the first-order partial derivatives of the smoothed video volume $L(x, y, t)$ with respect to $\{x, y, t\}$, the first-order derivative matrix

$$\boldsymbol{M}' = \begin{bmatrix} L_x^2 & L_x L_y & L_x L_t \\ L_x L_y & L_y^2 & L_y L_t \\ L_x L_t & L_y L_t & L_t^2 \end{bmatrix} \quad \textbf{(16)}$$

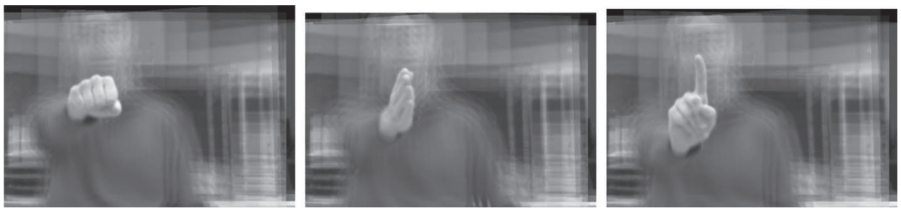
is smoothed to obtain a second-order matrix $\boldsymbol{M} = g(x, y, t; \sigma_i^2, \tau_i^2) * \boldsymbol{M}'$, where $\sigma_i^2 = s\ \sigma_l^2$ and $\tau_i^2 = s\ \tau_l^2$ for a given constant value $\boldsymbol{s}$.

As a final step, a Harris 3D detector is defined from the matrix $\boldsymbol{M}$ as follows: $H = \det(\boldsymbol{M}) - k\mathrm{trace}^3(\boldsymbol{M})$. This is based on an extension of the 2D Harris detector [69]. The STIPs of the 3D video sequence $f(x, y, t)$ correspond to local maxima of H, i.e., points that show large variations in the spatial and temporal dimensions. The Harris 3D detector is applied to detect the STIPs at different spatial $\{\sigma_l = 2^{(l+1/2)} : l \in \{1, 2, \ldots, 6\}\}$ and temporal $\{\tau_l = 2^l : l \in \{1, 2\}\}$ scales. A common technique to deal with short videos is to increase the frame rate, in order to have longer video sequences [69] and obtain more stable STIPs. Other pre-processing techniques prior to extracting the STIPs involve resizing the frames to a mid-level spatial resolution, e.g., 780×270 pixels, and in the case of RGB or multi-band images, applying the Harris 3D detector to a single band, e.g., the R band in the RGB images.

Figure 12 shows examples, for three different gestures, of reconstructed scenes at the depth where the hand is in focus using the R channel from the RGB images. The caption of Fig. 12 includes links to two videos showing where the STIPs are located.

The next stage of video characterization using the STIPs consists of extracting some features from local information around where the STIPs are located. The HOG and HOF are the most popular tools used to characterize STIPs.

Figure 12



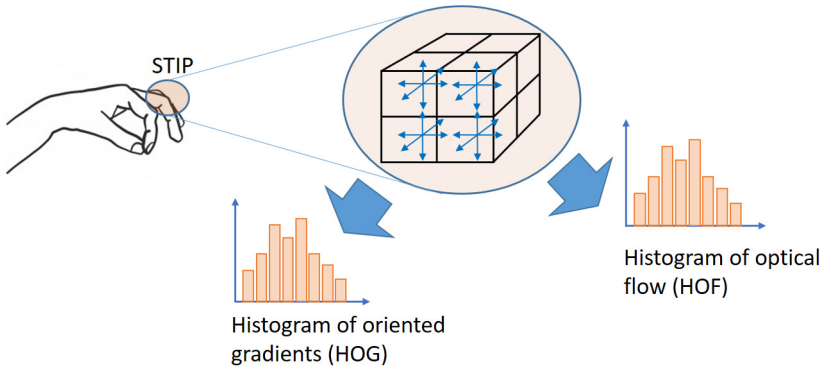
Images corresponding to the 3D reconstruction at the depth where the hand is in focus, for three gestures: (a) open, (b) left, and (c) deny gestures. Spatio-temporal interest points (STIPs) were applied to the videos for this depth reconstruction. Visualization 1 and Visualization 2 show two different gestures, where only the hand is in focus. Visualization 3 and Visualization 4 show the detected STIPs. Most of them appear where the gesture is taking place [7].

A common approach for extracting the HOG and HOF features consists of defining a 3D volume of size ($\delta_x \times \delta_y \times \delta_t$) around the STIP location, depending on the scale at which the STIP was detected, that is, $\delta_x = \delta_y = 18\sigma_l$ and $\delta_t = 8\tau_l$. Further, each 3D volume is sub-divided by a regular grid into $n_x \times n_y \times n_t$ sub-volumes, where $n_x = n_y = 3$ and $n_t = 2$ (see Fig. 13). For a given STIP, once the spatial and temporal gradients in all the frames of the video sequence are computed, the HOG and HOF can be computed for each sub-volume using four and five bins, respectively. All the histograms in each sub-volume are normalized. As a final step, the feature vector representing the STIP is defined as the concatenation of the HOG and/or HOF in different ways. The usual combinations are concatenating 72 HOGs, 90 HOFs, or 162 HOGs + HOFs.

Eventually, the complete video sequence is characterized using the set of STIPs extracted in the sequence and by building a bag-of-words (BoW) model to characterize all the videos in the same feature space. A BoW model (see Fig. 14) consists of defining a set of C visual code words by grouping the STIPs extracted and represented in the HOG/HOF features from a set of training videos.

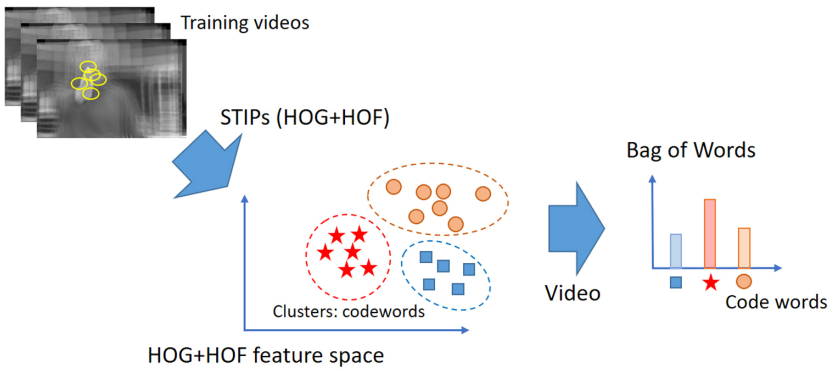
Each $\{\boldsymbol{c}_j, \ j = 1, \ldots, \ C\}$ code word represents a cluster of STIPs that have similar HOF/HOG characteristics.

Figure 13



HOG and HOF characterization of spatio-temporal interest points (STIPs) from the 3D volume around the STIP location. The 3D volume is divided into $\eta_x \times \eta_y \times \eta_t$ sub-volumes where HOG and HOF are computed.

Figure 14



Bag-of-words (BoW) model to represent videos as histograms of code words, which are defined by grouping STIPs from a set of training videos into C code words. HOG, histogram of oriented gradients; HOF, histogram of optical flow.

A BoW model is a powerful tool used to characterize a series of videos in a common feature space, in order to classify them. In particular, a BoW model defines a set of video features (code words), each one representing a cluster of video STIPs. Each cluster has similar HOF/HOG video sub-volume characteristics. In a BoW representation of a video sequence, the STIPs that appear in the video are assigned to the corresponding code word (each one of the C clusters). Code-word counts are then used to build the corresponding histogram.

Therefore, each STIP extracted from a video can be assigned to the most similar code word or cluster. Once the code words are defined, a video sequence can be represented by a histogram of visual code words using the STIPs extracted [66]. Each STIP is associated with a code word; we can find the most similar code-word cluster and then vote for the corresponding visual code word in the BoW representation.

To define the C visual code words, the most common clustering algorithm used is the K-means. After grouping the STIPs from the training videos into C clusters, we can represent each cluster in terms of their mean or cluster center $\{\boldsymbol{\mu}_j,\ \ j = 1, \ldots,\ C\}$. Let $\boldsymbol{x}_i,\ i = 1, \ldots n$ be the n STIPs found in a video sequence. Each STIP $\boldsymbol{x}_i$ is represented in a d-dimensional space corresponding to the HOG/HOF features, where C clusters have been computed previously using their corresponding means $\{\boldsymbol{\mu}_j,\ \ j = 1, \ldots,\ C\}$.

Thus, each STIP $\boldsymbol{x}_i,\ i = 1, \ldots n$ is assigned to the cluster $c(\boldsymbol{x}_i)$ with the nearest mean, i.e.,

$$c(\boldsymbol{x}_i) = \text{argmin}_{j\in\{1,2,\ldots,C\}} \|\boldsymbol{x}_i - \boldsymbol{\mu}_j\|^2. \tag{17}$$

The final BoW representation $\boldsymbol{h} = (h_1, h_2, \ldots h_C)$ of the video sequence is then computed as the visual code-word count of the corresponding STIPs in the video sequence, i.e.,

$$h_j = \sum_{i=1}^{n} \delta\left(c(x_i) = j\right), \quad j = 1, \ldots, C. \tag{18}$$

A 3D video sequence, $f(x, y, t)$, is usually pre-processed via smoothing to eliminate noise. Therefore, we may apply a 3D Gaussian kernel such that $L(x, y, t; \sigma_l^2, \tau_l^2) = g(\cdot, \sigma_l^2, \tau_l^2) * f(x, y, t)$, where σ_l^2 and τ_l^2 denote the spatial and temporal widths of the Gaussian filter, respectively. The STIPs correspond to the maxima of $H = \det(M) - ktrace^3(M)$, where $\boldsymbol{M} = g(\cdot, \sigma_l^2, \tau_l^2) * \boldsymbol{M}'$, and $\boldsymbol{M}'$ is the first-order derivative matrix. The idea behind STIPs is to obtain points where some type of *activity* is happening in the video.

Here, δ is the Kronecker delta function, and h_j is the jth bin of the BoW histogram representation of the video sequence.

A 3D volume of size ($\delta x \times \delta y \times \delta t$) can be defined around each STIP extracted from the image sequence. Each 3D volume is then sub-divided into a regular grid of $\eta_x \times \eta_y \times \eta_t$ sub-volumes.

The HOF considers occurrences in the optical flow estimation in this grid. In both cases, this estimation is made inside each sub-volume of the regular $\eta_x \times \eta_y \times \eta_t$ grid.

Analogously, in the HOGs, the occurrences of common orientations in the gradient are counted in each 3D sub-volume grid defined around a STIP.

Histograms are then computed (the HOF and/or HOG) for each sub-volume through space quantization (into bins), and concatenated to define the final HOF or HOG descriptor.

4.1c. Three-Dimensional Silhouettes

Another technique involves a more detailed characterization of the hand shape. Three-dimensional silhouettes are an adaptation of 2D silhouettes and are conventionally used in computer vision for object recognition. Hand gesture characterization using 3D silhouettes [114] utilizes RGB-D systems to extract the 2D plane in the 3D space where the hand gesture is likely performed. The hand gesture characterization using an RGB-D video sequence comprises the following stages:

1. *Segmentation*. Each video RGB-D frame is first used to segment the depth map into regions where the hand is likely to be.
2. *Tracking and filtering*. The candidate regions found in each frame are used to perform correspondence and tracking with previous frames, in order to find the most probable region where the hand is in the current frame.
3. *Orientation normalization*. Once the hand region is located, the orientation and position of the hand are calculated, and a 3D plane is fitted to approximate the hand palm plane.
4. *Feature extraction*. The hand-depth map is normalized to extract the hand silhouette features.

The feature extraction step consists of the characterization of the hand contour in the plane fitted to the hand palm in each frame (see Fig. 15). Therefore, each RGB-D frame i of the hand gesture video sequence is characterized as a feature vector $\boldsymbol{F}_i = \{\boldsymbol{v}_i, \boldsymbol{r}_i, \boldsymbol{s}_i\}$, where $\boldsymbol{v}_i = \boldsymbol{x}_i - \boldsymbol{x}_{i-1}$ denotes the velocity of the hand center between frame i and $i-1$, where $\boldsymbol{x}_i$ is the 3D centroid position of the hand, $\boldsymbol{r}_i$ is the quaternion (rotation) parameter of the 3D plane of the hand with respect to the 3D reference system, and $\boldsymbol{s}_i = (s_1, \ldots, s_T)$ is a silhouette feature extracted from the T sectors of the hand shape in frame i (see Fig. 15). As the final characterization step for each frame, a principal component analysis (PCA) feature reduction technique is applied, keeping the most significant PCA coefficients (usually 10 to 30 features). This reduced feature vector is then used as the input for the final classification stage.

4.1d. Three-Dimensional Optical Flow

This type of strategy involves generating a 3D flow, characterizing the scene movement, and capturing the depth changes. In [71,72], a stereo camera system was used to estimate the disparity map between a camera stereo pair and the optical flow map between consecutive stereo images. However, flow estimation algorithms require a high computational cost; therefore, it is necessary to use faster approximations to

reduce the cost. In [73], the Fanerbäck algorithm [74] was applied to compute the 2D optical flow of the F_t frame with respect to the previous F_{t-1} frame. Thus, each pixel (x_{t-1}, y_{t-1}) belonging to a ROI of the F_{t-1} frame is reprojected in the 3D vector $(X_{t-1}, Y_{t-1}, Z_{t-1})$, where the Z_{t-1} coordinate corresponds to the depth estimates; that is,

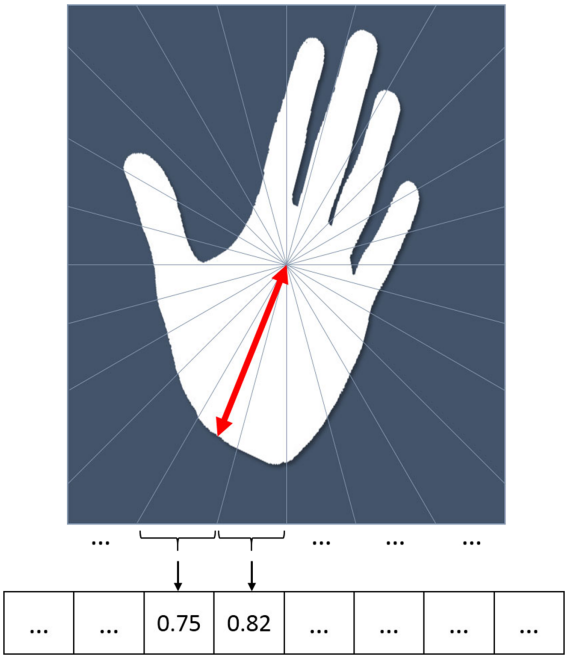
$$\begin{pmatrix} X_{t-1} \\ Y_{t-1} \\ Z_{t-1} \end{pmatrix} = \begin{pmatrix} \frac{(x_{t-1}-x_0)b}{d} \\ \frac{(y_{t-1}-y_0)b}{d} \\ \frac{bf}{d} \end{pmatrix}. \tag{19}$$

Here, b is the baseline of the stereo system, f is the focal length, d is the disparity, and $(x_0, y_0)^T$ is the principal point of the sensor. Analogously, it is possible to perform a reprojection of the point (x_t, y_t) in the F_t frame in the 3D vector (X_t, Y_t, Z_t). Thus, the 3D distance vector of a point between the two frames is defined as $\boldsymbol{d} = (X_t - X_{t-1}, Y_t - Y_{t-1}, Z_t - Z_{t-1})^T$. By normalizing with respect to the L_2-norm, we can obtain the movement of the n pixels of an ROI $(\boldsymbol{d}_1, \cdots, \boldsymbol{d}_n)$.

To generate a compact flow representation for each frame, a 3D HOF (3DHOF) is built. Therefore, the direction of the gestures is codified by a descriptor $\boldsymbol{z}(t) \in R^{h\times h\times h}$, where h is the bin size used to parametrize the space. In addition, each 3DHOF $\boldsymbol{z}(t)$ is normalized, i.e., $\sum_j \boldsymbol{z}(t)_j = 1$. Thus, each 3DHOF stores the ratio of the directions in the current gesture, and the descriptors are invariant to the scale. Figure 16 shows the high-level statistics for a scene.

When computing the HOG of the pixels belonging to an ROI, a new descriptor called the global histogram of oriented gradients (GHOG), $\boldsymbol{h}(t) \in R^{n_2}$, is generated from the

Figure 15



Hand silhouette characterization in the 3D plane fitted to the hand palm. The hand shape is divided into sectors with respect to its centroid and then characterized by the mean distance of the hand sector points from the hand contour. This approach is presented in [114].

depth map, where n_2 is the number of bins in representing the depth. The scale invariance property is preserved by normalizing the descriptor $\sum_j \boldsymbol{h}(t)_j = 1$. Fanello *et al.* [73] applied this descriptor to a depth map in combination with the 3DHOF to characterize gestures.

