

Hand Recognition Using Depth Cameras

Reconocimiento De Manos Usando Cámaras De Profundidad

Alexander Cardona López

Fundación Universidad Autónoma de Colombia, Bogotá, Colombia, alexander.cardona@fuac.edu.co

Received: Sept. 07 2015

Accepted: October 13 2015

Available Online: Oct. 26 2015

Abstract

Hand position and gesture recognition from a series of images is a topic of relevance for the development of human-machine interaction. The advent of low-cost consumer devices, such as Microsoft Kinect, leaves open the possibility of creating recognition applications that are not affected by low-light conditions. This paper is a survey of the literature on hand position and gesture recognition with the use of depth cameras. Most studies noticeably focus on the recognition of one-handed gestures and their classification within a pre-established set of gestures. Only in research from recent years does one see significant advances in the identification of unconstrained hand poses, that is, inference from the skeleton with the use of depth and color information. Nevertheless, the lack of a standardized set of tests and the diversity of hardware leaves unclear the extent to which these would prove effective with low-cost hardware.

Keywords: depth image, gesture recognition, tracking.

Resumen

La detección de la posición y postura de las manos en una serie de imágenes es un problema de interés para el desarrollo de interacciones hombre-máquina. El surgimiento de sensores de profundidad de bajo costo como el Kinect de Microsoft, ha abierto la posibilidad de crear aplicaciones de detección cuyo desempeño no se ve afectado por una deficiente iluminación. Este documento hace una revisión a diferentes investigaciones que hacen uso de sensores de profundidad, para el reconocimiento del movimiento o postura de las manos. Se puede observar que la mayoría de trabajos se centran en el reconocimiento de una sola mano y en la clasificación de su postura entre un conjunto pre-establecido de posturas. Únicamente en trabajos de años recientes se observan avances significativos en la identificación de la postura de la mano sin restricción alguna, esto es, la inferencia de un esqueleto a partir de la información de profundidad y color. Sin embargo, la falta de un conjunto estándar de pruebas y la multiplicidad de hardware empleado en las pruebas, no deja del todo claro el alcance que tendrían las técnicas presentadas en dispositivos de bajo costo.

Palabras clave: Imagen de profundidad, Reconocimiento de gestos, Seguimiento.

1. Introduction

The advent of inexpensive depth sensors such as Microsoft's Kinect has opened the possibility of new solutions to the traditional problem of detecting objects or people in an image or sequence of images. Generally, these devices capture the distance from a set of points relative to the location of the sensor (values that are commonly called depth image and depth map).

Thus, while the value of a traditional image represents the color of a point, the value of a depth image represents the distance from one point to the camera.

Much of the research on these devices is based on the modification of existing techniques or the creation of new techniques to take advantage of new depth data and to perform tasks that are difficult to perform, using only the color information given by traditional images. The recognition of hand position, orientation, and pose are examples of applications that are benefiting from the use of

*Corresponding Author.

E-mail: alexander.cardona@fuac.edu.co

How to cite: Cardona Lopez, Alexander; *Hand Recognition Using Depth Cameras*, TECCIENCIA, Vol. 10 No. 19., 49-48, 2015, DOI: <http://dx.doi.org/10.18180/tecciencia.2015.19.10>

depth maps and that are proving of great interest in the development of human-machine interaction software. This is a field that has received much attention in recent years, and it is thus possible to find a variety of interactive applications that use depth sensors. The complexity of the issue and the sheer variety of implementations have led to drawbacks for its general use [1].

Even though human body recognition has been worked on in a broad range of interactive applications, particularly video games [2] and robotics [3], hand gesture recognition curiously has not, at least in the case of depth images. One study [4] asserts that at the time of the study, 75% of applications belonged to one of just three categories: interactive displays, robot movement control, and sign language recognition. Moreover, some of these applications are still quite limited since the problem of unconstrained recognition is complex and remains an open research question, especially when using low-cost devices, which do not perform adequately for more specialized tasks. In this sense, one can still find recent research on hand recognition focused solely on the use of traditional images, such as studies [5], [6], [7] and [8].