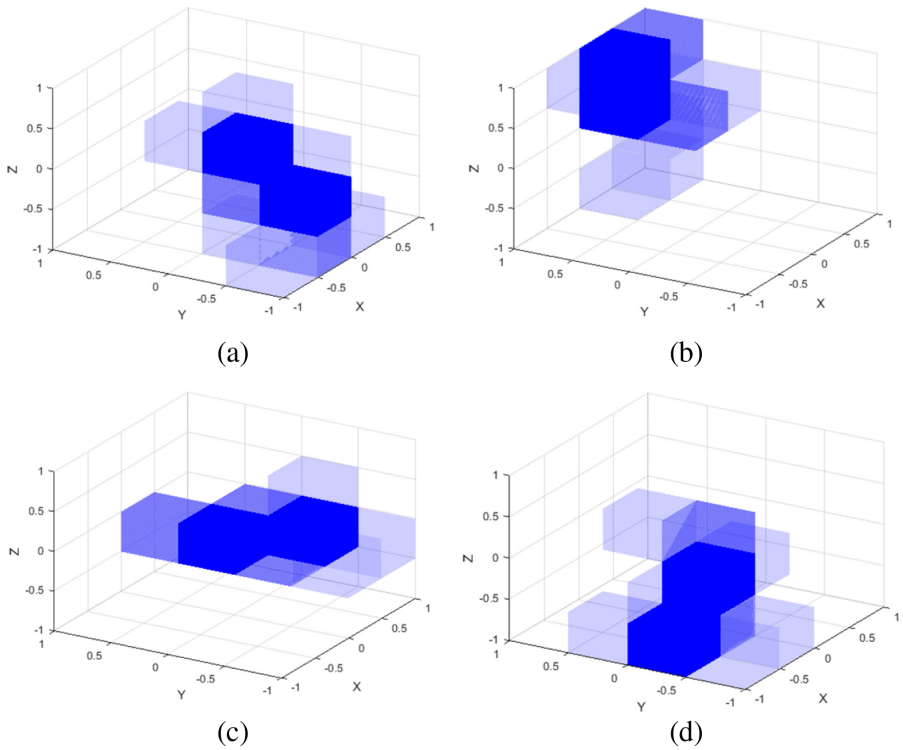
To retain only the relevant information from the data, feature selection is performed. This selection process discards background or body parts not involved in the gesture through a *sparse coding* technique. This method establishes a preliminary step called dictionary learning, where each element of the dictionary is learned from the data. Thus, given a set of 3DHOFs with $\boldsymbol{Z} = [\tilde{\boldsymbol{z}}(1), \ldots, \tilde{\boldsymbol{z}}(m)] \in R^{n_1 \cdot m}$, where m is the number of frames in the training dataset and $\tilde{\boldsymbol{z}}(i)] \in R^{n_1}$ is the feature vector with size n_1, the method builds one motion dictionary $\boldsymbol{D}_M(n_1 \times d_1)$ matrix, where d_1 is the dictionary size, and one code $\boldsymbol{U}_M(d_1 \times m)$ matrix that minimizes the objective function,

$$\min_{\boldsymbol{D}_M, \boldsymbol{U}_M} \|\boldsymbol{Z} - \boldsymbol{D}_M \boldsymbol{U}_M\|_F^2 - \lambda \|\boldsymbol{U}_M\|_1, \tag{20}$$

where $\|\cdot\|_F$ is the Frobenius norm. Note that by fixing $\boldsymbol{U}_M$, we can reduce the above optimization problem to a problem that can be solved using a least-squares algorithm, while given $\boldsymbol{D}_M$, it is equivalent to linear regression with the sparsifying norm L_1.

Similarly, we can define the optimization problem for the GHOG using a dictionary $\boldsymbol{D}_G(n_2 \times d_2)$ matrix and the code $\boldsymbol{U}_G(d_2 \times m)$ matrix for the descriptor

Figure 16



High-level statistics (3D histogram of optical flow) for a scene. The histograms of the scene flow directions at time t for the primitives are shown in the (a) *right*, (b) *left*, (c) *forward*, and (d) *backward* directions. Each cuboid represents one bin of the histogram, where the 3D space is divided into $h \times h \times h$ with $h = 4$. Filled cuboids (cuboids with a darker shade of blue) represent high-density areas [73].

$\boldsymbol{H} = [\boldsymbol{h}(1), \ldots, \boldsymbol{h}(m)] \in R^{n_2 \cdot m}$. Thus, each frame i can be encoded by the concatenation of the two descriptors of the motion and depth $\boldsymbol{U}(i) = [\boldsymbol{U}_M(i); \boldsymbol{U}_G(i)]$.

4.1e. Self-Learning Features: Convolutional Neural Networks

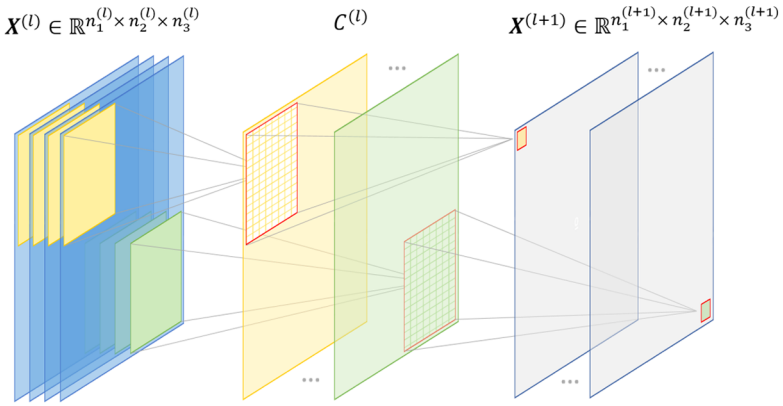
Inspired by biological neural systems, ANNs are computing systems composed of artificial neurons, which receive different values through their input connections, combine these values using a weighted sum with a bias term, and then produce the output result according to a nonlinear activation function. In general, ANNs are able to learn nonlinear mapping functions through a collection of connected layers that represent different abstraction degrees of the input data. Based on this idea, the so-called DL paradigm [75] aims to use multiple layers in order to extract higher-level semantic information from the raw input data without the need to compute handcrafted features. As a result, DL models have seen remarkable success in many computer vision applications and tasks, with human gesture recognition being no exception [76].

Deep neural networks have recently shown great potential in human gesture recognition using 3D optical imaging [77]. The high flexibility of the network design allows the use of DL models at two different segments of the gesture recognition pipeline: (1) feature extraction and (2) gesture classification. In this section, we review the fundamentals of DL for characterizing human gestures from RGB-D data, with a particular focus on CNNs.

From stack auto-encoders to sparse and denoising coding networks [78], many unsupervised architectures have been employed to characterize human gestures from image sequences. The performance of these approaches depends typically on single data modalities, which often makes other network designs more effective when combining RGB and depth information. CNNs have emerged as one of the most popular state-of-the-art technologies for characterizing human gestures from RGB-D images [79].

The general objective of a CNN is to approximate a mapping function of the form $f : \boldsymbol{X} \rightarrow \boldsymbol{Y}$ through the hierarchical concatenation of different transformation blocks. Within the human gesture recognition field, the elemental CNN architecture is made of several convolutional blocks aimed at extracting relevant features from the input data $\boldsymbol{X}$, as well as several final fully connected dense layers that enable discrimination between different gestures $\boldsymbol{Y}$.

Figure 17



Graphical visualization of a convolutional block.

A typical CNN comprises a set of convolutional and pooling layers followed by a fully connected layer and a classification layer. The convolutional and pooling layers filter the input data spatially to produce feature vectors. These feature vectors can be utilized by a fully connected layer and a classification layer for high-level inferencing tasks such as object detection and classification. Mathematically, the spatial filtering considered is a discrete convolution operation (hence its name).

Typically, a convolutional block (l) includes three different components: a convolutional layer, a nonlinear activation step, and a pooling layer. Figure 17 shows a graphical visualization of these operations. The convolutional layer [Eq. (21)] consists of a collection of convolutional filters used to extract different feature maps from the input data. Thus, the kernels act as sliding windows that convolve their weights ($\boldsymbol{W}$) and add their bias ($\boldsymbol{b}$) to each input data location. The nonlinear component [Eq. (22)] then uses an element-wise activation function (H) to generate a nonlinear activity volume that encodes the internal structures and relationships of the data. This function is often implemented by a rectified linear unit (ReLU) defined as the positive part of its input argument. Finally, the pooling layer [Eq. (23)] sub-samples the activation volume to compress the obtained feature maps and to provide a certain degree of robustness with respect to small spatial variations. These processes are expressed as follows:

$$\boldsymbol{C}^{(l)} = \boldsymbol{W}^{(l)} * \boldsymbol{X}^{(l)} + \boldsymbol{b}^{(l)}, \tag{21}$$

$$\hat{\boldsymbol{C}}^{l} = H(\boldsymbol{C}^{(l)}), \tag{22}$$

$$\boldsymbol{X}^{(l+1)} = P(\hat{\boldsymbol{C}}^{(l)}). \tag{23}$$

Figure 18 shows a graphical visualization of a 2D convolution kernel (in blue) and the generation of the corresponding spatially reduced feature maps (in green) using the spatial pooling operator (in orange). Because the 2D-CNN approach can extract convolutional features involving bi-dimensional information only, the temporal and multi-modal nature of recognizing human gestures from RGB-D data makes it suitable to extend the 2D convolutional scheme to higher-dimensional orders [80]. Thus, it is possible to account for the temporal domain and additional information within the feature extraction process naturally. Figure 18 shows (in red) the graphical visualization of a 3D convolution over t input frames. As shown, the 3D kernels extract features from both spatial and temporal domains jointly to capture spatio-temporal information encoded in the adjacent image frames.

Despite the advantages of the 3D-CNN feature extraction scheme, the need for a fixed kernel size makes this approach rigid for managing gestures with a substantially different temporal duration and motion speed [76]. To overcome this limitation, several CNN-based extensions have been proposed in the context of human gesture recognition [78]. Among these proposals, a technique known as temporal pooling [81] has yielded the most accurate results. This approach performs a pooling operation on the convolutional features of all the data frames. As a result, it is possible to obtain temporally reduced feature maps, instead of spatially reduced maps, with variable time depth. Figure 18 shows a conceptual diagram (in yellow) of the temporal pooling process used to obtain temporally reduced feature maps (in green).

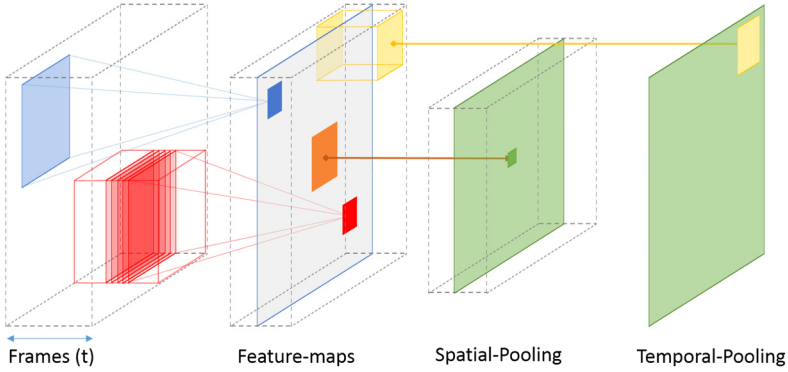
Another important aspect is the process of integrating RGB data and depth information into a CNN-based feature extraction architecture for gesture recognition. One straightforward option to integrate both data modalities consists of stacking together the RGB and depth frames of each timestamp. However, other alternatives have been shown to obtain better results due to the inherent differences between RGB and depth data. Specifically, one of the most successful approaches to integrate these two data modalities is based on using a specific CNN for each modality and then fusing the resulting feature maps for the gesture classification module.

Figure 19 shows a graphical visualization of the CNN-based feature extraction process from RGB-D data where the feature maps generated by the sub-networks are concatenated for the classification system. The fundamentals of gesture classification methods based on DL are further described in a dedicated section in this tutorial.

4.2. Three-Dimensional Image/Video Recognition

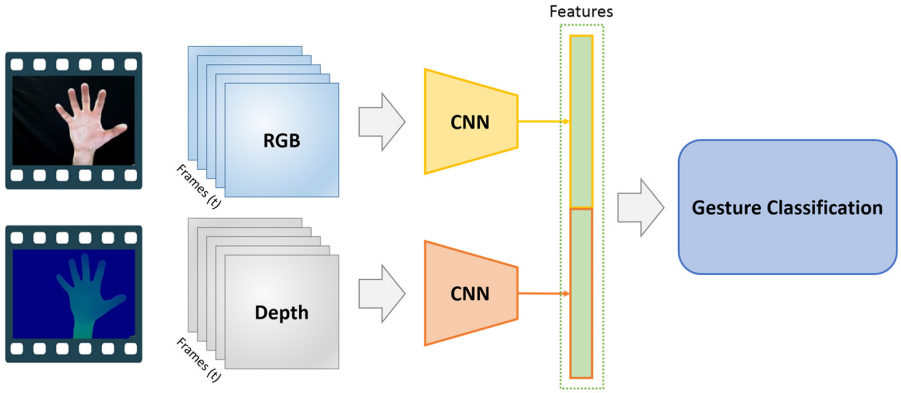
Given the characterized 3D image/video data obtained using techniques such as those described in the previous section, the final step in hand gesture recognition will be the application of an automatic classification technique utilizing several popular hand gesture recognition classification methods. The methods used in hand gesture recognition are summarized in Table 4 with respect to a representative sample, where three main types of approaches have been considered: general-purpose classification

Figure 18



Graphical visualization of 2D convolution (blue), 3D convolution (red), conventional spatial pooling (orange), and temporal pooling (yellow).

Figure 19



Graphical visualization of the process of extracting convolutional neural network (CNN)-based features from RGB-D data for gesture recognition.

methods, end-to-end DL approaches, and techniques designed specifically for the 3D hand gesture recognition problem.

General-purpose classifiers follow the classical pattern recognition approach, where a set of problem-oriented features is extracted in the first stage before a classical classification technique is applied. SVM is a representative example of a classic technique, which has been applied successfully in a wide range of classification problems. SVM has been chosen because it outperforms other classical, and sometimes simpler, algorithms such as kNN in situations where a limited number of training samples is available. SVM has provided impressive gesture recognition results [73], outperforming other classification techniques that, in principle, should be better suited to identifying continuous gestures because they consider the temporal sequence of events. Notable examples include hidden Markov models (HMM) and dynamic time warping (DTW) [5].

DL methods have developed quickly in recent years, demonstrating remarkable success with regard to recognition problems when a significant number of training samples are available, and the 3D hand gesture recognition problem is no exception [77]. Therefore, DL has been chosen as the prototypical end-to-end approach in the literature, and some specific DL architectures have been quite successful at gesture recognition. Hybrid convolutional neural network long short-term memory (CNN-LSTM) ANN architectures are the most relevant for 3D hand gesture recognition [82].

Finally, distortion-invariant correlation filter techniques have been applied for 3D hand gesture recognition under degraded conditions, which include low illumination and occlusion [83]. As an example of how this type of problem-oriented method can compete with classical and end-to-end approaches overall, when there are limited training samples and degraded conditions, this method can exploit an ad-hoc feature extraction and combine it with a distance-based classifier technique.

4.2a. Support Vector Machines

In recent years, use of the SVM classification algorithm has become increasingly widespread [70]. It is a general-purpose supervised classifier with high generalization power, making it an appropriate choice for diverse applications. It performs satisfactorily when the amount of training data is small, making it a useful alternative to the most recent and popular DL algorithms.

The basic SVM is a two-class linear classifier that can be generalized to both nonlinear problems, by means of the "kernel trick," and multi-class problems alike. Let $\{(\boldsymbol{x}_1, y_i : \boldsymbol{x}_i \in \mathbb{R}^d, y_i \in \{-1, 1\}),\ i = 1. \ldots, n\}$ be a set of n hand gesture video sequences. Each video is characterized in a d-dimensional space using a feature vector $\boldsymbol{x}_i$ that is labeled with a gesture class $y_i = 1$ if the video belongs to the target gesture; otherwise, $y_i = -1$. The objective of the SVM is to find a hyperplane in the d-dimensional space in order to separate the training videos of the two classes (Fig. 20, left). From all the possible hyperplanes that may separate the two classes, the objective is to find the hyperplane with the maximum margin between video samples from the two classes (Fig. 20, right).

Table 4. Taxonomy of 3D Image/Video Characterization Methods for Hand Gesture Recognition

Approach	Example
General-purpose classification technique	Support vector machines
Deep learning methods	Artificial neural networks
Correlation filters designed for hand gesture recognition	Distortion-invariant correlation filter

The SVM classifier is a linear classifier that can be adapted to nonlinear separable problems using the so-called "kernel trick" (see Fig. 21). The "kernel trick" involves applying a mapping function from the original d-dimensional space to a higher-dimensional space using a kernel function $\boldsymbol{K}(\boldsymbol{x}_i, \boldsymbol{x}_j)$. The kernel functions used are the linear, polynomial, or radial basis function (RBF) kernels.

Therefore, the objective function that the SVM tries to solve can be expressed as follows:

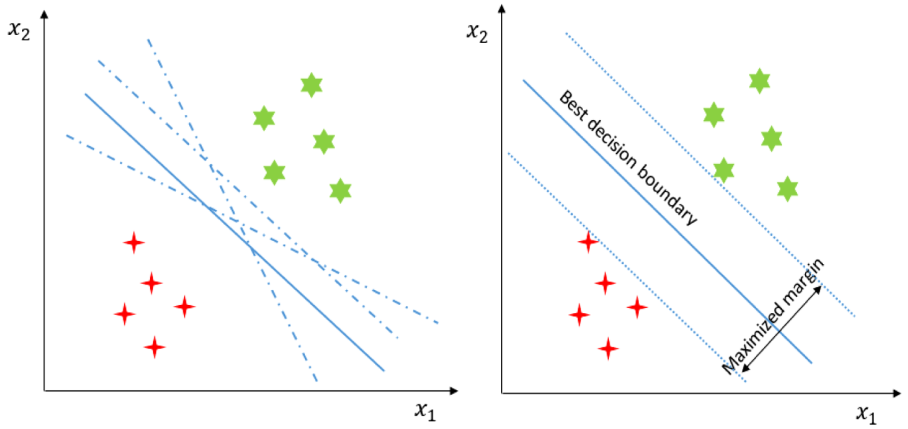
$$\arg\min_{\boldsymbol{w},\beta,b} \frac{1}{2}\ ||\boldsymbol{w}|| + C\sum_{i=1}^{n}\beta_i, \quad \text{subject to } y_i\ (\boldsymbol{w}\cdot\boldsymbol{x}_i - \boldsymbol{b}) \geq = \beta_i, \tag{24}$$

where the vector $\boldsymbol{w} = (w_1, \cdots, w_d)$ corresponds to the weight vector and the constant term representing the hyperplane, and $1/\|\boldsymbol{w}\|$ corresponds to the margin around the decision hyperplane that has to be maximized. $\boldsymbol{\beta} = \{\beta_i\} i = 1, \ldots, n$ refers to the slack variables for each video sample, which are required to handle the classification error, and C is a regularization parameter used to balance the optimization between obtaining a high margin and a low classification error, i.e., between good classifier generalization and fitting, while also obtaining video training samples.

The main learning algorithm employed to find the optimal hyperplane parameters $\boldsymbol{w}$, $\boldsymbol{b}$, and $\boldsymbol{\beta}$ is based on an iterative gradient descent of Eq. (24). The regularization parameter C is estimated by means of an exhaustive validation search in the set $10^{\{-4,-3,\cdots,4\}}$, setting C to the value of the lowest mean classification error using a leaving-on-subject-out validation strategy (see Subsection 5.1).

The SVM objective function [Eq. (24)] is called the *primal* form of the SVM. An alternative formulation of the SVM problem implements the *dual* form SVM based on a Lagrange multiplier formulation, that is,

Figure 20



(Left) Infinite number of potential solutions to a linearly separable problem; (right) the best solution is the one that maximizes the margin (i.e., the distance between the closest samples from the two classes).

$$\begin{aligned} &\arg\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \tfrac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \boldsymbol{K}\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right), \\ &\text{subject to } 0 \le \alpha_i \le C \text{ for all } i = 1, 2, \ldots, n \text{ and } \sum_{i=1}^{n} \alpha_i y_i = 0. \end{aligned} \tag{25}$$

The "kernel trick" refers to the technique employed to generalize the SVM for nonlinear classification problems (Fig. 21). The "kernel trick" involves mapping the original d-dimensional space, where the dot product $(\boldsymbol{x}_i\, \boldsymbol{x}_j)$ of two video sequences is represented, onto a higher-dimensional space using a kernel function $\boldsymbol{K}(\boldsymbol{x}_i, \boldsymbol{x}_j)$. The kernel functions used are the linear, polynomial, or RBF kernels.

Once the SVM parameters $(\boldsymbol{w}, \boldsymbol{b})$ are trained, a new video sample is classified using the decision function $y(\boldsymbol{x}) = \text{sign}(\boldsymbol{w}^T \boldsymbol{x} + \boldsymbol{b})$ by assigning the new gesture video sample $\boldsymbol{x}$ to the target gesture sample if $y(x) > 0$; otherwise, it is not.

In the data pre-processing for the SVM, each feature of the feature vectors $\boldsymbol{x}$ is usually L_1 normalized, with each feature value rescaled within the [0, 1] range to the minimum and maximum values of each feature in the video training dataset. Finally, as a last pre-processing step, the $\boldsymbol{x}$ feature vectors with L_1-normalized features are L_2-normalized.

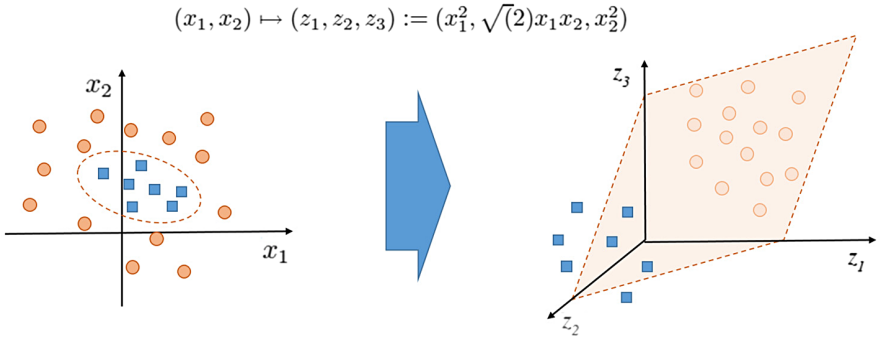
4.2b. Deep Learning: Hybrid Convolutional-Recurrent Neural Network Approach

The most elemental approach to classify human gestures via ANNs is based on the use of several layers of artificial neurons to map the features extracted from the input data onto the final class label space. That is, the uncovered multi-modal features are fed into several final dense layers, enabling human gesture discrimination. However, other more sophisticated designs based on the DL paradigm have demonstrated better performance [76]. In this subsection, we focus on the hybrid CNN-LSTM approach [84] to classify human gestures and the application of this model to 3D imaging, for both integral imaging and RGB-D input data.

Recurrent Neural Networks. The recurrent neural network (RNN) model provides a natural method to exploit the temporal nature of a human gesture classification problem. These networks use an internal feedback loop such that each data sequence depends on the previous sequences. Thus, RNNs can model temporal dynamic behaviors from sequential data. Figure 22 (left) shows a standard RNN cell structure that replicates a linear dynamic system.

The RNN unit takes $\boldsymbol{x}_t$ as the input data and produces an output based on its hidden state $\boldsymbol{h}_t$, which is computed from the current input data, previous output, and hyperbolic tangent layer. For an input sequence of T time steps, $\boldsymbol{x} = [x_1, x_2, x_3, \ldots, x_T]$

Figure 21



Example of applying the "kernel trick" to transform the original (x_1, x_2) space into a transformed (z_1, z_2, z_3) space, where the classes are linearly separable.

is fed into an RNN. Let $\boldsymbol{h} = [h_1, h_2, h_3, \ldots, h_T]$ be the hidden vector and $\boldsymbol{y} = [y_1, y_2, y_3, \ldots, y_T]$ be the output vector. A standard RNN computes $\boldsymbol{h}$ and $\boldsymbol{y}$ using the following relationships [81]:

$$h_t = \sigma(\boldsymbol{W}_{ih} x_t + \boldsymbol{W}_{hh} h_{t-1} + \boldsymbol{b}_h) \tag{26}$$

and

$$y_t = \boldsymbol{W}_{ho} h_t + \boldsymbol{b}_o, \tag{27}$$

for time $t = (1, 2, 3, \ldots, T)$. Here, $\sigma(x) = 1/(1 + e^{-x})$ is the element-wise logistic sigmoid function, $\boldsymbol{W}_{kk}$ for $k \epsilon \{i, f, x, h, o, c\}$ represents the corresponding weight matrices, and $\boldsymbol{b}_k$ for $k \epsilon \{o, c, f, i, h\}$ represents the corresponding bias terms of the network. Despite the potential of this scheme to model data sequences, the vanishing (or exploding) gradient problem makes this straightforward model very difficult to train in practice because of its recursive structure [76].

To address these practical limitations, the long short-term memory (LSTM) network was developed, a variant of RNN exhibiting remarkable performance within the human gesture recognition field. Unlike standard RNNs, the LSTM [85] uses memory cells to identify the long-range temporal relationships hidden in the sequence. It has a self-connection with unity weight. This self-connection causes it to copy its own real-valued state and gather the external signal. In addition, the self-connection is gated multiplicatively by another unit that learns to decide when to clear the content of the memory [75]. A standard LSTM network computes the hidden vector as follows [85]:

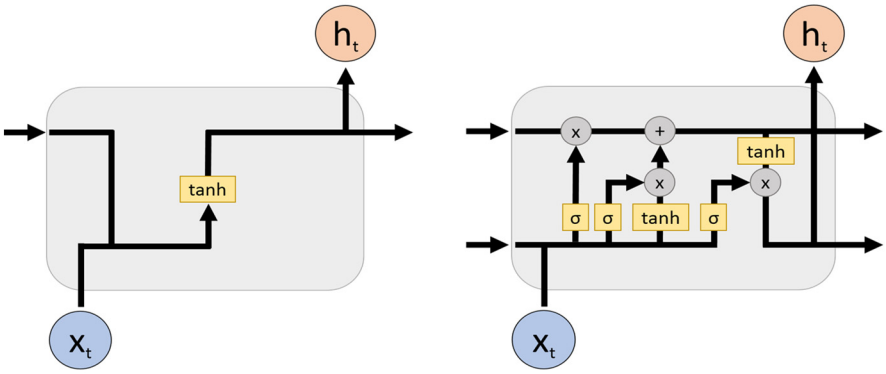
$$i_t = \sigma(\boldsymbol{W}_{xi} x_t + \boldsymbol{W}_{hi} h_{t-1} + \boldsymbol{b}_i), \tag{28}$$

$$f_t = \sigma\left(\boldsymbol{W}_{xf} x_t + \boldsymbol{W}_{hf} h_{t-1} + \boldsymbol{b}_f\right), \tag{29}$$

$$c_t = f_t c_{t-1} + i_t \tan h(\boldsymbol{W}_{xc} x_t + \boldsymbol{W}_{hc} h_{t-1} + \boldsymbol{b}_c), \tag{30}$$

$$o_t = \sigma(\boldsymbol{W}_{xo} x_t + \boldsymbol{W}_{ho} h_{t-1} + \boldsymbol{b}_o), \tag{31}$$

Figure 22



RNN (left) and LSTM (right) cell structures. The RNN replicates a linear dynamic system, and the LSTM includes additional control gates.

$$h_t = o_t \tanh(c_t)\,, \tag{32}$$

where i, f, and o are the input, forget, and output gates, respectively, and c represents the cell state vectors.

Figure 22 (right) shows the LSTM cell structure including three different control gates. From left to right, the first one (forget gate) decides the amount of input information that remains in the cell by passing the input data ($\boldsymbol{x}_t$) and the previous state through a sigmoid activation function. The second one (input gate) updates the cell content as follows: initially, a sigmoid function is applied to both the previous state and $\boldsymbol{x}_t$ to set the values that will be updated. Next, the sigmoid output is multiplied by the output of a hyperbolic tangent layer. At this point, the cell state is updated by multiplying the previous state with the output of the forget gate and adding it to the output of the input gate. The final step (output gate) determines the system output by passing the previous state and the input data $\boldsymbol{x}_t$ through a sigmoid function. This result is multiplied by the current cell state and passed to the hyperbolic tangent layer to produce the final output result.

The literature on LSTM technologies presents different variations for human gesture recognition, e.g., the bi-directional long short-term memory (BiLSTM) network, which makes use of two LSTM networks, the first one for learning in the positive time direction (forward states) $\boldsymbol{h}_{\text{forward}}$ and the other for learning in the negative time direction (backward states)$\boldsymbol{h}_{\text{reverse}}$. The output layers are merged for classification, i.e., $\mathbf{h} = [\boldsymbol{h}_{\text{forward}}, \boldsymbol{h}_{\text{reverse}}]$, where the resulting output is typically normalized via a fully connected layer and the softmax nonlinear activation as follows:

$$\boldsymbol{y} = \text{sm}(\boldsymbol{W}_h \boldsymbol{h} + \boldsymbol{b}_h)\,, \tag{33}$$

where sm represents the softmaxfunction. Notably, as in the case of CNN, the LSTM-based methods normally utilize several fully connected layers to discriminate the different gestures. Most recent studies have shown that hybrid CNN-LSTM models give better results. In the following subsection, we review a representative hybrid approach and provide more details on this type of architecture.

Spatio-temporal human gesture recognition using 3D imaging and deep neural networks. In this subsection, we discuss the technical details of dynamic gesture recognition using 3D imaging and deep neural networks. In particular, we discuss the DL architectures used for gesture recognition via two different 3D imaging techniques; the first architecture considers integral imaging-based data acquisition, while the second considers RGB-D sensor-based data acquisition. In both cases, we can use a hybrid CNN-LSTM approach. The CNN is useful for extracting the spatial features, while the LSTM network assists in capturing the temporal dependency of feature vectors. Integral imaging can be considered as a depth-based filtering process, which outputs the reconstructed video at the depth of the gesture of interest. Unlike integral imaging, RGB-D sensors such as Kinect provide the RGB and the depth videos separately, which requires a multi-modal fusion for classification. The block diagrams and their descriptions for deep neural network-based gesture recognition using integral imaging and RGB-D sensors have been provided in detail in the following subsections.

Hybrid CNN-LSTM for integral imaging-based data acquisition. This subsection describes a representative approach using CNN and LSTM for integral imaging-based data acquisition. CNNs are a popular method used for object recognition and image classification [75]. As the name suggests, conventional CNNs consist of stacked convolutional and pooling layers followed by one or more fully connected layers.

The convolutional layers use a series of convolutional kernels for feature learning. The output of the convolutional layers is a set of feature vectors, which represents the spatial features in an image. The pooling layer combines semantically similar features into a single feature [75]. In the case of gesture recognition, the temporal dependencies of feature vectors from adjacent frames are also important, which a CNN alone cannot capture. Therefore, a cascaded network is used, as shown in Fig. 23. The convolutional layers of the CNN produce the feature vector, which is fed as an input into a BiLSTM network, which captures the temporal dependency of the adjacent frames in the gesture videos. The following subsections discuss the proposed system in detail. Figure 23 shows the block diagram of the gesture recognition strategy using integral imaging-based data acquisition.

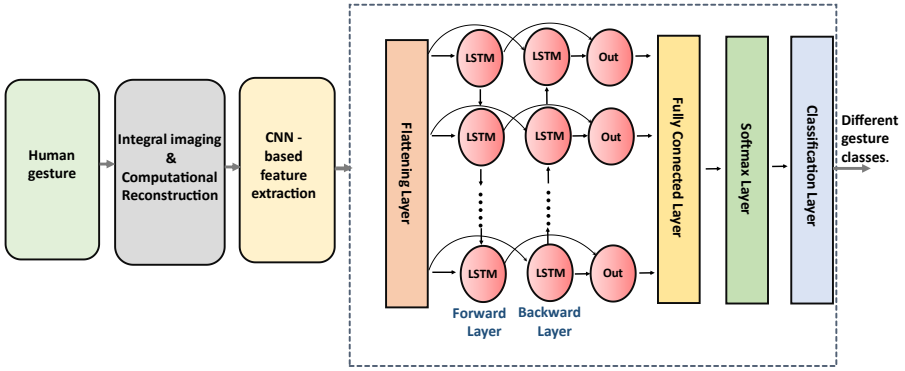
The computational reconstruction algorithm is an inverse mapping procedure, which extracts pixels from each elemental image and displays the corresponding voxels at coordinates [86]. In this tutorial, we considered the spatio-temporal volume video data for the reconstruction process:

$$r(x, y, z, t) = \frac{1}{O(x, y, t)} \sum_{i=0}^{K-1} \sum_{j=0}^{L-1} EI_{i,j}\left(x - i\frac{r_x \times p_x}{M \times d_x}, y - j\frac{r_y \times p_y}{M \times d_y}, t\right). \quad \textbf{(34)}$$

Here, $r(x, y, z, t)$ is the integral imaging-reconstructed video, x and y represent the pixel indices in each frame, z represents the reconstruction depth, and t is the frame index. The reconstructed frame at depth z is obtained by shifting and overlapping $K \times L$ elemental images $EI_{i,j}$ at a specific depth z. In Eq. (34), r_x and r_y represent the resolutions, d_x and d_y represent the physical size of the image sensor, and p_x and p_y indicate the pitches of adjacent image sensors on the camera array. The $O(x, y, t)$ matrix contains information regarding the number of overlapping pixels and the magnification factor $M = z/f$, where f represents the focal length.

The 3D reconstructed video obtained using Eq. (34) is fed as the input data into a CNN, in order to extract its spatial features. The input data are passed through a series of convolutional and pooling layers, which spatially filter the input to extract relevant features. The output of the last pooling layer is taken as the feature vector representing the spatial information of the video, which can be used for high-level interpretation tasks. For small datasets, we can use a pretrained network such as a deep GoogLeNet network, pretrained on the well-known ImageNet [87] database.

Figure 23



Block diagram of the deep learning strategy for 3D integral imaging-based gesture recognition. CNN, convolutional neural network; LSTM, long short-term memory network. (Reprinted from [108].)

Similar to the architecture proposed by Serre *et al*. [88], which used a series of Gabor filters to handle multiple scales, GoogLeNet uses an inception module. The filters in the inception module are learned [89]. GoogLeNet takes a single 224×224 image as its input and passes it through multiple filters in parallel, using 1×1, 3×3, and 5×5 convolution and max-pooling operations, and concatenates the resulting filter outputs [89]. Finally, the activations are average-pooled to obtain a feature vector $\boldsymbol{x}$. For a video with N frames, $\{I_1, I_2, I_3, \ldots, I_N\}$, where I represents the frame and the index represents the frame number. For each frame I_n, $n = (1, 2, \ldots, N)$, the CNN produces a P-dimensional feature vector $\boldsymbol{X} \epsilon \mathbb{R}^P$. Thus, an N frame video can be represented by a feature matrix $\boldsymbol{X} \epsilon \mathbb{R}^{P \times N}$, i.e.,

$$\boldsymbol{X} = \boldsymbol{x}_1 \odot \boldsymbol{x}_2 \odot \boldsymbol{x}_3 \odot \ldots \odot \boldsymbol{x}_N, \tag{35}$$

where $\odot$ represents the concatenation operator. Therefore, for each input video, the convolutional and pooling layers of the CNN output the feature matrix $\boldsymbol{X} \epsilon \mathbb{R}^{P \times N}$, where P represents the number of features. The row and column sequences of $\boldsymbol{X} \epsilon \mathbb{R}^{P \times N}$ encode the spatial and temporal information of the video, respectively.