Several studies have been performed in the last four years using depth values to recognize people, objects, and gestures ([9], [10], [11], [12]); likewise, there are few studies that highlight the importance of hand gesture recognition ([13], [4]). One notices that in these studies there exists no commonly used set of algorithms for recognizing one or both hands. Nonetheless, the process is very similar to that performed in applications that process traditional images, so it is common to observe filtering, segmentation, and classification. The choice of the specific algorithm to carry out each of the above tasks varies widely from one study to another, even when the applications presented have a similar objective.

In this paper, I will briefly describe the difficulties that often arise during the use of depth maps in recognition tasks, as well as some of the approaches taken in various studies to achieve full or partial recognition of hand positions and poses. After briefly describing the features of a depth map, I will detail the path followed by several studies for performing segmentation, tracking, and classification tasks. Noise filtering tasks will be not addressed here because many of the techniques therein are quite similar to those used on images.

2. Depth Map

Depth maps can be viewed as an array of values where each value represents the distance from the capture device to a fixed point. This is in contrast to traditional images, where each value or pixel represents a point that is captured by the

camera. When the location of the camera relative to the depth sensor is known, the relationship between the color image and the depth map can be calculated so as to obtain the distance where a point of a specified color is found.

Today, various types of affordable devices can be found that generate color images and related depth images, both normally accessible from libraries that allow access to such equipment [14], [15].

Depth maps have been used in different fields for decades and are not only produced by specialized devices. It is also possible to obtain a depth map by projecting a surface onto a flat plane, a technique that is regularly used in computer graphics for generating images from three-dimensional surfaces, with the depth map in this case commonly called a depth buffer [16]. It is also possible to generate a depth map from two traditional images that focus on the same scene from different points of view (stereogram). Map quality naturally varies depending on the technique or the hardware used to produce it.

Depth maps aid some of the processing tasks previously done only with color images, especially the work of segmentation and tracking, which in traditional image techniques are affected by the illumination received by the objects.

However, depth map values have limitations that influence the development of new algorithms, including:

- The depth map and the color image are captured by different sensors, so it is necessary to know the exact characteristics of each sensor in order to link the two datasets.
- As with color images, a depth map does not have exact figures, since it is influenced by a variety of noise. Such noise must be filtered or at least taken into account in the recognition algorithm to minimize erroneous classification.
- It is common for the depth map to contain “gaps” corresponding to parts of the surface that are not reached by the sensor. There can even be pixels in the color image corresponding to “gaps” in the depth map.
- The capture rate (frames per second) of low-cost depth sensors is usually lower than the rate at which color images are captured.

Likewise, while some algorithms work directly on the depth information contained in the map, others work on the point cloud in the 3D space. The point cloud is the set of points resulting from determining the spatial position of each point with the depth map and the physical characteristics of the

sensor used (sampling capability, wheel sensors, viewing angle).

The point cloud thus maintains the XYZ coordinates of each point, so it is better suited for distance measurements.

Overall, despite the differences in color images, most applications that work with depth maps end up doing similar processing tasks: noise removal, segmentation, tracking, and classification, all complements to existing algorithms.

3. Hand Recognition

For hand recognition, the focus is on the region of the image corresponding to the hands, so it is common to need to select the entire set of points that matter. This segmentation work is complicated by the non-rigid nature of the hands, which means that the shape as it appears to the camera is not always the same. Moreover, the speed of hand motion often exceeds the capture speed of the sensor. Also, the problem is even more complicated if one must consider the interaction of both hands together or of a hand with other objects.

Often the problem lies not only in finding the position and orientation of the hand but in tracking its movement over time (in a series of images). In this case, it is not usually possible to perform segmentation on each frame, either because the segmentation algorithm is not sufficient or because the capture rate of the device does not allow it. Hence, it is necessary to use techniques that help to "predict" the movement made by the hand in each frame.

In the case of interactive applications, the variable of interest is not only tracking hand position, but also knowing which gesture is being performed (e.g. if a finger points in a particular direction, if the fingers make a symbol, etc.). In these cases, one turns to classification methods, in which a previously stored set of gestures is used to determine if the captured hand region resembles any.

It is also possible to achieve a simplified representation of the performed gesture (usually a skeleton composed of line segments) by trying to determine the location and orientation of each of the components of the hand. This is a more complex task but one that enables applications not to be restricted to a limited set of gestures.

Yet an application does not necessarily have to resolve each of the aforementioned shortcomings. The actions taken depend on the purpose of the application being developed. Nor is it necessary to use an algorithm to solve every problem, as it is possible for a single algorithm to confront several of the aforementioned aspects simultaneously.