The CNN is followed by a BiLSTM network, which learns the temporal dependency between the feature vectors extracted from the adjacent frames of the input video. Finally, it is fed to a fully connected layer and classification layer for gesture classification.

Hybrid CNN-LSTM for RGB-D sensor-based data acquisition. In this subsection, we present specific details of a representative approach, namely the hybrid CNN-LSTM model, for human gesture recognition using RGB-D data. We focus on the work presented in [84], which provides a multi-modal human gesture recognition approach based on 3D CNN and LSTM. The rationale behind this method is based on learning short-term spatio-temporal gesture features first using a CNN before learning long-term features via convolutional LSTM. Figure 24 shows an overview of the five steps included in this method.

In the pre-processing step [Fig. 24(a)], the RGB and depth modalities are initially normalized and down-sampled to transform each gesture sequence into a fixed length L. Specifically, a temporal jitter strategy is used to sample a gesture sequence with S frames as follows:

$$I_{X_i} = \frac{S}{L} \times \left(i + \frac{r}{2}\right), \tag{36}$$

where I_{X_i} represents the index of the ith frame, and r is a random value between -1 and 1, sampled from the uniform distribution. The second step [Fig. 24(b)] processes each data modality (i.e., RGB and depth) to extract the corresponding features using the 3D CNN model defined in [90].

This network design includes four Conv3D layers with 64, 128, 256, and 256 kernels as well as batch normalization, ReLU activation, and spatio-temporal pooling. The third step [Fig. 24(c)] consists of a convolutional LSTM, which, unlike regular LSTMs, utilizes convolutional structures in both the input-to-state and state-to-state transitions. The gates of the ConvLSTM can be described using Eqs. (28)–(33).

The fourth step [Fig. 24(d)] applies a spatial pyramid pooling layer between the ConvLSTM and the fully connected layers to reduce the data dimensionality and, consequently, the number of model parameters. This process pools the generated feature maps at four different levels to generate feature representations with a fixed size of $49 + 16 + 4 + 1 = 70$. Finally, the multi-modal fusion step [Fig. 24(e)] adopts a late

fusion approach by averaging the predictions generated by the network modalities to produce a final gesture prediction for the input RGB-D data.

4.2c. *Correlation-Based Spatio-Temporal Human Gesture Recognition in Degraded Environments*

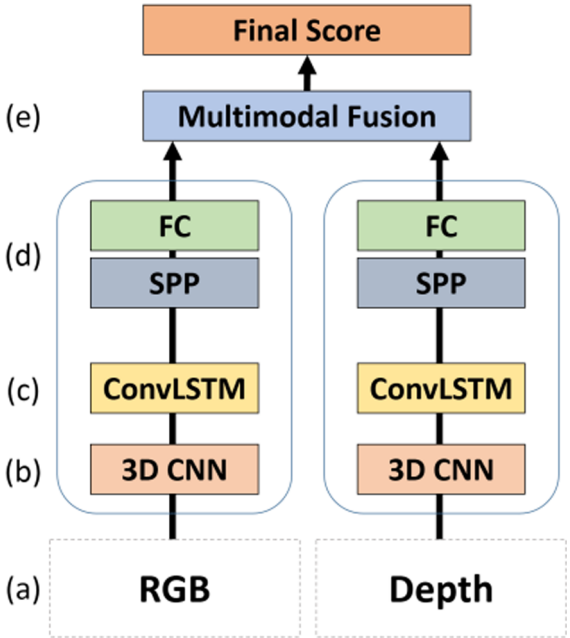
The methodologies and strategies discussed thus far require the acquisition of accurate human gesture features from the scene. However, under some degraded imaging conditions, it can be challenging to achieve the desired quality. Correlation-based matched filters have been investigated and used for target detection and object recognition [91,92] because these approaches do not require prior object detection or segmentation, and they are important for solving pattern recognition problems.

The concept of a conventional matched filter is equivalent to a linear space-invariant filter. Consider a 3D signal $s(x, y, t)$; the impulse response of the filter $h(x, y, t)$ is designed to correspond to the model signal as a template [93]:

$$h(x, y, t) = s^*(-x, -y, -t), \tag{37}$$

where $s*(\cdot)$ represents the complex conjugate of the signal. The correlation process applies the designed filter template to the unknown test data $t(\cdot)$ at every spatio-temporal location. The correlation output $g(\cdot)$ reaches a large value at the location where the test data is similar to the template. The cross-correlation process between the test signal $[t(\cdot)]$ and the template $[s(\cdot)]$ can be expressed in the spatio-temporal domain as

Figure 24



LSTM-based approach, presented in [84], composed of five blocks: (a) pre-processing, (b) 3D CNN, (c) convolutional LSTM, (d) spatial pyramid pooling, and (e) multi-modal fusion.

$$g(x, y, t) = \iiint_{-\infty}^{\infty} t(\xi, \eta, \gamma) h(x - \xi, y - \eta, t - \gamma) d\xi d\eta d\gamma$$
$$= \iiint_{-\infty}^{\infty} t(\xi, \eta, \gamma) s^*(\xi - x, \eta - y, \gamma - t) d\xi d\eta d\gamma. \quad (38)$$

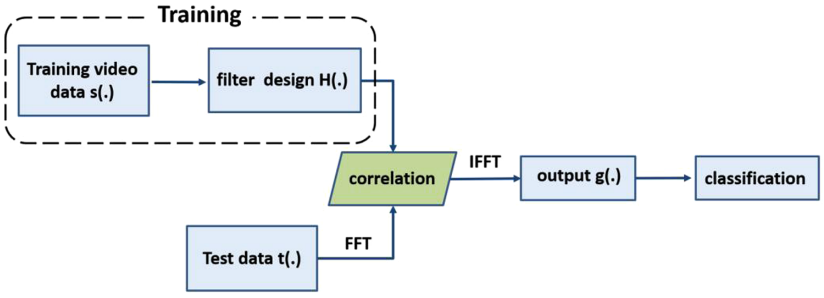
For correlation-based human gesture recognition, the correlation filters will be designed based on 3D $(x - y - t)$ or 4D $(x - y - z - t)$ video sequence templates. To perform high-quality correlation between the templates and the test video sequence, care must be taken when designing the filter to minimize the difference between the correlation response of the filter and the desired correlation output. Once the filter is synthesized, it can be correlated with the test data in the frequency domain:

$$g(\cdot) = \mathrm{FT}^{-1}\left\{T\left(f_x, f_y, f_t\right) \times S^*\left(f_x, f_y, f_t\right)\right\}, \quad (39)$$

where $T(f_x, f_y, f_t) = \mathrm{FT}[t(x, y, t)]$, $S(f_x, f_y, f_t) = \mathrm{FT}[s(x, y, t)]$, and $\mathrm{FT}[\cdot]$ is the Fourier transform operation. The correlation output is obtained by applying an inverse Fourier transform operation $\mathrm{FT}^{-1}[\cdot]$. As shown in Fig. 25, the correlation-based object recognition concept comprises three main steps: (1) training and design of a correlation filter, (2) frequency domain correlation, and (3) classification (for human gesture recognition).

Compared with the pattern recognition techniques discussed in previous sections, correlation-based approaches for object recognition are computationally efficient, spatially shift-invariant, and can provide a large peak indicating the presence and location of the desired target template. Correlation-based approaches are easy to understand and implement; they have advantages when limited samples are available for training, and they do not require the computation of features. In addition, correlation is robust to certain types of noise. However, the drawbacks of correlation-based approaches may include the fact that consistent similarity between the target and the template is needed for high-quality performance. Correlation filters are particularly sensitive to rotations and scale size changes because they are both amplitude and phase matched [93–95]. For more complex objects and scenes, the sharpness of the correlation output may be reduced, and correlation-based approaches may not be able to solve classification problems with very complex decision boundaries.

Figure 25



Flowchart of correlation-based object recognition. FFT, fast Fourier transform; IFFT, inverse fast Fourier transform. (Reprinted from [83].)

In the end, classification is performed to recognize the object of interest based on specific features. The correlation-based methods described in this subsection can apply the nearest neighbor (NN) type classifier, which uses the correlation output as a dissimilarity/distance measurement. This is performed using the correlation in relation to an "exemplar"-type video, which acts as the training set for the NN classifier. The NN classifier expects the class conditional probabilities to be constant at a localized scale, and classifies each piece of test data by labeling it with respect to multiple nearest neighbors, which are defined in the training dataset. The NN classifier will first calculate the distance between the object and the model points in each class in the feature domain. It then classifies the object to a class with the shortest distance, which can be either the Euclidian distance or another weight combined distance [96]. The NN classifier is a simple method for pattern recognition. However, if the features of the object are not distributed in the domain in a straightforward way, overlaps between different classes may occur, and the performance of the NN classifier may be reduced. In this situation, we may consider the Bayesian classifier, which uses the frequency information of the object and the probabilistic information of its features.

Conventional correlation-based matched filters used for target detection were designed to optimize the output SNR in the presence of additive overlapping stationary noise. However, such filters are sensitive to distortion and changes in the target, which may lead to poor correlation performance for object recognition under degraded conditions such as when the objects are partially occluded and/or there is low illumination. To solve this issue, optimum distortion-invariant filters have been proposed [97,98]. In Subsection 3.6, we explained the advantages of integral imaging-based 3D imaging in a degraded environment. By combining these concepts, we can use integral imaging to apply the spatio-temporal correlation method for human gesture recognition, especially under degraded conditions. The details of the distortion-invariant filter design and the classification process will be discussed in Appendix A. In Subsection 5.2, we will show the corresponding experimental results to illustrate the potential of the spatio-temporal correlation-based approach with 3D integral imaging and TV denoising algorithms for efficient human gesture recognition in degraded environments. Comparative analysis against other well-known algorithms further indicates that the nonlinear correlation filtering approach is more robust for human gesture recognition under degraded conditions.

The correlation-based method described in Subsection 4.2c applies a nearest neighbor (NN) type classifier, which uses correlation as a dissimilarity/distance measure. This is carried out by means of the correlation in relation to an "exemplar"-type video, which acts as the training set for the NN classifier.

4.3. Performance Metrics of Human Gesture Recognition Systems

Thus far, we have described how different technologies can be used to acquire and process human gestures. The final processing step concerns the quantification of their characterization performance. Human activity recognition is a complex classification problem, which can be tackled, in terms of performance, from vastly different viewpoints. Depending on whether the activity is continuous or not, we might have to consider the classification quality from a timestamp viewpoint (e.g., fragmentation of the video, or when the action starts and finishes). This affects how input data is extracted before it is classified.

In this regard, research addressing the problem of failures in the data capture of the action, specifically in video event segmentation, event merging, and timing offsets, has been conducted previously [99,100]. This raises the possibility of inserting, deleting, fragmenting, merging, or combining fragmented and merged operations of the information provided by the sensors. This is especially useful in cases where the nature of the sensors is not visual, e.g., in WiFi [13–16], or when there is an additional complexity of visual scenarios, such as in the behavior of a crowd of people, which makes it difficult to temporally segment the actions or gestures. Nevertheless, this specific problem is beyond the scope of the proposed tutorial, which is focused on the main techniques and methods for 3D hand gesture characterization and recognition.

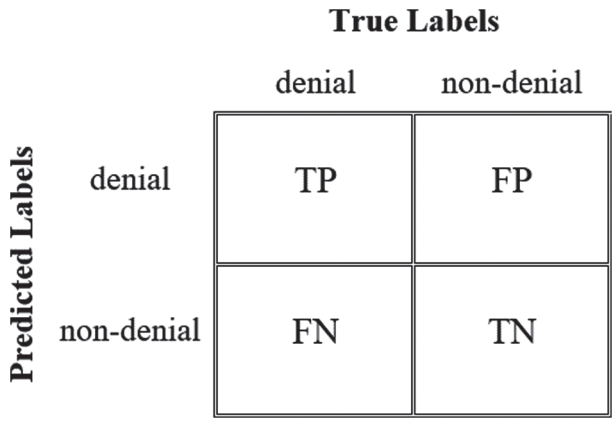
Once the trajectory of the gesture is well characterized in the temporal domain, we can focus on the meaning of the gesture within that video. The gesture recognition process can be described by building a feature space in which each gesture can be represented. In this case, the problem may be solved using a conventional classifier approach such as the SVM. This classifier performs well in situations where the number of data samples is scarce and dispersed in spaces of high dimensionality, as in problems related to gesture recognition.

In the case of a binary classification, such as the recognition of a certain gesture, the denial gesture might be labeled as "denial" or "positive" and "non-denial" or "negative." The terms "positive" and "negative" are related to the presence or absence of a certain event, in this case the gesture "denial." Therefore, in binary classification, the confusion matrix is limited to a 2×2 matrix between the classifier prediction and the real value for each sample. Four possible results may be obtained: true positive (TP), which represents the number of samples where both the prediction and true labels coincide with the gesture "denial"; false positive (FP) when its true label is "non-denial" but its prediction is "denial"; false negative (FN) when the label is "denial" but the prediction is "non-denial"; and true negative (TN) when both the prediction and true labels coincide with "non-denial" (see Fig. 26).

In this type of problem, the measures used most frequently to quantify the classification results are *precision* and *sensibility* or $recall$: $precision = \mathrm{TP}/(\mathrm{TP} + \mathrm{FP})$ and $\text{sensibility} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FN})$. These two measures can be combined with the F1 score measure via the harmonic mean between them:

$$F1_{\text{score}} = 2 \cdot \frac{\text{precision} \cdot \text{sensibility}}{\text{precision} + \text{sensibility}}. \tag{40}$$

Figure 26



Confusion matrix in a binary classification problem.

Other measures include the *accuracy* and *Matthews's correlation coefficient* (MCC) [101]:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \tag{41}$$

and

$$\text{MCC} = \frac{\text{TP}\cdot\text{TN} - \text{FP}\cdot\text{FN}}{\sqrt{(\text{TP} + \text{FP}) \cdot (\text{TP} + \text{FN}) \cdot (\text{TN} + \text{FP}) \,.\, (\text{TN} + \text{FN})}}. \tag{42}$$

We can analyze the performance of the binary classifier through the receiver operating characteristic (ROC) curve. This curve represents the relationship between the true positive rate (*sensibility*) on the ordinate axis and the false positive rate, FPR = 1–*specificity*, on the abscissa axis, where *specificity* = TN/(TN + FP). Each prediction or instance of the confusion matrix represents a point in the ROC space. The classification is perfect at the upper left corner where we have a 100% specificity (no false negative) and 100% specificity (no false positive).

Any point along the diagonal line (red line) gives a random classification, also called the line of no-discrimination. The diagonal ROC divides the space into two regions: classification results that represent points above the diagonal (better classification than random), and those representing the points below the line (poorer classification than random). Figure 27 shows the prediction results, indicated in blue, for the case of the classifier with better behavior than the random classifier. The commonly used statistic is the area under the ROC curve (Fig. 27), typically referred to as the AUC (area under the curve). This index can be interpreted as the probability that a classifier scores a positive instance, chosen at random, is higher than a negative instance [102].

In the general case of multiclass classification, different strategies have been proposed to evaluate the performance. In the binary classification case, there is a confusion matrix whose range coincides with the number of classes k, as shown in Table 5. The first evaluation scheme is to generalize the different criterion-defined binary classifiers from the multiclass confusion matrix, obtaining the following expressions for the class n:

$$\begin{aligned} &\text{TP}_{(n)} = C_{n,n}, \;\; \text{FN}_{(n)} = \text{NP}_{l,n} - C_{n,n}\,, \;\; \text{FP}_{(n)} = \text{NP}_{n,l} - C_{n,n}\,, \\ &\text{TN}_{(n)} = \sum_{l=1}^{k}\sum_{j=1}^{k} C_{l,j} - \text{TP}_{(n)} - \text{FN}_{(n)} - \text{FP}_{(n)}. \end{aligned} \tag{43}$$

Similarly, we can obtain the rest of the expressions for class n as follows:

$$\begin{aligned} &\text{accuracy}_{(n)} = \frac{\text{TP}_{(n)}}{\text{TP}_{(n)} + \text{FP}_{(n)} + \text{FN}_{(n)} + \text{FN}_{(n)}}, \; \text{precision}_{(n)} = \frac{\text{TP}_{(n)}}{\text{TP}_{(n)} + \text{FP}_{(n)}}, \\ &\text{sensibilit}y_{(n)} = \frac{\text{TP}_{(n)}}{\text{TP}_{(n)} + \text{FN}_{(n)}}, \;\; F1_{\text{score}(n)} = 2 \cdot \frac{\text{precision}_{(\text{n})} \cdot \text{sensibility}_{(n)}}{\text{precision}_{(n)} + \text{sensibility}_{(n)}}, \\ &\text{MCC}_{(n)} = \frac{\text{TP}_{(n)} \cdot \text{TN}_{(n)} - \text{FP}_{(n)} \cdot \text{FN}_{(n)}}{\sqrt{\left(\text{TP}_{(n)} + \text{FP}_{(n)}\right) \cdot \left(\text{TP}_{(n)} + \text{FN}_{(n)}\right) \cdot \left(\text{TN}_{(n)} + \text{FP}_{(n)}\right) \cdot \left(\text{TN}_{(n)} + \text{FN}_{(n)}\right)}}. \end{aligned} \tag{44}$$

This allows the use of these measures to divide a multiclass problem into several class problems and later take the average of the obtained measures. In the case of the *accuracy* measure, this is given by the sum of the accuracies of each class,

$$\text{accuracy} = \sum_{l=1}^{k} \text{accuracy}_{(l)}, \quad F1_{\text{score}} = \frac{1}{k}\sum_{l=1}^{k} F1_{\text{score}\,(l)}, \quad \text{MCC} = \frac{1}{k}\sum_{l=1}^{k} \text{MCC}_{(l)}. \tag{45}$$

Another typical measure is *Cohen's kappa* measure [103]. It was developed in the field of psychology to establish a statistical significance test of the diagnosis by different experts. This idea has been used in the multiclass confusion matrix for classification problems. Let us consider Table 5, where the confusion matrix contains a total of n samples. The marginal accumulated samples are shown in the last row and column. Let us denote the sum of samples in the main diagonal using P_a and the product of the marginal accumulated using P_e such as

$$P_a = \sum_{l=1}^{k} C_{l,l} \tag{46}$$

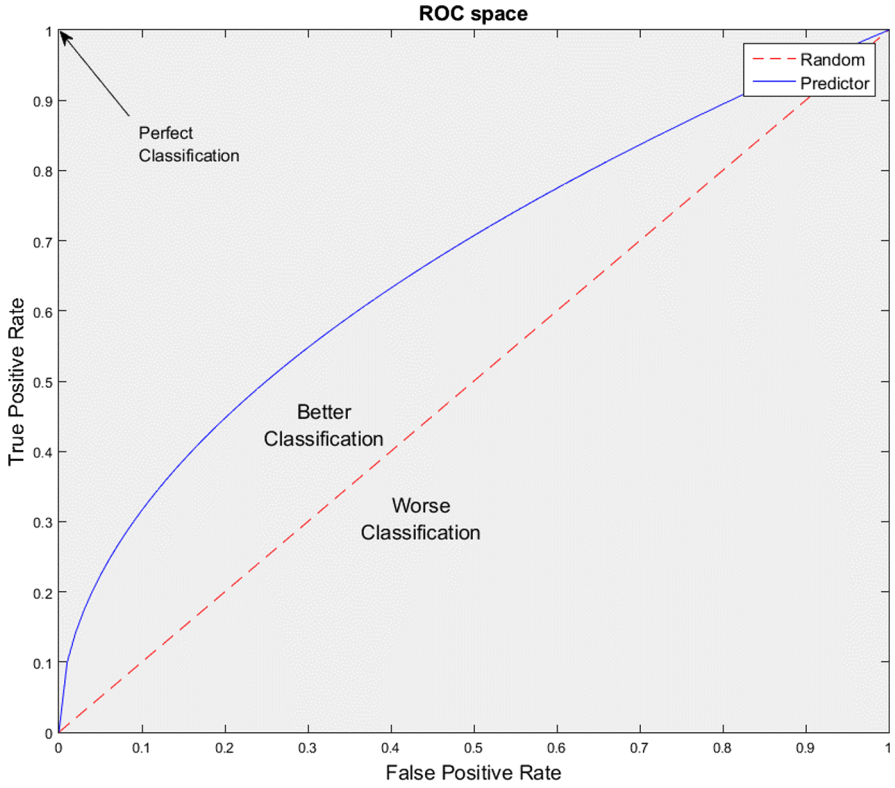
and

$$P_e = NP_{1,l} \cdot NP_{l,1} + NP_{2,l} \cdot NP_{l,2} + \ldots + NP_{k,l} \cdot NP_{l,k}. \tag{47}$$

Thus, we define the kappa coefficient κ as

$$P_e = NP_{1,l} \cdot P_{l,1} + NP_{2,l} \cdot P_{l,2} + \cdots + NP_{k,l} \cdot P_{l,k}. \tag{48}$$

Figure 27



ROC space with random results and with a possible classifier.

When the true labels match the predictions of the classifier, we obtain a value of $\kappa = 1$, whereas if there is no similarity, the true labels and predictions take the value $\kappa = 0$.

5. EXPERIMENTS ILLUSTRATING HUMAN GESTURE RECOGNITION

This section shows three examples of the application of integral imaging for human gesture recognition. In the first example (Subsection 5.1), integral imaging is compared with Kinect in terms of the gesture recognition performance when there is an occlusion that may (totally or partially) impede the correct visualization of the gesture. In the second example (Subsection 5.2), another series of real experiments is performed using the correlation-based approach explained in Subsection 4.2c., when there are degraded environmental conditions (in particular, with partial occlusions and under low illumination). The experimental results shown in these sections verify that integral imaging can enhance the performance of human gesture recognition in degraded environments. In the third example, a deep learning approach is used. A cascaded network formed by a CNN, fed into a BiLSTM network, is used for classifying two different types of gestures under different acquisition conditions.

5.1. Experimental Results of Gesture Recognition under Occlusions

In this subsection, we compare two 3D imaging methodologies by conducting real experiments [8]: (1) elemental images are obtained using integral imaging from a 3×3 camera array and (2) RGB-D data are generated by a Kinect. Eleven subjects performed three different gestures, repeating them twice in front of the camera array, once in an unobstructed scenario, and once obstructed by a plant. The hand gestures used for recognition were "open," "left," and "deny." In addition, a Kinect was placed under the camera setup so that the gestures were recorded by the devices at the same time (see Fig. 28).

The two systems have different image resolutions. The Kinect's images have a larger FOV but a lower resolution (640×480 pixels). The images from integral imaging are 1024×768 pixels in size. To make an equivalence between the two resolutions, cropping and resizing can be performed to obtain a common interest region (subjects' upper bodies). In Fig. 29, we show the image representation under RGB-D, monocular, and integral imaging for both the non-occluded and occluded views. In the case of the monocular image, it corresponds to the central camera image of the setup.

Figures 29(a) and 29(b) show the resulting RGB-D information obtained by the Kinect, whereas Figs. 29(c) and 29(d) show the corresponding information acquired by integral imaging. For both the sensors, the same scene of the person is shown, in which we can see a hand gesture with and without occlusions. We can observe that in the scenes with occlusion, the Kinect generates a significant amount of noise that can

Table 5. Confusion Matrix and Marginal Accumulated Values

		True Labels				
		Class 1	*Class 2*	$\cdots$	*Class k*	
Predicted labels	*Class 1*	$C_{1,1}$	$C_{1,2}$	$\cdots$	$C_{1,k}$	$NP_{1,l} = \sum_{l=1}^{k} C_{1,l}$
	Class 2	$C_{2,1}$	$C_{2,2}$			$\vdots$
	$\vdots$	$\vdots$		$\ddots$	$\ddots$	
	Classk	$C_{k,1}$	$C_{k,2}$		$C_{k,k}$	$NP_{k,l} = \sum_{l=1}^{k} C_{k,l}$
		$NP_{l,1} = \sum_{l=1}^{k} C_{l,1}$	$\cdots$		$NP_{l,k} = \sum_{l=1}^{k} C_{l,k}$	

be solved by integral imaging if it is focused at an appropriate depth with respect to the gesture.

As for obtaining features from STIPs, different local visual descriptors can be used: the HOG, the HOF, and their concatenation (HOG + HOF). In this experiment, only the performance with HOG + HOF is reported. Histograms were L_1-normalized, and the individual features were rescaled independently to the range $[\{0,1\}]$. Finally, the histograms were L_2-normalized. Different vocabulary sizes $K \in \{10, 25, 50, 100, 200, 500, 1000, 2000\}$ were tested using the K-means algorithm of the *VLFeat library* [104].

When classifying hand gestures, a SVM [70] was tested with two kernel models, namely, a linear and a nonlinear model, with the RBF leading to similar performance in each case; therefore, only the learning outcomes related to the linear SVM

Figure 28

Two-sensor setup used in the experiments: 3×3 camera array on top and Kinect device below. (Reprinted from [8].)

Figure 29

(a) (b) (c) (d)

(a) and (b) RGB-D data without and with occlusion, respectively. On the top, we show the color images obtained by the Kinect, and on the bottom their corresponding depth maps. (c) and (d) On the top, we show the monocular images obtained using the central camera without and with occlusion. On the bottom, we show the 3D reconstructed images obtained by integral imaging in the hand-depth plane. (Reprinted from [8].)

are reported. For the implementation, the *LIBSVM library* [105] was used. As for the parameters, we need only to adjust the parameter C between a set of values $\{10^e : e \in \{-4, -3, \cdots, 4\}\}$ via cross-validation. Regarding the process of data splitting, a "leave a subject out" protocol was used. Moreover, given the variability in the results of the K-means algorithm, the entire process (clustering + learning + classification) was repeated $n = 10$ times and the average accuracy reported. The performance plot (see Fig. 30) includes these averages and their standard errors as a measure of the variance.

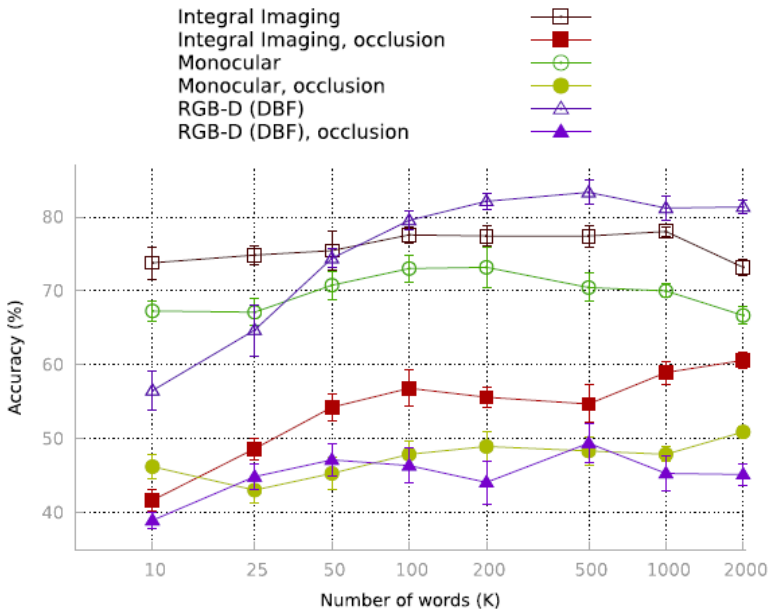
In the experiments with occlusion, the training set was constructed using the STIPs from videos of the non-occluded gestures in order to have "clean" gestures, while the test set included the STIPs from videos with occlusions. The process of partial occlusion of the gesture throughout the video brings a level of unpredictability, while including it in the training process makes learning difficult.

To study the effectiveness of depth-based filtering (DBF), we compare the performance of using the entire set of STIPs detected in the RGB Kinect's images with the filtered set of STIPs resulting from the DBF.

Figure 30 shows that a sufficiently large vocabulary, e.g., $K > 500$, is required to have an adequate learning rate. The learning rate is two percentage points higher with DBF than with RGB-D for a vocabulary of $K = 1000$ words. This advantage is maintained even in the case of larger vocabularies ($K \in \{3000, \ldots, 7000\}$). However, given the small number of samples available, as we increase the dimensionality of the space, the data are more dispersed. To demonstrate this tendency, we should perform the experiments with larger gesture databases.

The following observations can be made by comparing the three sensory modalities with and without occlusion (see Fig. 30):

Figure 30



Comparison of monocular, RGB-D, and integral imaging with and without occlusion. The horizontal axis represents the number of words in the logarithmic scale to emphasize the wide range of vocabulary sizes. (Reprinted from [8].)

1. *Occlusion:* There is a significant difference (approximately 20%) in the learning rate between the results with occlusion and without occlusions for the three proposals (integral imaging, monocular, and RGB-D).
2. *Integral imaging versus monocular:* Integral imaging obtains better results under occlusion than the monocular images. The use of an image in focus around the depth of the gesture integrates the camera-array information, allowing the elimination of some of the information occluded in the hand gesture. This difference increases approximately with vocabulary size.
3. *Integral imaging versus RGB-D*: In the case of no occlusion, integral imaging obtains better results than RGB-D for small vocabularies; RGB-D (through the DBF mechanism) outperforms integral imaging for larger vocabularies. However, although the DBF has some positive effect without occlusions, it does not find a solution in scenes with occlusions, where integral imaging is clearly a better option. To understand the reasons behind the difference in performance between integral imaging and RGB-D, note that in RGB-D, DBF eliminates some potentially noisy STIPs detected in monocular images (RGB), whereas the STIPs detected from the integral image are different from the monocular case.
4. *RGB-D versus monocular*: In the case of no occlusion, RGB-D outperforms the monocular images as a result of its learning capability; however, in scenes with DBF occlusion, it is inferior to the monocular case because the DBF procedure can eliminate unnecessary STIPs.

Integral imaging implemented via obtaining a video of images in focus naturally solves a part of the occlusion problem owing to the multi-view nature of the method. Thus, with just a few words (only 10), acceptable performance may be obtained. Experimentally, we found that $K \approx 10$ seems to be the minimum required number of words.

Table 6 lists the changes in the precision for situations with and without occlusions, comparing monocular images with images obtained through integral imaging. In the case of no occlusions, a higher resolution in the number of words is similarly beneficial to the two methods, with an average accuracy increase of 5%. Increasingly large vocabularies are required to obtain constant performance, possibly because more STIPs are found. In the case of occlusions, learning using monocular images only occurs from 100 words onwards. Therefore, we can rely on integral imaging owing to its ability to focus as well as the quality of the resolution provided.

Regarding the ability to recognize each of the three hand gestures, multi-class criteria have been applied to each of them. Table 7 shows the confusion matrix and the marginal accumulated values, in the case of recognition by means of integral

Table 6. Change in Average Accuracy (%) with Respect to the Low-Resolution Case, in Monocular (Mono) and Integral Imaging (II) (* = with Occlusion)

K	10	25	50	100	200	500	1000	2000
Mono	−20.8	0.0	+2.7	+2.3	+4.4	+7.0	+6.8	+6.5
II	−12.9	−1.8	+6.2	+3.6	+4.7	+5.6	+5.2	+7.1
*Mono**	−12.6	−5.5	−4.2	−6.2	−3.9	+2.8	+9.7	+6.3
*II**	+3.3	+3.0	+0.3	−0.8	+4.1	+10.0	+2.4	−0.5

Table 7. Confusion Matrix and Marginal Accumulated Values

		True Labels			
		Open	Left	Deny	
	Open	17	1	1	$NP_{1,l} = 19$
Predicted	*Left*	1	14	0	$NP_{2,l} = 15$
Labels	*Deny*	4	7	21	$NP_{3,l} = 32$
		$NP_{l,1} = 22$	$NP_{l,2} = 22$	$NP_{l,3} = 22$	

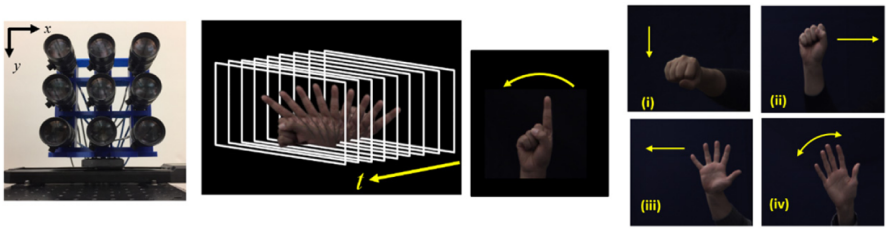
imaging without occlusions, and with a BoW of 1000 words. In the case of *sensitivity*, the aim is to reduce false negatives (FNs), whereas in the case of specificity, the objective is to reduce false positives (FPs). When analyzing the confusion matrix, we obtain the following values: $\text{Sensibility}_{(\text{Open, Left, Deny})} = (0.77, 0.63, 0.95)$ and $\text{Specificity}_{(\text{Open, Left, Deny})} = (0.95, 0.97, 0.75)$. We can observe that in almost all cases, the hand gesture with true label "Deny" is identified correctly and only contains one FN. However, other hand gestures used during the recognition are identified as the label "Deny," which implies that the classifier generates a significant number of FPs for this gesture. This is corroborated when estimating the $\text{Precision}_{(\text{Open, Left, Deny})} = (0.89, 0.93, 0.65)$, where it is observed that the prediction of the "Deny" label contains more FPs than those in the other two classes. As for the final results globally, we have the following measures: $\text{acy} = 0.788$, $F1\_\text{score} = 0.788$, $\text{MCC} = 0.704$, and $\kappa = 0.682$. In this case, the MCC is more informative than the other measures, such as the *F1_score* or the *accuracy*, in the sense that it has better considered the relationships between the different values obtained in the confusion matrix. Thus, it obtains a result similar to Cohen's kappa (κ), altogether penalizing the difference between the quantities observed by its true labels and classification predictions.

5.2. Experimental Results of Correlation-Based Spatio-Temporal Human Gesture Recognition

A series of experiments was designed and implemented to verify the validity of the recognition methodology using the spatio-temporal correlation-based approach (presented in Subsection 4.2c. and Appendix A) for human gesture recognition under degraded environmental conditions. An integral imaging system consisting of a 3×3 synchronized camera array in the horizontal and vertical directions was used. The nine-camera (Mako G192C machine vision camera) system captured the video data of the scene from different viewing perspectives. The intrinsic parameters of each camera are identical, including a focal length of 50 mm, F/# of 1.8, and a frame rate of 20 frames/s.

Figure 31(a) shows the camera array used in the experiments. Figure 31(b) shows a complete human hand gesture action and a sampling stack formed by nine frames. In the experiments, five video datasets were collected for training the correlation filters (see Appendix A). A forefinger waving from left to right corresponding to the observer's viewing direction constitutes the training data [see Fig. 31(c)]. The test datasets include both true class hand gestures [see Fig. 31(c)] and false class hand gestures [see Fig. 31(d)]. The test data were collected from six people, and each person performed 10 gestures with five true class and five false class datasets.