4. Segmentation

With segmentation, we seek to subdivide the input data to form subsets that are easier to analyze. In the case of hand recognition in depth images, the goal of segmentation is normally to differentiate the set of points corresponding to the hands, or at least to locate the possible position of the hand relative to the sensor, in order to facilitate the creation of further algorithms.

There are several approaches to this problem. One of the most popular is to use information from the color image to determine the regions of the hand. Bhuyan et al. [17], for example, use the color histogram of images, classified using a previously created histogram database. Similar techniques are employed in other studies ([18], [19], and [20]), where color is the main source of data for segmentation.

This approach proved successful in these cases and has the advantage of being widely used in applications that do not use depth sensors. However, because it is based on images, the process is affected by skin tone variation and changes in illumination, which is reflected in the use of many descriptors for the classification algorithm.

It is also possible to utilize the information in the depth map to determine hand position. Several applications ([21], [22], [23], [24]) make the assumption that the hand corresponds to the region closest to the sensor. This assumption is usually adequate for applications oriented to human-machine interaction, when the hands are away from the body. Under this assumption, finding the region corresponding to the hand can be seen as a question of finding the regions closest to the sensor points. One study [25] simply selects the closest points in a distance range less than a given value. Another [26] similarly performs the segmentation using a range of distances, but, in this case, the range is small, leading to the use of an iterative process that adds small sets of points to the segment in each iteration.

Furthermore, the resulting region is refined by comparing the color of the points with previously stored skin colors. One study [23] accelerates this process with the use of a black band around the wrist.

Another alternative is to look for the detection of certain parts of the hand, usually the fingertips or palm. Liang et al. [24], for example, establish a series of sub-regions that together make up the hand. Then, to identify each region, they record the ratio of each pixel depth in comparison with its neighbors and the temporal relationship with previous frames. Suau et al. [27] attempt to detect the position of the fingertips. In this case, the proposed method uses ORD (Oriented Radial Distribution) to determine the positions of the hand and fingertips. ORD is based on dividing the point

cloud of the hand into sectors that together form a circle. The average radius of the points in each sector is used to measure the curvature of the points belonging to it. Dominio et al. [26] perform distance measurements on the hand region to establish the position of the palm, fingers, and forearm. The PCA algorithm (Principal Component Analysis) is used on the set of points to define a coordinate system on which to perform the measures.

We can also utilize existing methods to determine the skeletal body position. The idea here is first to determine the position of the skeleton in the body and to take from it the approximate position of the hands.

One study [28] uses the functions of existing libraries for this purpose. Another [29] similarly uses an algorithm to maximize the geodesic distances on a surface in order to find the projecting parts of the body, which are subsequently classified (hands, feet, head). The significant drawback of this approach is the need for most of the body to be in front of the sensor, even if only hand position is desired.

5. Tracking

Tracking methods are designed to follow hand movement over time. While segmentation methods give us hand regions, tracking permits us to establish how these regions move over time.

Due to the speed of hand movement, in many real-time applications it is not feasible to use the same method of segmentation for each frame of video. Put simply, most algorithms are not fast enough when not run on specialized hardware.

A common alternative in applications is to try to predict the movement of the hands. To do this, we store the position of the hand region (usually the center or some particular point in the region), as well as the vector speed of the region. With this information, we can "deduce" the possible position of the hand in the next frame and thus accelerate the process of segmentation. In these cases, an initial segmentation step often must be performed with the hands at rest to determine the initial position of the hand region. Chen et al. [30] present an example of this approach. Here, the potential new position of the center of the hand is predicted using a velocity vector and the region is subsequently segmented using a growth technique based on Euclidean distances. In contrast, another study [25] employs a Kalman filter to establish the position in each frame.

In order to speed up processing, one study [31] associates an ellipsoid with the space occupied by the hand. This is then used to facilitate hand position tracking over time since it is assumed that the center of the hand in each new frame will always be within the ellipsoid.

In more traditional applications that only make use of color images, it is common to use filters for recognition tasks. Such techniques are considered an alternative even in the case of depth sensors. Tang et al. [20], for example, use a Kalman filter for hand tracking. Park et al. [32] start with the creation of an image corresponding to the cumulative difference of five depth images. Connected regions are obtained from the image, which are refined until the region of hand movement is obtained by means of a regression method, with the assumption that the hand is the closest object to the camera. Finally, a Kalman filter is used for tracking.