Figure 31



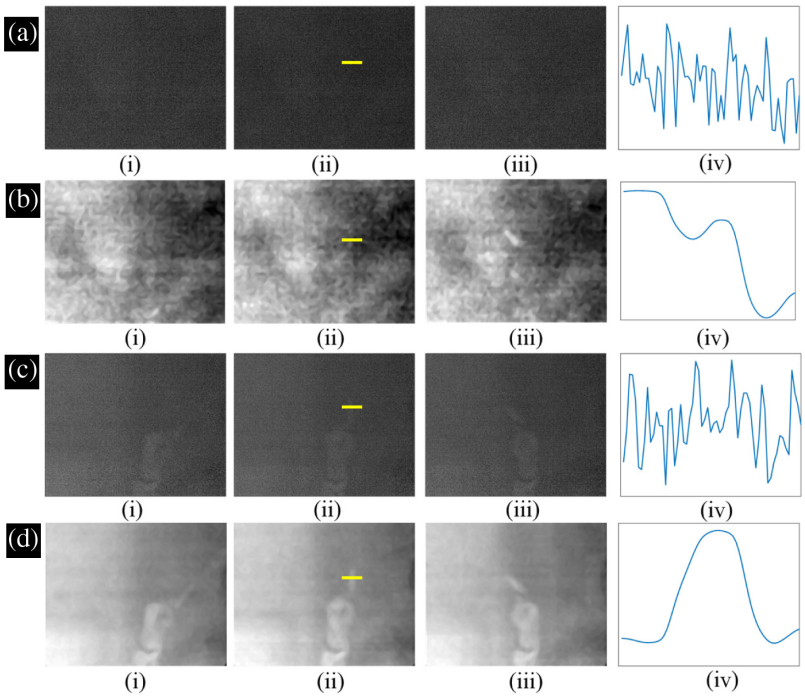
(a) (b) (c) (d)

(a) Integral imaging system used in the human gesture recognition experiment. (b) Examples of the spatio-temporal video frames. (c) Example of a true class human hand gesture frame. (d) Examples of false class human gestures [83].

The experiment was performed under low illumination conditions, in a low dynamic range scenario, and with a partial occlusion in front of the target. Figure 32(a) (i–iii) illustrates the original captured video frames, for which the reader may find it difficult to locate the target. With the application of the integral imaging computational reconstruction algorithm, the video sequence reconstructed at the object depth plane enhanced the signal intensity, with the occlusion out of focus in front of the target [see Fig. 32(c) (i–iii)], and the object of interest can be observed as well. To further improve the image quality, we applied the total variation (TV) denoising algorithm [106], which may smoothen the reconstructed image and preserve the edges. The results, shown in Fig. 32(d) (i–iv), provide enhanced video frames and clear human hand gesture features. The comparison results obtained by applying the total variation algorithm to the 2D frame without integral imaging reconstruction [see Fig. 32(i–iii)] indicate the advantages of the integral imaging reconstruction. The 1D intensity profiles [see Fig. 32(iv)] along the yellow lines [illustrated in Fig. 32(ii)] verify the capability of combining integral imaging and the TV algorithm for noise removal.

The test video datasets captured under degraded conditions are correlated in the frequency domain with the filters discussed in Subsection 4.2c and Appendix A. The correlation filters are trained using the true class dataset [see Fig. 31(b)], and we set the operator k as 0.3 for the nonlinear correlation process [see Eq. (A5) in Appendix A]. The ROC curves were applied to analyze the performance of the linear ($k = 1$) and nonlinear ($k < 1$) classifiers with the different computational imaging algorithms.

Figure 32



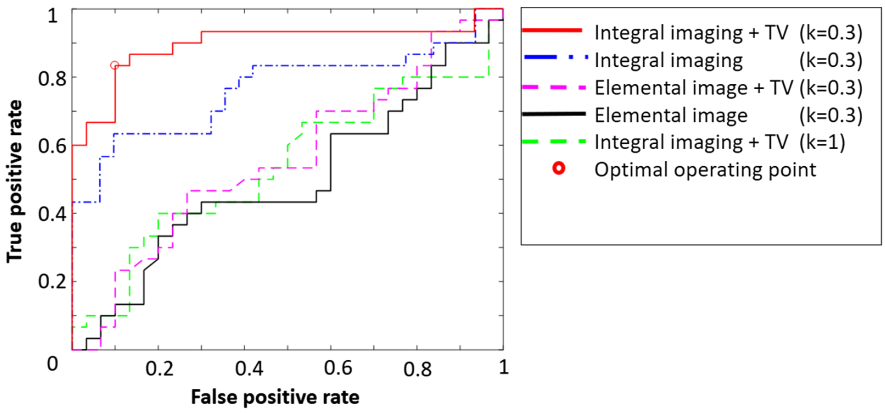
Video frames for a true class partially occluded human hand gesture under low light environment: (a) captured video sequence, (b) captured video sequence with the total variation (TV) denoising algorithm, (c) integral imaging-reconstructed video sequence, and (d) integral imaging-reconstructed video sequences with the TV denoising algorithm. (i)–(iii) Examples of video sequence and (iv) one-dimensional intensity profiles of the finger along the yellow lines in (ii) [83].

Figure 33 shows five ROC curves corresponding to the distortion-invariant filter [see Eq. (A2)] with a nonlinear ($k = 0.3$) and a linear correlation value ($k = 1$) for the distortion-invariant filtering process. Under the nonlinear correlation operation, we compared the results obtained by applying various algorithms to the captured video frames: (i) the original captured video sequences, (ii) the original captured video sequences with the TV denoising algorithm, (iii) integral imaging-reconstructed video sequences, and (iv) integral imaging-reconstructed video sequences with the TV denoising algorithm. The ROC curves show that the nonlinear correlation process outperforms the corresponding linear correlation counterpart. In addition, the ROC curve of the integral imaging-reconstructed video sequences with the TV denoising algorithm (see the red line in Fig. 34) gives the highest AUC value (AUC = 0.897).

Figure 34 shows five ROC curves corresponding to the nonlinear distortion-invariant correlation filter [see Eq. (A4)] with a nonlinear operation ($k = 0.3$). We also compared the results by applying various algorithms to the captured video frames: (i) the original captured video sequences, (ii) the original captured video sequences with the TV denoising algorithm, (iii) integral imaging-reconstructed video sequences, and (iv) integral imaging-reconstructed video sequences with the TV denoising algorithm. In addition, the simplified filter trained by the average templates was analyzed as indicated by the brown dotted line in Fig. 34. Among the various algorithms and approaches, the ROC curve corresponding to the integral imaging-reconstructed video sequences with the TV denoising algorithm (see red line in Fig. 34) yielded the highest AUC value (equal to 0.921). The AUC value for the case in which the filer was trained with the averaged template videos was 0.887, which can be considered an option for reducing the computational cost. The experimental results demonstrate the potential of using the spatio-temporal correlation-based approach when considering integral imaging and TV denoising algorithms for efficient human gesture recognition under degraded conditions.

Finally, we show a confusion matrix to compare the performance between the correlation-based approach with distortion-invariant correlation filters discussed

Figure 33



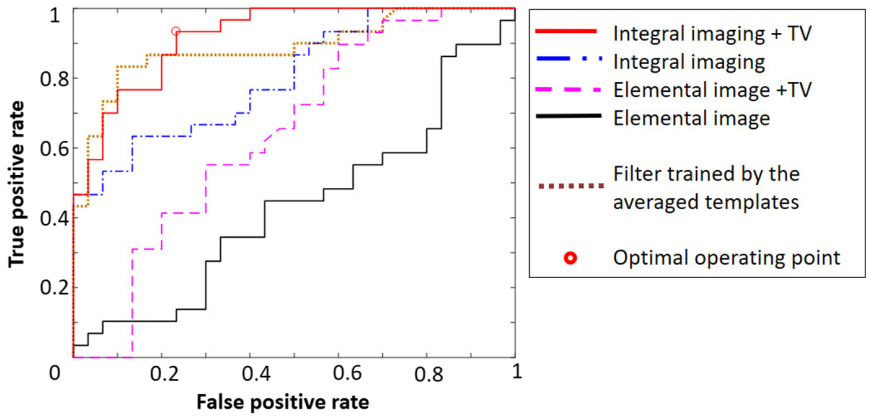
Receiver operating characteristic (ROC) curves for human gesture recognition using the optimum linear distortion-invariant filter and nonlinear transformations of the filter. For nonlinear correlation ($k = 0.3$), (a) red line: integral imaging-reconstructed video with the TV denoising algorithm, (b) blue line: integral imaging-reconstructed video, (c) magenta line: original video data with the TV algorithm, and (d) black line: original video data. For linear correlation ($k = 1$), (e) green line: integral imaging-reconstructed video with the TV denoising algorithm [83].

in this section and a previously reported method with a standard bag-of-features SVM framework [66] for human gesture recognition in degraded environments. The thresholds for the classification of the correlation-based approaches are calculated based on the optimal operating points obtained from the corresponding ROC curves as shown in Figs. 33 and 34. As illustrated in Table 8, the results of the comparison show that the nonlinear distortion-invariant filter (see Appendix A) outperforms the bag-of-features SVM framework for human gesture recognition in degraded conditions.

5.3. Human Gesture Recognition using 3D Integral Imaging and Deep Learning

In this subsection, we demonstrate representative results for 3D integral imaging-based human gesture recognition using deep neural networks. A two-class spatio-temporal gesture problem without occlusions was considered in this case. We used the integral imaging-based hybrid CNN-BiLSTM approach for gesture classification. The integral imaging-based reconstructed video has been fed into a CNN in order to extract the spatial feature vectors. The temporal dependency between the feature vectors of adjacent frames was captured using a BiLSTM network. The details of this neural network model are explained in Subsection 4.2b.

Figure 34



ROC curves for human gesture recognition using the nonlinear distortion-invariant filter with $k = 0.3$. (a) red line: integral imaging-reconstructed video with the TV denoising algorithm, (b) blue line: integral imaging-reconstructed video, (c) magenta line: original captured video with the TV denoising algorithm, and (d) black line: original captured video. For the simplified filter training process by averaging the template videos, (e) integral imaging-reconstructed video with the TV algorithm [83].

Table 8. Confusion Matrix for a Variety of Algorithms for Human Gesture Recognition and Classification[a]

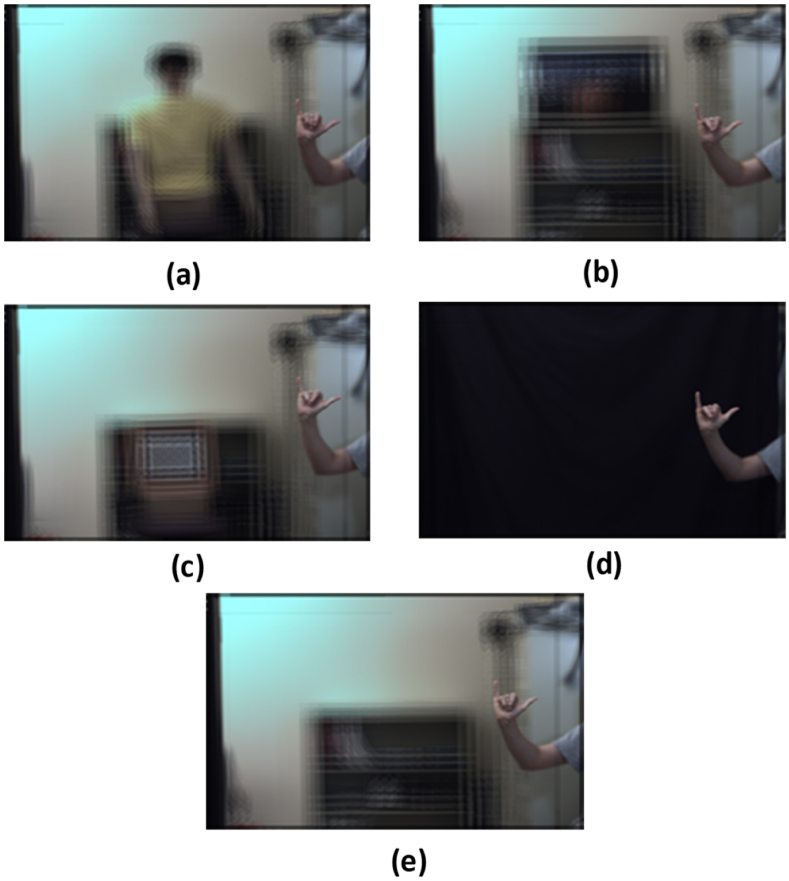
		Actual Condition								
		Linear Correlation Filter Used with Nonlinear Architecture, $k = 0.3$			kth-order Nonlinear Distortion-Invariant Filter, $k = 0.3$			Bag-of-Features SVM Framework		
Classified condition		True	False	Tot.	True	False	Tot.	True	False	Tot.
	True	TP = 25	FP = 3	28	TP = 28	FP = 7	35	TP = 13	FP = 8	21
	False	FN = 5	TN = 27	32	FN = 2	TN = 23	25	FN = 17	TN = 22	39
	Tot.	30	30	60	30	30	60	30	30	60
Sensitivity/TP rate			83.3%			93.3%			43.3%	
Specificity/TN rate			90%			76.7%			73.3%	
Accuracy/ACC			86.7%			85%			58.3%	

[a]TP, true positive; TN, true negative; FP, false positive; FN, false negative; Tot., total [83].

A series of videos was acquired from four participants for five different scene backgrounds using integral imaging capture without occlusion (Figs. 35 and 36). Each participant was asked to repeat the gesture twice (in order to capture the slow and the fast variations of the same gesture). In total, a series of 80 videos was recorded, with 40 videos corresponding to each gesture. A 3×3 camera array was used to capture the data. For each gesture, the data recorded were divided into two sets using a random split strategy. The first set, containing 30 videos for each gesture, was used for model training. To improve the performance and generalization capabilities of the model, we used data augmentation techniques (such as affine transformation, blurring, flipping, inversion, resizing, and noise addition). We trained the network using 420 videos, with 210 videos corresponding to each gesture class. The second (test) set, containing 10 videos for each gesture, was used to test the model. Figures 35 and 36 show the sample data collected for gestures 1 and 2, respectively.

The detection performance was compared using comparison metrics such as the accuracy, AUC, F1 score [107], precision, and MCC. We obtained an accuracy of 100% and a value of 1 for each of the F1 score, precision, and MCC. This is a clear indication that combining integral imaging with deep learning offers a promising approach to tackle human gesture recognition problems. The previous sections demonstrated the advantages of integral imaging techniques over alternative methodologies in degraded environments where there is occlusion, low levels of light, etc. This approach could be extended for multi-class gesture classification and more complex scenes under various degradation conditions. A more detailed study regarding

Figure 35



Sample video frames for gesture 1 with different backgrounds: (a) mannequin, (b) ball and cityscape, (c) checkboard, (d) curtain, and (e) wall. (Reprinted from [108].)

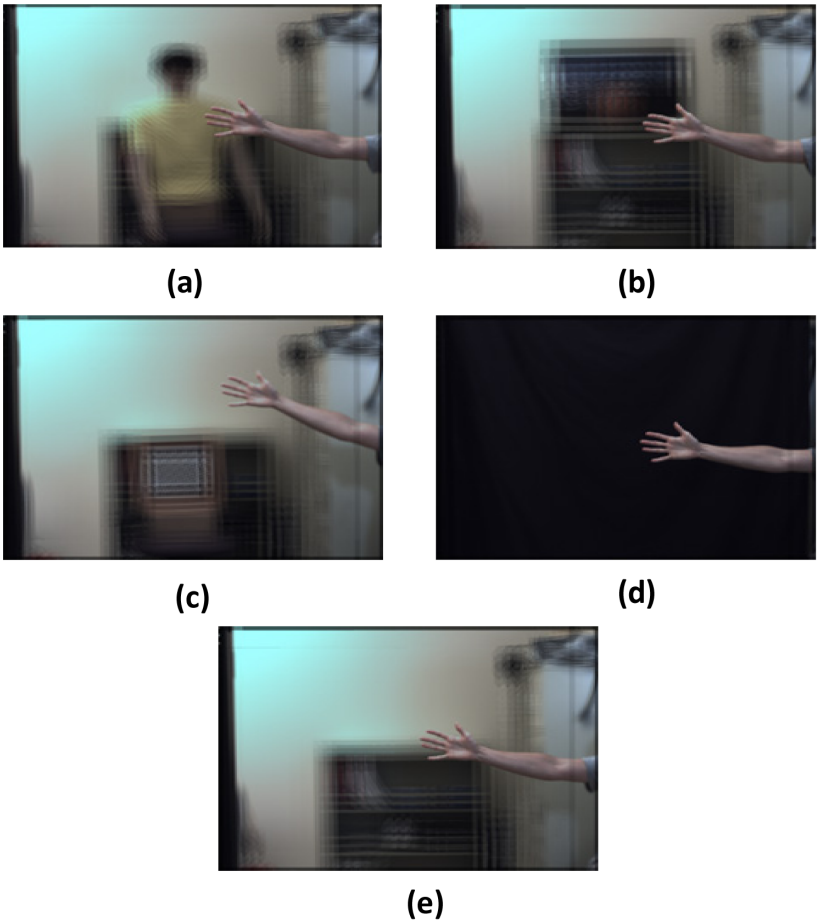
human gesture recognition under various degradations, such as partial occlusions, low illumination conditions, and multi-gesture scenarios, using an integral imaging-based deep learning algorithm is presented in [108]. Therefore, the approach demonstrated in this section can be extended for building a spatio-temporal gesture recognition system using 3D integral imaging and deep learning in degraded environments [108].

6. CONCLUSIONS

Automated human gesture recognition is an active research area encompassing a large number of application fields. In fact, research on human gesture recognition has increased significantly in the last 20 years, with particular focus on new sensing and acquisition technologies, and processing/classification algorithms and methodologies. This paper serves as a tutorial, presenting an overview of the current 3D image acquisition technologies used for gesture recognition, and discusses the most promising human gesture recognition algorithms. We presented experimental results to illustrate some examples of human gesture recognition using 3D integral imaging, comparing the results with those of 2D imaging. We provided examples of classifying human gestures under normal illumination, under low illumination, and in the presence of degraded conditions, such as occlusions, for both 2D and 3D imaging.

The tutorial is aimed at readers who may or may not be familiar with current 3D optical acquisition techniques as well as classification algorithms and methodologies

Figure 36



Sample video frames for gesture 2 with different backgrounds: (a) mannequin, (b) ball and cityscape, (c) checkboard, (d) curtain, and (e) wall. (Reprinted from [108].)

applied to human gesture recognition. We presented a systematic tutorial overview of the classification algorithms and techniques and 3D gesture acquisition methodologies based on 3D integral imaging and conventional 2D imaging. As with any tutorial of this nature, it is not possible to cover every related topic extensively in a single paper, and we may have inadvertently overlooked some relevant works. Nevertheless, 114 papers are cited, including overview articles, to assist the reader by covering relevant topics of interest in more detail [1–114].

APPENDIX A: SPATIO-TEMPORAL HUMAN GESTURE RECOGNITION IN DEGRADED ENVIRONMENTS

We consider the application of optimum distortion-invariant filters for high-quality correlation-based spatio-temporal human gesture recognition under degraded imaging conditions. The flowchart for the spatio-temporal correlation-based human gesture recognition is shown in Fig. 37. The filter design involves developing an optimum linear distortion-invariant filter [97] and/or a nonlinear distortion-invariant correlation filter [109]. The optimum linear distortion-invariant filter operates by maximizing the output peak-to-output-energy (POE) ratio. To train the filter, a continuous human gesture action video template $r_i(x, y, t)$ is used, where x and y are the spatial coordinates in each video frame, t represents the frame index, and i [1, 2, …, N] denotes the template video index.

The training data in the frequency domain can be expressed as a 1D vector, $S_{\text{vec}}(\omega) = S(u, v, \varphi) = \text{FT}[s(x, y, t)]$, which is stacked from the corresponding 3D data matrix for simplicity. FT represents the Fourier transform operation. The expression for the video data in the spatio-temporal domain can be obtained as follows [83]:

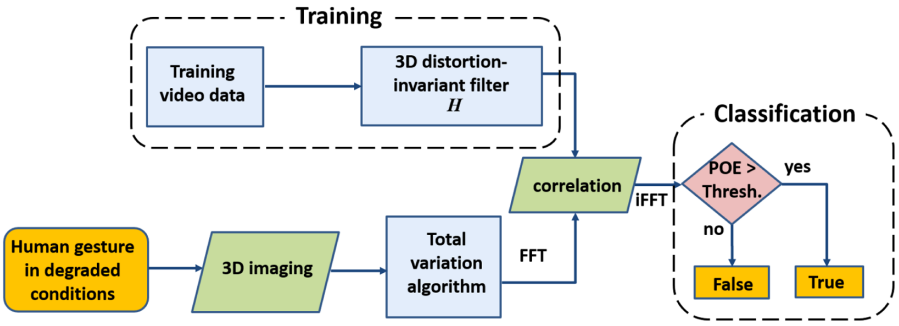
$$s(p) = \text{FT}^{-1}[S_{\text{vec}}(\omega)] = \text{vec}[s(x, y, t; z)], \tag{A1}$$

where FT^{-1} represents the inverse Fourier transform operation. The synthesized optimum linear distortion-invariant filter can then be given by [83,97]

$$H^*_{\text{opt}}(\omega) = \frac{E[S(\omega, \tau) \exp(j\omega\tau)]}{E[|S(\omega, \tau)|^2]}. \tag{A2}$$

A nonlinear distortion-invariant correlation filter [106,109] has also been considered for the spatio-temporal human gesture recognition. Compared with the conventional linear correlation filtering process, the nonlinear correlation in the Fourier plane can further enhance the correlator's performance with improved sharpness

Figure 37



Flowchart of the spatio-temporal correlation-based human gesture recognition proposed in [83]. FFT, fast Fourier transform; iFFT, inverse fast Fourier transform; Thresh, threshold; POE, correlation peak-to-output-energy ratio.

of the correlation peak. The nonlinearity extends the immunity to both in-plane distortions, such as object rotations, and out-of-plane distortions, such as viewing from different angles. For consistency, the series of training videos is denoted by $r_i(x, y, t)$. The corresponding vectorized dataset in the frequency domain is $R_i(\omega) = \text{vec}\{\text{FT}[r_i(x, y, t)]\}$, where $i = 1, 2, \ldots, N$. An N column matrix is formed as $S^k = [R_1^k(\omega), R_2^k(\omega), \ldots, R_N^k(\omega)]$, where the vector operator v^k is defined as [109]

$$v^k \triangleq \left[|v_1|^k \times \exp(j\phi_1), \quad |v_2|^k \times \exp(j\phi_2), \ldots, \quad |v_d|^k \times \exp(j\phi_d)\right]^T, \tag{A3}$$

where $(.)^T$ represents the transpose operation. Finally, the nonlinear distortion-invariant correlation filter can be synthesized as [109]

$$\boldsymbol{H}_k(\omega) = \left\{ S^k \left(\left[S^k \right]^+ S^k \right)^{-1} c* \right\}^{\frac{1}{k}}, \tag{A4}$$

where $(\cdot)^{-1}$ and $(\cdot)^+$ represent the inverse operation and complex-conjugate transpose, respectively, c is a constant for cross-correlation output origin values [109], $(\cdot)^*$ is the complex conjugate, and $\cdot^{1/k}$ follows Eq. (A3). In addition, a simplified implementation can be performed by applying all the reference videos to obtain an averaged template; the corresponding experimental results are shown in Fig. 34. As an extension to Eq. (39), the linear/nonlinear correlation can be expressed as

$$g(x, y, t; z) = \text{FT}^{-1}\left\{ [abs(S) \cdot abs(T)]^k \times \exp[j(\angle T - \angle S)] \right\}, \quad (k \in [0, 1]), \tag{A5}$$

where T is the Fourier transform of the test video data $[t(x, y, t; z)]$ with the in-focus depth z. Integral imaging reconstruction of the test data will be processed to focus on a specific depth, and the total variation (TV) denoising algorithm [106] will then be applied to the reconstructed dataset, in order to reduce the noise. The correlation output is obtained by applying an inverse Fourier transform operation. In Eq. (A5), $[\cdot]^k$ is an exponential operator, which determines if the correlation approach is linear $[k = 1]$ or nonlinear in nature. When $k < 1$, the correlation becomes a kth-order nonlinear correlation process [110,111].

Finally, the correlation peaks are classified and localized by analyzing the correlation POE ratio on the correlation output matrix. The POE is the ratio between the expected energy of the correlation peak and the expected energy of the output signal:

$$\text{POE} = \frac{|E[g(\tau, \tau)]|^2}{E\left\{ \overline{[g(p, \tau)]^2} \right\}}. \tag{A6}$$

FUNDING

Ministerio de Ciencia, Innovación y Universidades (Research Network RED2018-102511-T, RTI2018-099041-B-I00); Air Force Office of Scientific Research (FA9550-18-1-0338); Office of Naval Research (N000141712405, N000141712561, N000142012690); Generalitat Valenciana (PROMETEO/2019/048).

ACKNOWLEDGMENT

We thank the editor-in-chief, Prof. Guifang Li, and the editorial staff for the support and encouragement provided to prepare this manuscript. Many thanks to the anonymous reviewers for their detailed comments and suggestions to improve the

paper. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the U.S. Department of Defense.

DISCLOSURES

The authors declare no conflicts of interest.

REFERENCES

1. M. J. Cheok, Z. O. Mohamed, and H. Jaward, "A review of hand gesture and sign language recognition techniques," Int. J. Mach. Learn. Cybern. **10**, 131–153 (2019).
2. F. Chen, H. Lv, Z. Pang, J. Zhang, Y. Hou, Y. Gu, H. Yang, and G. Yang, "WristCam: a wearable sensor for hand trajectory gesture recognition and intelligent human–robot interaction," IEEE Sens. **19**, 8441–8451 (2019).
3. J. Yang, Y. Wang, Z. Lv, N. Jiang, and A. Steed, "Interaction with three-dimensional gesture and character input in virtual reality: recognizing gestures in different directions and improving user input," IEEE Consum. Electron. Mag. **7**(2), 64–72 (2018).
4. H. Cheng, L. Yang, and Z. Liu, "Survey on 3D hand gesture recognition," IEEE Trans. Circuits Syst. Video Technol. **26**, 1659–1673 (2016).
5. A. Al-Shamayleh, R. Ahmad, M. Abushariah, K. A. Alam, and N. Jomhari, "A systematic literature review on vision based gesture recognition techniques," Multimedia Tools Appl. **77**, 28121–28184 (2018).
6. M. Karam, "A framework for research and design of gesture-based human-computer," Ph.D. thesis (University of Southampton, 2006).
7. V. J. Traver, P. Latorre-Carmona, E. Salvador-Balaguer, F. Pla, and B. Javidi, "Human gesture recognition using three-dimensional integral imaging," J. Opt. Soc. Am. A **31**, 2312–2320 (2014).
8. V. J. Traver, P. Latorre-Carmona, E. Salvador-Balaguer, F. Pla, and B. Javidi, "Three-dimensional integral imaging for gesture recognition under occlusions," IEEE Signal Process. Lett. **24**, 171–175 (2017).
9. S. Giancola, M. Valenti, and R. Sala, *A Survey on 3D Cameras: Metrological Comparison of Time-of-Flight, Structured-Light and Active Stereoscopy Techniques* (Springer, 2018).
10. D. Bachmann, F. Weichert, and G. Rinkenauer, "Review of three-dimensional human-computer interaction with focus on the leap motion controller," Sensors **18**, 2194 (2018).
11. D. Pavllo, T. Porssut, B. Herbelin, and R. Boulic, "Real-time marker-based finger tracking with neural networks," in *IEEE Conference on Virtual Reality and 3D User Interfaces (VR)* (2018).
12. C. Zhu and W. Sheng, "Wearable sensor-based hand gesture and daily activity recognition for robot-assisted living," IEEE Trans. Syst. Man Cybern.—Part A: Systems and Humans **41**, 569–573 (2011).
13. J. Lien, N. Gillian, M. E. Karagozler, P. Amihood, C. Schwesig, E. Olson, H. Raja, and I. Poupyrev, "Soli: ubiquitous gesture sensing with millimeter wave radar," ACM Trans. Graph. **142**, 142:1–19 (2016).
14. F. Khan, S. K. Leem, and S. H. Cho, "Hand-based gesture recognition for vehicular applications using IR-UWB radar," Sensors **17**, 833 (2017).
15. H. Abdelnasser, K. Harras, and M. Youssef, "A ubiquitous WiFi-based fine-grained," IEEE Trans. Mobile Comput. **18**, 2474–2487 (2019).
16. Z. Tian, J. Wang, and X. Yang, "WiCatch: a Wi-Fi based hand gesture recognition system," IEEE Access **6**, 16911–16923 (2018).

17. H. Liu and L. Wang, "Gesture recognition for human-robot collaboration: a review," Int. J. Ind. Ergon. **68**, 355–367 (2018).
18. "Mobile hand tracking with gesture recognition," 2020, https://www.youtube.com/watch?v=hgTSQkoRUwU.
19. L. Motion, "Introducing the leap motion," 2012, https://www.youtube.com/watch?v=_d6KuiuteIA.
20. K. F. Windows, "Kinect for Windows retail clothing scenario video," 2013, https://www.youtube.com/watch?v=Mr71jrkzWq8.
21. BMW, "Gesture controls | BMW Genius How-To," 2015, https://www.youtube.com/watch?v=wqvAPskg_k0.
22. T. D'Orazio, R. Marani, V. Renò, and G. Cicirelli, "Recent trends in gesture recognition: how depth data has improved classical approaches," Image Vis. Comput. **52**, 56–72 (2016).
23. G. Lippmann, "Epreuves reversibles donnant la sensation du relief," J. Phys. **7**, 801–825 (1908).
24. D. F. Coffey, "Apparatus for making a composite stereograph,"U.S. patent 2,063,985 (15 November 1936).
25. N. Davies, M. McCormick, and L. Yang, "Three-dimensional imaging systems: a new development," Appl. Opt. **27**, 4520–4528 (1988).
26. H. Arimoto and B. Javidi, "Integral three-dimensional imaging with computed reconstruction," Opt. Lett. **26**, 157–159 (2001).
27. S. Manolache, A. Aggoun, M. McCormick, N. Davies, and S. Y. Kung, "Analytical model of a three-dimensional integral image recording system that uses circular- and hexagonal-based spherical surface microlenses," J. Opt. Soc. Am. A **18**, 1814–1821 (2001).
28. F. Okano, H. Hoshino, J. Arai, and I. Yuyama, "Real-time pickup method for a three-dimensional image based on integral photography," Appl. Opt. **36**, 1598–1603 (1997).
29. B. Javidi and F. Okano, *Three-Dimensional Television, Video, and Display Technologies* (Springer, 2002).
30. A. Isaksen, L. McMillan, and S. J. Gortler, "Dynamically reparameterized light fields," in *Proceedings of ACM Siggraph* (2000).
31. E. H. Adelson and J. R. Bergen, "The plenoptic function and the elements of early vision," Comput. Models Vis. Process. **1**, 3–20 (1991).
32. E. H. Adelson and J. Y. A. Wang, "Single lens stereo with plenoptic camera," IEEE Trans. Pattern Anal. Mach. Intell. **14**, 99–106 (1992).
33. R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan, *Light Field Photography with a Hand-Held Plenoptic Camera* (Stanford University, 2005).
34. A. Stern and B. Javidi, "Three-dimensional image sensing and reconstruction with time-division multiplexed computational integral imaging (CII)," Appl. Opt. **42**, 7036–7042 (2003).
35. M. Martinez-Corral, A. Dorado, J.-C. Barreiro, G. Saavedra, and B. Javidi, "Recent advances in the capture and display of macroscopic and microscopic 3-D scenes by integral imaging," Proc. IEEE **105**, 825–836 (2017).
36. B. Javidi, X. Shen, A. Markman, P. Latorre-Carmona, A. Martinez-Uso, J. Martinez Sotoca, F. Pla, M. Martinez-Corral, G. Saavedra, Y.-P. Huang, and A. Stern, "Multidimensional optical sensing and imaging systems (MOSIS): from macro to micro scales," Proc. IEEE **105**, 850–875 (2017).
37. X. Xiao, B. Javidi, M. Martinez-Corral, and A. Stern, "Advances in three-dimensional integral imaging: sensing, display, and applications," Appl. Opt. **52**, 546–556 (2013).

38. A. Stern and B. Javidi, "Three-dimensional image sensing, visualization and processing using integral imaging," Proc. IEEE **94**, 591–607 (2006).
39. X. Lin, J. Wu, G. Zheng, and Q. Dai, "Camera array based light field microscopy," Biomed. Opt. Express **6**, 3179–3189 (2015).
40. B. Wilburn, N. Joshi, V. Vaish, E.-V. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz, and M. Levoy, "High performance imaging using large camera arrays," ACM Trans. Graph. **24**, 765–776 (2005).
41. M. Martínez-Corral and B. Javidi, "Fundamentals of 3D imaging and displays: a tutorial on integral imaging, light-field, and plenoptic systems," Adv. Opt. Photon. **10**, 512–566 (2018).
42. M. Levoy, R. Ng, A. Adams, M. Footer, and M. Horowitz, "Light field microscopy," ACM Trans. Graph. **25**, 924–934 (2006).
43. M. Levoy, Z. Zhang, and I. McDowall, "Recording and controlling the 4D light field in a microscope using microlens arrays," J. Microsc. **235**, 144–162 (2009).
44. J. Kramer, N. Burrus, F. Echtler, H. Daniel, and M. Parker, *Hacking the Kinect* (Springer, 2012).
45. A. Shpunt and Z. Zalevsky, "Depth-varying light fields for three dimensional sensing," U.S. patent 8,504,618B2 (1 November 2008).
46. K. Khoshelham, "Accuracy analysis of kinect depth data," Int. Arch. Photogrammetry, Remote Sens. Spatial Inf. Sci. **38**, 133–138 (2011).
47. A. Fossati, J. Gall, H. Grabner, X. Ren, and K. Konolige, *Computer Depth Cameras for Computer Vision* (Springer, 2013).
48. D. Nitzan, A. E. Brain, and R. O. Duda, "Measurement and use of registered reflectance and range data in scene analysis," Proc. IEEE **65**, 206–220 (1977).
49. R. Lange, P. Seitz, A. Biber, and R. Schwarte, "Time-of-flight range imaging with a custom solid-state image sensor," Laser Metrol. Insp. **3823**, 180–191 (1999).
50. S. Hong, G. Saavedra, and M. Martínez-Corral, "Full parallax 3D display from Kinect v1 and v2," Opt. Eng. **56**, 041305 (2016).
51. H. Gonzalez-Jorge, P. Rodríguez-Gonzálvez, J. Martínez-Sánchez, D. González-Aguilera, P. Arias, M. Gesto, and L. Díaz-Vilariño, "Metrological comparison between Kinect I and Kinect II sensors," Measurement **70**, 21–26 (2015).
52. Y. He, B. Liang, Y. Zou, J. He, and J. Yang, "Depth errors analysis and correction for time-of-flight (ToF) cameras," Sensors **17**, 92 (2017).
53. K. Taguchi and J. S. Iwanczyk, "Vision 20/20: single photon counting x-ray detectors in medical imaging," Med. Phys. **40**, 100901 (2013).
54. E. L. Dereniak and G. L. Boreman, *Infrared Detectors and Systems*, Wiley Series in Pure and Applied Optics (1996).
55. R. A. Schowengerdt, *Remote Sensing: Models and Methods for Image Processing* (Academic, 2006).
56. B. Javidi, I. Moon, and S. Yeom, "Three-dimensional identification of biological microorganism using integral imaging," Opt. Express **14**, 12096–12108 (2006).
57. K. Lange and R. Carson, "EM reconstruction algorithms for emission and transmission tomography," J. Comput. Assist. Tomogr. **8**, 306–316 (1984).
58. S. D. Konecky and B. J. Tromberg, "Imaging: focusing light in scattering media," Nat. Photonics **5**, 135–136 (2011).
59. J. Rosen and D. Abookasis, "Seeing through biological tissues using the fly eye principle," Opt. Express **11**, 3605–3611 (2003).
60. V. Durán, F. Soldevila, E. Irles, P. Clemente, E. Tajahuerce, P. Andrés, and J. Lancis, "Compressive imaging in scattering media," Opt. Express **23**, 14424–14433 (2015).
61. A. Stern, D. Aloni, and B. Javidi, "Experiments with three-dimensional integral imaging under low light levels," IEEE Photon. J. **4**, 1188–1195 (2012).