6. Classification

Once you have a region that tentatively corresponds to a hand (the segmentation process), classification techniques can help establish whether this region truly is the hand and can determine whether it corresponds to a particular pose. For hand recognition, classification techniques are usually used to identify whether a set of points corresponds to a set of predefined positions.

The matter of hand classification is particularly complex due to the lack of rigidity of hands and because each segment can take on a variety of positions and orientations. Indeed, a glance at the literature on hand recognition from recent years shows a variety of classification methods, with no single method standing out.

Among the alternatives, the contours of the hand are a characteristic principally used for classification. Tang et al. [20] create a function whose input variables are determined using the contours of the hand and the depth values thereof. The contours of the hand are similarly used by Ren et al. [33] and are then represented as a time series curve. The classification is based on the similarity of the curve with pre-established templates.

Some studies consider the position or orientation of the fingers for classification, especially in applications where the hand extends its fingers, and there are no significant changes in hand orientation in front of the sensor. One study [21] tries to resolve the problem of orientation by obtaining the hand region, then rotating the points until the palm becomes parallel to the plane of the image. They thus get a descriptor based on geometric properties, relatively independent of orientation. Another study [17] uses distance measurements to extract geometrical characteristics and create a model of the hand, through the utilization of an "initial pose." The classification is carried out by identifying the geometric positions of the fingers and joints, which are compared with a set of pre-modeled gestures.

It is common for a classification method to require a set of previously stored descriptors with which to compare input data. Dominio et al. [26], for example, generate a histogram representing the curvature of the fingers with respect to the palm, to be used as a descriptor. Subsequently, they use a Support Vector Machine algorithm to classify the histogram in comparison with those previously stored.

Using a similar approach, after extracting the hand region, Tang [34] generates descriptors from the radial histogram and the Speeded Up Robust Features (SURF) method. These descriptors are then compared to a set of 2,901 previously stored descriptors, all in order to determine if the hand is in the open or closed position.

Another study [35] employs a method called *Finger-Earth Mover's Distance* (FEMD) for the classification. For this, a function is first generated that describes the contours of the hand with respect to a fixed point within it, which is then used as a descriptor for classification. A further study by the same authors [36] follows this up by comparing this method with others that make use of geometric features. Another paper [37] also makes use of the contours of the hand as a descriptor, but in this case the contours are determined by a growth algorithm starting from one point in the hand and then growing, taking into account the distance between neighboring pixels in the depth map.

In many cases, the amount of pre-stored data must be high to ensure that the used algorithm classifies correctly, especially if the problem is approached as an optimization problem. Due to the large amount of data necessary, some studies have opted to create a 3D digital model of a hand, which they use to generate the pre-stored data.

Having a digital model facilitates obtaining points of depth or color with projections commonly used in computer graphics. One of these cases [38] approaches the classification problem as an optimization problem. Their proposed method consists of several steps: removing noise in the image, estimating hand orientation, selecting candidates for poses, and determining the final pose by solving an optimization problem. A regression method (Hough forest) is used to estimate the orientation of the hand, and a new regression method using data generated from an articulated 3D hand model is used to select candidates.

A less common approach is addressed by Trindade et al. [39], using an additional hardware device to facilitate hand gesture recognition. They obtain the points of the hand region with a depth sensor and additional hardware to determine the angle of the hand. The classification is not performed directly on the points in the region, but rather defines voxels (easily-represented volume elements that

together form a three-dimensional space), which are compared with voxels from previously stored positions by the ICP (Iterative Closest Point) algorithm.

7. Unconstrained Poses

Some studies have focused on a more complex problem of determining unconstrained hand poses. The idea is to obtain representation in the form of segments and joints (the skeleton) that describe the position and orientation of the palm and the different segments of the fingers, regardless of the position taken by the hand. The objective is equivalent to recognizing the whole skeleton, as performed in applications such as games. Most applications, therefore, use a 3D digital model of the hand, either to generate a set of tentative poses, to validate whether a pose is valid, or to generate variables used by an optimization method, among others.

One study [40] uses Random Forests to detect different regions of the hand, and then to create a skeleton by joining the various regions. The method used is similar to the method proposed in other publications to determine the skeleton of the entire body. Because this requires a relatively large training set, they generate the different poses required with a 3D model.

Another study [41] uses a relatively simple 3D model, which is projected onto the depth map of a hand silhouette. The similarity between the projection and the silhouette is used to estimate the probability of belonging to the hand points by applying a particulate filter called Sampling Importance Resampling.

Schröder et al. [18] use Inverse Kinematics to determine hand position from the corresponding point cloud. The idea is to minimize an objective function that represents the correspondence between points on the surface with the position of each joint. To expedite the process and to deal with potential incomplete data entry, they reduce the number of possible poses (variables of the function to be minimized) using the synergies derived from the analysis of a set of previously stored data.

The use of inverse kinematics is not very common in the studies reviewed. It is much more common to use an optimization method on an articulated 3D hand model. These cases attempt to align the projection model with the observed image. The model also serves to impose restrictions on the positions that the hand can take.

De La Gorce et al. [42] give us an example of a solution that tries to determine hand position with an articulated 3D hand model. The angles used in the 3D model are determined during execution of approximation methods. Unlike other

methods, this method involves a fairly detailed 3D model, which also permits the use of geometric features, textures, and shading, values that assist in the estimation of the objective function. Oikonomidis et al. likewise [43] seek to minimize the discrepancy between the observed hand region and the potential pose generated by using a 3D model of a hand (modeled with cylinders and ellipsoids). The optimization method used here is an evolutionary method called Swarm Optimization.

Another study by Oikonomidis et al. [19], similar to that above [43], presents the problem of recognition as an optimization problem. But in this case the authors attempt to find the pose of a hand holding an object, and thus some parts may be obscured by the object. They include the parameters of the held object, which must be known, in the optimization function. Particle Swarm (PSO) is again used for optimization and a 3D model composed of spheres is used for performing calculations of collision between the hand and the held object. The use of spheres allows for quicker calculation of collisions from different parts of the model (albeit without the same precision). A further study by the same authors [44] extends this study [43] in order to address the problem of recognizing the pose of both hands interacting in an unconstrained manner. Again the Swarm Optimization method is used to minimize the discrepancy between what is observed and the regions generated from pre-defined 3D models.

Qian et al.'s study [23] is perhaps, of those reviewed, the study that most advances progress towards pose recognition real time. As in other studies, they define a function to be minimized, which seeks to reduce the difference between what is captured and a pre-defined model. However, the 3D model used is simpler than in other studies, since, as with Oikonomidis et al. [19], only spheres are used to build the model. Additionally, they do not use every part of the hand region, but rather obtain 256 random samples. For the minimization algorithm, they employ a method that combines the evolutionary algorithm, Particle Swarm, with the best known, Iterated Closed Point (ICP). To facilitate the re-initialization in each frame, they use geometric characteristics to establish the possible location of the fingertips.

8. Applications

Current hand recognition applications do not appear as varied as classification methods. One review [4] of 37 papers, although not directly related to hand recognition, includes a section that classifies the main applications for gesture recognition, underlining its use in systems of interaction, robotics, and symbol recognition in sign language.

For example, one study [28] presents an application for interacting with objects in a 3D scene is presented. The application makes use of existing libraries to determine the skeleton (pose) of the person. The variation in arm position of the skeleton is then used to determine the input command. The skeleton can also identify the location of the hands, which are evaluated to determine whether or not they are closed, in order to improve interaction. With a similar purpose, another study [45] creates an interaction system for manipulating a digital model (with selection, rotation, translation, and magnification). In this case, both hands are used for the interaction. The system is based on the classification of input data using a reduction method called Average Neighborhood Margin Maximization (ANMM), which is fed with color and depth information about the hand region.

Another study [46] seeks to synchronize the movement of a robotic arm with the movement of an arm captured by the Kinect device. The CAMSHIFT algorithm is used to track the arm and Markov chains (Hidden Markov Model) in order to classify the position of the arm within one of four patterns of movement.

Li [47] presents research using the Kinect to recognize a reduced set of North American Sign Language gestures. The classification is based on the detection of finger position using geometric properties. Oszust and Wysocki [48] develop a system to recognize poses from Polish sign language, using a classification algorithm based on PCA.