62. A. Markman, X. Shen, and B. Javidi, "Three-dimensional object visualization and detection in low light illumination using integral imaging," Opt. Lett. **42**, 3068–3071 (2017).
63. A. Stern and B. Javidi, "Random projections imaging with extended space-bandwidth product," J. Disp. Technol. **3**, 315–320 (2007).
64. L. Cao and J. Peter, "Iterative reconstruction of projection images from a microlens-based optical detector," Opt. Express **19**, 11932–11943 (2011).
65. J. K. Aggarwal and L. Xia, "Human activity recognition from 3D data: a review," Pattern Recognit. Lett. **48**, 70–80 (2014).
66. H. Wang, M. M. Ullah, A. Kläser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," *British Machine Vision Conference (BMVC)*, London, UK, September 7–10, 2009.
67. E. Tapia, "A note on the computation of high-dimensional integral images," Pattern Recognit. Lett. **32**, 197–201 (2011).
68. L. Wang and K. L. Chan, "Learning Kernel parameters by using class separability measure," in *Neural Information Processing Systems* (2002).
69. C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings of the Alvey Vision Conference* (1988).
70. N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods* (Cambridge University, 2002).
71. A. Wedel, T. Brox, T. Vaudrey, C. Rabe, U. Franke, and D. Cremers, "Stereoscopic scene flow computation for 3D motion understanding," Int. J. Comput. Vis. **95**, 29–51 (2011).
72. J. Cech, J. Sanchez-Riera, and R. Horaud, "Scene flow estimation by growing correspondence seeds," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)* (2011).
73. S. R. Fanello, I. Gori, G. Metta, and F. Odone, "Keep it simple and sparse: real-time action recognition," J. Mach. Learn. Res. **14**, 2617–2640 (2013).
74. G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," in *Scandinavian Conference on Image Analysis (SCIA)* (2003).
75. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature **521**, 436–444 (2015).
76. S. Herath, M. Harandi, and F. Porikli, "Going deeper into action recognition: a survey," Image Vis. Comput. **60**, 4–21 (2017).
77. P. Wang, W. Li, P. Ogunbona, J. Wan, and S. Escalera, "RGB-D-based human motion recognition with deep learning: a survey," Comput. Vis. Image Underst. **171**, 118–139 (2018).
78. F. Zhu, L. Shao, and J. Xie, "From handcrafted to learned representations for human action recognition: a survey," Image Vis. Comput. **55**, 42–52 (2016).
79. B. Liu, H. Cai, Z. Ju, and H. Liu, "RGB-D sensing based human action and interaction analysis: a survey," Pattern Recognit. **94**, 1–12 (2019).
80. S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," IEEE Trans. Pattern Anal. Mach. Intell. **35**, 221–231 (2012).
81. J. Y.-H. Ng, M. Hausknecht, M. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: deep networks for video classification," in *Proceedings of the IEEE Conference On Computer Vision and Pattern Recognition (CVPR)* (2015).
82. L. Pigou, A. van den Oord, S. Dieleman, M. Van Herreweghe, and J. Dambre, "Beyond temporal pooling: recurrence and temporal convolutions for gesture recognition in video," Int. J. Comput. Vis. **126**, 430–439 (2018).

83. X. Shen, H.-S. Kim, K. Satoru, A. Markman, and B. Javidi, "Spatial-temporal human gesture recognition under degraded conditions using three-dimensional integral imaging," Opt. Express **26**, 13938–13951 (2018).
84. G. Zhu, L. Zhang, P. Shen, and J. Song, "Multimodal gesture recognition using 3-D convolution and convolutional LSTM," IEEE Access **5**, 4517–4524 (2017).
85. S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Comput. **9**, 1735–1780 (1997).
86. S. H. Hong, J. S. Jang, and B. Javidi, "Three-dimensional volumetric object reconstruction using computational integral imaging," Opt. Express **12**, 483–491 (2004).
87. J. Deng, D. Wei, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: a large-scale hierarchical image database," in *Proceedings of the IEEE International Conference On Computer Vision and Pattern Recognition* (2009).
88. T. Serre, L. Wolf, S. M. Bileschi, M. Riesenhuber, and T. Poggio, "Robust object recognition with cortex-like mechanisms," IEEE Trans. Pattern Anal. Mach. Intell. **29**, 411–426 (2007).
89. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," arXiv:1409.4842 (2014).
90. D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proceedings of the IEEE International Conference On Computer Vision* (2015).
91. G. Turin, "An introduction to matched filters," IRE Trans. Inf. Theory **6**, 311–329 (1960).
92. A. V. Lugt, "Signal detection by complex spatial filtering," IEEE Trans. Inf. Theory **10**, 139–145 (1964).
93. J. W. Goodman, *Introduction to Fourier Optics* (Roberts & Company, 2005).
94. D. Casasent and D. Psaltis, "Position, rotation, and scale invariant optical correlation," Appl. Opt. **15**, 1795–1799 (1976).
95. F. Sadjadi and A. Mahalanobis, "Target-adaptive polarimetric synthetic aperture radar target discrimination using maximum average correlation height filters," Appl. Opt. **45**, 3063–3070 (2006).
96. R. Jain, R. Kasturi, and B. G. Schunck, *Machine Vision* (Mc-Graw-Hill, 1995).
97. B. Javidi and J. Wang, "Optimum distortion-invariant filter for detecting a noisy distorted target in nonoverlapping background noise," J. Opt. Soc. Am. A **12**, 2604–2614 (1995).
98. P. Réfrégier, V. Laude, and B. Javidi, "Basic properties of nonlinear global filtering techniques and optimal discriminant solutions," Appl. Opt. **34**, 3915–3923 (1995).
99. J. A. Ward, P. Lukowicz, and H. W. Gellersen, "Performance metrics for activity recognition," ACM Trans. Intell. Syst. Technol. **2**, 6 (2011).
100. L. V. Nguyen-Dinh, A. Calatroni, and G. Tröster, "Robust online gesture recognition with crowd sourced annotations," J. Mach. Learn. Res. **15**, 3187–3220 (2014).
101. B. W. Matthews, "Comparison of the predicted and observed secondary structure of T4 phage lysozyme," Biochim. Biophys. Acta, Mol. Basis Dis. **405**, 442–451 (1975).
102. T. Fawcett, "An introduction to ROC analysis," Pattern Recognit. Lett. **27**, 861–874 (2006).
103. J. Cohen, "A coefficient of agreement for nominal scales," Educ. Psychol. Meas. **20**, 37–46 (1960).
104. A. Vedaldi and B. Fulkeson, "VLFeat: an open and portable library of computer vision algorithms," 2019, http://www.vlfeat.org.

105. C.-C. Chang and C.-J. Lin, "LIBSVM—a library for support vector machines," 2019, https://www.csie.ntu.edu.tw/~cjlin/libsvm/.
106. L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," Physica D **60**, 259–268 (1992).
107. I. Martin-Diaz, D. Morinigo-Sotelo, O. Duque-Perez, and R. J. Romero-Troncoso, "Advances in classifier evaluation: novel insights for an electric data-driven motor diagnosis," IEEE Access **4**, 7028–7038 (2016).
108. G. Krishnan, R. Joshi, T. O'Connor, F. Pla, and B. Javidi, "Human gesture recognition under degraded environments using 3D-integral imaging and deep learning," Opt. Express **28**, 19711–19725 (2020).
109. B. Javidi and D. Painchaud, "Distortion-invariant pattern recognition with Fourier-plane nonlinear filters," Appl. Opt. **35**, 318–331 (1996).
110. B. Javidi, "Nonlinear joint power spectrum based optical correlation," Appl. Opt. **28**, 2358–2367 (1989).
111. P. Refregier, V. Laude, and B. Javidi, "Nonlinear joint transform correlation: an optimum solution for adaptive image discrimination and input noise robustness," Opt. Lett. **19**, 405–407 (1994).
112. B. Javidi, A. Carnicer, J. Arai, T. Fujii, H. Hua, H. Liao, M. Martínez-Corral, F. Pla, A. Stern, L. Waller, Q. H. Wang, G. Wetzstein, M. Yamaguchi, and H. Yamamoto, "Roadmap on 3D integral imaging: sensing, processing, and display," Opt. Express **28**, 32266–32293 (2020).
113. J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu, "Robust 3D action recognition with random occupancy patterns," in *Computer Vision—(ECCV)*, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, eds., Vol. **7573** of Lecture Notes in Computer Science (Springer, 2012), Vol. **7573**.
114. A. Kurakin, Z. Zhang, and Z. Liu, "A real time system for dynamic hand gesture recognition with a depth sensor," *Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, Bucharest, Romania (2012), pp. 1975–1979.

Prof. Bahram Javidi received his B.S. degree from George Washington University and the Ph.D. from the Pennsylvania State University in electrical engineering. He is a Board of Trustees Distinguished Professor at the University of Connecticut. Professor Javidi's interests are in a broad range of transformative imaging approaches using optics and photonics, and he has made seminal contributions to passive and active multi-dimensional imaging from nano- to micro- and macroscales. His recent research activities include 3D visualization and recognition of objects in photon-starved environments; automated disease identification using biophotonics with low-cost compact sensors; information security, encryption, and authentication using optical imaging; non-planar flexible 3D image sensing; and bio-inspired imaging. Professor Javidi has been named Fellow of several societies, including IEEE, The Optical Society (OSA), SPIE, EOS, and IoP. Early in his career, the National Science Foundation named him a Presidential Young Investigator. Professor Javidi has received the OSA C. E. K. Mees medal (2019), the IEEE Photonics Society William Streifer Scientific Achievement Award (2019), the OSA Fraunhofer Award/Robert Burley Prize (2018), the Prize for Applied Aspects of Quantum Electronics and Optics of the European Physical Society (2015), the SPIE Dennis Gabor Award in Diffractive Wave Technologies (2005), and the SPIE Technology Achievement Award (2008). In 2008, he was awarded the IEEE Donald G. Fink Paper Prize and the John Simon Guggenheim Foundation Fellow Award. In 2007, the Alexander von

Humboldt Foundation (Germany) awarded Prof. Javidi the Humboldt Prize. He is an alumnus of the Frontiers of Engineering of The National Academy of Engineering (2003-). His papers have been cited more than 47,000 times (h-index=101) according to a Google Scholar citation report.

Filiberto Pla received his B.Sc. and Ph.D. degrees in physics from the Universitat de Valéncia, Valencia, Spain, in 1989 and 1993, respectively. He is currently a Full Professor with the Departament de Llenguatges i Sistemes Informatics, Universitat Jaume I, Castellón de la Plana, Spain. He has been a Visiting Scientist with the Silsoe Research Institute, University of Surrey, Guildford, UK; the University of Bristol, Bristol, UK; CEMAGREF, Montpellier, France; the University of Genoa, Genoa, Italy; the Instituto Superior Técnico, Lisbon, Portugal; the Swiss Federal Institute of Technology, ETH Zurich, Zürich, Switzerland; the idiap Research Institute, Switzerland; the Delft University of Technology, Delft, The Netherlands; and the Mid Sweden University, Sweden. He was Director of the Institute of New Imaging Technologies, Universitat Jaume I. His research interests include color and spectral image analysis, visual motion analysis, 3D image processing, and pattern recognition and learning techniques applied to image processing. Professor Pla is a member and was chairman of the Spanish Association for Pattern Recognition and Image Analysis, which is part of the International Association for Pattern Recognition.

José Martínez Sotoca received his B.Sc. degree in physics from the Universidad Nacional de Educación a Distancia, Madrid, Spain, in 1996 and the M. Sc. and Ph. D. degrees in physics from the University of Valencia, Valencia, Spain, in 1999 and 2001, respectively. His Ph.D. work was on surface reconstructions with structured light. He is currently an Associate Professor in the Department of Computer Languages and Systems, Universitat Jaume I, Castellón de la Plana, Spain. He has collaborated in different projects of computer science, machine learning, etc. He has published more than 100 scientific papers in national and international conference proceedings, books, and journals. His research interests include pattern recognition, image analysis, quantum mechanics, and game engines. He is a member of the International Association for Pattern Recognition.

Xin Shen received his B.Sc. degree in Optical Information Science and Technology from Xidian University, Xi'an, China, in 2010. In 2013, Xin received a dual M.S. degree: one M.S. degree in Electronics and Communication Engineering from Xidian University and one M.S. degree in Engineering from Doshisha University, Kyoto, Japan. Xin later received a third M.S. degree in Electrical Engineering from the University of Connecticut (UConn), Storrs, Connecticut, USA, in 2016. Xin completed his Ph.D. work on Three-Dimensional Image Sensing and Visualization with Augmented Reality and received his Ph.D. degree in Electrical Engineering at UConn in 2018. Dr. Xin Shen is currently an Assistant Professor in the Computer Science Department,

Massachusetts College of Liberal Arts. Xin was the vice president of the student chapter of The Optical Society (OSA) at the University of Connecticut and the vice president of the student chapter of the Society of Photo-Optical Instrumentation Engineers (SPIE) at the University of Connecticut. In 2018, Xin was trained as a research intern at Bell Labs, New Jersey, USA. Xin is a member of SPIE and OSA and a recipient of the 2017 UTC-IASE Endowed Graduate Fellowship, the 2016 Graduate Predoctoral Fellowship. His research interests include optical sensing, optical and digital image processing, and 3D visualization.

Pedro Latorre Carmona is an assistant professor at the University of Burgos, Spain. He received his B.S. degree in physics from the University of Valencia, Spain, in 1999 and his Ph.D. degree in computer science from the Polytechnical University of Valencia in 2005. His current research interests are 3D image processing and analysis, feature selection and extraction, pattern recognition, multispectral (including remote sensing) image processing, and Colorimetry and Vision Physics.

Manuel Martinez-Corral received his Ph.D. degree in Physics (Best Thesis Award) from the University of Valencia in 1993. Currently Full Professor of Optics at the University of Valencia, he co-leads the "3D Imaging and Display Laboratory." He was elected Fellow of the SPIE in 2010 and Fellow of the OSA in 2017. His research interests include resolution procedures in 3D scanning microscopy and 3D imaging and display technologies. He has supervised 17 Ph.D. theses on these topics (three honored with the Best Thesis Award), published more than 120 technical articles in major journals (which received more than 3000 citations), and pronounced a number of invited and keynote presentations in international meetings. He is also co-inventor of 12 patents, one of them supporting the creation of one Spin-off of the University of Valencia. He is has served in the Program Committee of a number of Conferences sponsored by SPIE, OSA, IEEE, etc., and currently is co-chair of the Three-Dimensional Imaging, Visualization, and Display Conferences within the SPIE meeting in Defense, Security, and Sensing. He is a Topical Editor of the OSA Journal Applied Optics.

Ruben Fernandez-Beltran earned his B.Sc. degree in Computer Science, his M.Sc. in Intelligent Systems, and his Ph.D. degree in Computer Science from Universitat Jaume I (Castellon de la Plana, Spain) in 2007, 2011, and 2016, respectively. He was awarded the Outstanding Ph.D. Dissertation Award at Universitat Jaume I in 2017. He is currently a postdoctoral researcher within the Computer Vision Group of the University Jaume I as a member of the Institute of New Imaging Technologies. He has been a visiting researcher at the University of Bristol (UK), the University of Extremadura (Caceres, Spain), and the Technische Universität Berlin in Germany. He is a member of the Spanish Association for Pattern Recognition and Image Analysis (AERFAI), which is part of the International Association for Pattern Recognition (IAPR).

Gokul Krishnan received his B.Tech. degree in Electronics and Communication engineering from Mahatma Gandhi University, India and his M.S. in Electrical Engineering with specialization in signal processing and communication from the Indian Institute of Technology (IIT), Kanpur, in 2015 and 2018, respectively. He worked as a Systems Engineer in the CTO—Research and Innovation Labs, Tata Consultancy Services (TCS), Bangalore, from 2018 to 2019. Currently, he is pursuing a Ph.D. degree in Electrical and Computer Engineering from the University of Connecticut, Storrs, USA. His research interests are in multidimensional signal processing, 3D imaging, and deep learning.