Another study [49] recognizes four hand movements recorded in an XML format called SiGML. Recognition is performed by capturing two frames of video. For each frame, the X,Y coordinates are determined from the center of the hand region, which are subsequently used to classify the movement of the hand. The classified movements are clearly linear.

A different application of interaction [50] uses recognition of finger position on both hands to play a virtual instrument (such as guitar or piano). The recognition is based on the geometric characteristics of the hand, especially the fingers.

A further study presents a system with three sensors, including a depth camera, to detect gestures made by the driver of an automobile. The idea is that the driver can perform gestures in front of the camera with one hand to interact with the car. In this case, the driver's gestures are classified by putting depth and color information into a previously trained neural network (DNN).

9. Conclusions

Hand recognition using depth maps is a problem that has been heartily investigated in recent years. Although it is possible to find a variety of solutions for specific applications, none of the studies reviewed were able to take advantage of low-cost devices to determine real-time two-hand unconstrained poses performing quick movements. Certainly in specific cases researchers were able to track the movement of the hands or determine their pose, but the techniques are limited in their application.

Hand region recognition in many cases still lies in the use of color information (which can be affected by lighting) or assumes that hands represent the region closest to the capture device. This is suitable for applications involving human-machine interaction, but can be inconvenient for other applications.

Hand position tracking is usually limited to only one hand and requires an initial calibration or that the hand is in a fixed pose during movement. In general the speed of the input devices, coupled with the performance of the segmentation algorithms, make it impractical to determine the position and orientation of the hand in each frame without using hand information from previous frames. This generally causes tracking algorithms to be affected by sudden movements or by the appearance of new objects in the scene (like a new hand), cases in which information from previous frames is not as useful.

Although we have reviewed some algorithms for detecting unconstrained skeletal position, we can see that most applications still make use of classification techniques among a limited set of poses. This may be because the performance of traditional algorithms is not the best for commonly used hardware. Among the papers evaluated, only the most recent contain proposals that may become useful for real-time applications with consumer hardware. However, these proposals are focused on recognizing one hand that is facing the camera, unobstructed, and not interacting with other objects.

The above limitations are reflected in the slim variety of recognition applications, which suggest greater progress in applications of interaction, either for virtual environments or the control of devices and robots. Moreover, most applications (interactive or not) work with a small set of pre-established positions when the recognition of the position of one or both hands is required.

Finally, we should highlight the difficulty of comparing the techniques used in the reviewed studies. Indeed, they do not make use of a similar set of input data, since each paper performs their tests with a different set of data, captured

under a variety of conditions and with a variety of types of devices.

Bibliography

- [1] A. Jana, [Kinect for windows SDK programming guide](#), Birmingham, UK: Packt Publishing Ltd, 2012.
- [2] Z. F. Z. Feng, B. Y. B. Yang, Y. Z. Y. Zheng, Z. W. Z. Wang y Y. L. Y. Li, «[Research on 3D Hand Tracking Using Particle Filtering](#),» de *Fourth International Conference on Natural Computation*, Washington, 2008.
- [3] S. Falahati, [OpenNI Cookbook](#), Birmingham, UK: Packt Publishing Ltd, 2013.
- [4] W. Engel, ShaderX3 : [Advanced Rendering with DirectX and OpenGL](#), Hingham, MA, USA: Charles River Media / Cengage Learning, 2004.
- [5] M. Corporation, [Kinect for Windows](#), 20145.
- [6] A. Corporation, [Xtion Pro](#), 2014.
- [7] M. Bhuyan, D. Neog y M. Kar, «[Hand pose recognition using geometric features](#),» de *National Conference on communications (NCC)*, 2011.
- [8] C.-P. Chen, Y.-T. Chen, P.-H. Lee, Y.-P. Tsai y S. Lei, «[Real-time hand tracking on depth images](#),» de *Visual Communications and Image Processing (VCIP)*, Tainan, 2011.
- [9] L. Chen, H. Wei y J. Ferryman, «[A survey of human motion analysis using depth imagery](#),» *Pattern Recognition Letters*, vol. 34, n° 15, pp. 1995-2006, #nov# 2013.
- [10] L. Cruz, D. Lucio y L. Velho, «[Kinect and RGBD Images: Challenges and Applications](#),» de *2012 25th SIBGRAPI Conference on Graphics, Patterns and Images Tutorials*, Ouro Preto, 2012.
- [11] M. De La Gorce, D. J. Fleet y N. Paragios, «[Model-based 3D hand pose estimation from monocular video](#),» *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, n° 9, pp. 1793-1805, 2011.
- [12] F. Dominio, M. Donadeo, G. Marin, P. Zanuttigh y G. M. Cortelazzo, «[Hand gesture recognition with depth data](#),» de *Proceedings of the 4th ACM/IEEE international workshop on Analysis and retrieval of tracked events and motion in imagery stream (ARTEMIS)*, Barcelona, 2013.
- [13] V. Frati y D. Prattichizzo, «[Using Kinect for hand tracking and rendering in wearable haptics](#),» de *IEEE World Haptics Conference, WHC 2011*, Istanbul, 2011.
- [14] Y. F. A. Gaus y F. Wong, «[Hidden Markov Model-Based Gesture Recognition with Overlapping Hand-Head/Hand-Hand Estimated Using Kalman Filter](#),» de *Third International Conference on Intelligent Systems Modelling and Simulation*, Kota Kinabalu, 2012.
- [15] J. Han, L. Shao, D. Xu y J. Shotton, «[Enhanced computer vision with Microsoft Kinect sensor: A review](#),» *IEEE Transactions on Cybernetics*, vol. 43, n° 5, pp. 1318-1334, 2013.
- [16] R. Hartanto, «[Real Time Hand Gesture Movements Tracking and Recognizing System](#),» de *2014 Electrical Power, Electronics, Communications, Controls, and Informatics Seminar (EECCIS)*, Malang, 2014.
- [17] H. S. Hasan y S. A. Kareem, «[Human Computer Interaction for Vision Based Hand Gesture Recognition: A Survey](#),» de *International Conference on Advanced Computer Science Applications and Technologies (ACSAT)*, Koala Lumpur, 2012.
- [18] M.-h. Hsu y T. K. Shih, «[Real-Time Finger Tracking for Virtual Instruments](#),» de *7th International Conference on Ubi-Media Computing and Workshops*, Ulaanbaatar, 2014.
- [19] Y.-c. H. Y.-s. Huang Chen, «[An Occlusion-Resolving Hand](#)

- [Tracking Method](#),» de *7th International Conference on Ubi-Media Computing and Workshops*, Ulaanbaatar, 2014.
- [20] C. Keskin, F. Kirac, Y. E. Kara y L. Akarun, «[Real Time Hand Pose Estimation using Depth Sensors](#),» de *IEEE International Conference on Computer Vision Workshops*, Barcelona, 2011.
- [21] K. H. Kim, D. U. Jung, S. H. Lee y J. S. Choi, «[A hand tracking framework using the 3D active tracking volume](#),» de *FCV 2013 - Proceedings of the 19th Korea-Japan Joint Workshop on Frontiers of Computer Vision*, Incheon, 2013.
- [22] A. Kurakin, «[A real time system for dynamic hand gesture recognition with a depth sensor](#),» de *20th European Signal Processing Conference (EUSIPCO 2012)*, Bucharest, 2012.
- [23] J. Lee, H. Gul, H. Kim, J. Kim y H. Kim, «[Interactive manipulation of 3D objects using Kinect for visualization tools in education](#),» de *13th International Conference on Control, Automation and Systems (ICCAS 2013)*, Gwangju, 2013.
- [24] Y. Li, «[Hand gesture recognition using Kinect](#),» de *IEEE 3rd International Conference on Software Engineering and Service Science (ICSESS)*, Beijing, 2012.
- [25] H. Liang, J. Yuan y D. Thalmann, «[Parsing the hand in depth images](#),» *IEEE Transactions on Multimedia*, vol. 16, n° 5, pp. 1241-1253, 2014.
- [26] P. Molchanov, S. Gupta, K. Kim y K. Pulli, «[Multi-sensor System for Driver's Hand-Gesture Recognition](#), » *IEEE International Conference on Automatic Face and Gesture Recognition (FG2015)* 2015, .
- [27] L. T. Nguyen, C. D. Thanh y T. N. Ba, «[Contour Based Hand Gesture Recognition Using Depth Data Hand Gesture recognition](#),» *Advanced Science and Technology Letters*, vol. 29, n° Sip, pp. 60-65, 2013.
- [28] C. M. Oh, M. Z. Islam y C. W. Lee, «[Articulated hand tracking using key poses driven particle filtering](#),» de *ICCET International Conference on Computer Engineering and Technology, Proceedings*, Chengdu, 2010.
- [29] I. Oikonomidis, N. Kyriazis y a. Argyros, «[Tracking the articulated motion of two strongly interacting hands](#),» *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1862-1869, 2012.
- [30] I. Oikonomidis, N. Kyriazis y A. Argyros, «[Efficient model-based 3D tracking of hand articulations using Kinect](#),» *Proceedings of the British Machine Vision Conference 2011*, pp. 101.1--101.11, 2011.
- [31] I. Oikonomidis, N. Kyriazis y A. a. Argyros, «[Full DOF tracking of a hand interacting with an object by modeling occlusions and physical constraints](#),» *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2088-2095, 2011.
- [32] M. Oszust y M. Wysocki, «[Recognition of signed expressions observed by Kinect Sensor](#),» *2013 10th IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2013*, pp. 220-225, 2013.
- [33] M. Panwar, «[Hand gesture recognition based on shape parameters](#),» de *2012 International Conference on Computing, Communication and Applications*, Dindiquil, Tamilnadu, 2012.
- [34] S. Park, S. Yu, J. Kim, S. Kim y S. Lee, «[3D hand tracking using Kalman filter in depth space](#),» *EURASIP Journal on Advances in Signal Processing*, vol. 2012, n° 1, p. 36, 2012.
- [35] C. Plogemann, V. Ganapathi, D. Koller y S. Thrun, «[Real-time identification and localization of body parts from depth images](#),» *2010 IEEE International Conference on Robotics and Automation*, pp. 3108-3113, #may# 2010.
- [36] C. Qian, X. Sun, Y. Wei, X. Tang y J. Sun, «[Realtime and Robust Hand Tracking from Depth](#),» *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014.
- [37] K. Qian, J. Niu y H. Yang, «[Developing a Gesture Based Remote Human-Robot Interaction System Using Kinect](#),» *International Journal of Smart Home*, vol. 7, n° 4, pp. 203-208, 2013.
- [38] J. Raheja, R. Shyam, U. Kumar y P. Prasad, «[Real-Time Robotic Hand Control Using Hand Gestures](#),» de *Second International Conference on Machine Learning and Computing (ICMLC)*, Bangalore, 2010.
- [39] Z. Ren, J. Meng y J. Yuan, «[Depth Camera Based Hand Gesture Recognition and its Applications in Human-Computer-Interaction](#),» de *IEEE International Conference on Information Communication and Signal Processing*, Singapore, 2011.
- [40] Z. Ren, J. Yuan y Z. Zhang, «[Robust hand gesture recognition based on finger-earth mover's distance with a commodity depth camera](#),» *Proceedings of the 19th ACM international conference on Multimedia*, pp. 1093-1096, 2011.
- [41] Z. Ren, J. Yuan, J. Meng y Z. Zhang, «[Robust part-based hand gesture recognition using kinect sensor](#),» *IEEE Transactions on Multimedia*, vol. 15, n° 5, pp. 1110-1120, 2013.
- [42] P. S. S. Shaikh, S. L. Dhebe, P. D. Zambare, A. D. Jivanwal y P. P. Luniya, «[Human Computer Interaction \(Robot Handling\) using Hand Gesture Recognition](#),» *International Journal of Engineering Research and Technology (IJERT)*, vol. 3, n° 2, pp. 712-715, 2014.
- [43] P. Sonwalkar, T. Sakhare, A. Patil y S. Kale, «[Hand Gesture Recognition for Real Time Human Machine Interaction System](#),» *International Journal of Engineering Trends and Technology (IJETT)*, vol. 19, n° 5, pp. 262-264, 2015.
- [44] J. Suarez y R. R. Murphy, «[Hand gesture recognition with depth images: A review](#),» *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*, Paris, 2012.
- [45] P. Srilatha y T. Saranya, «[Advancements in Gesture Recognition Technology](#),» *IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)*, vol. 4, n° 4, pp. 1-7, 2014.
- [46] X. Suau, M. Alcoverro, A. López-Méndez, J. Ruiz-Hidalgo y J. R. Casas, «[Real-time fingertip localization conditioned on hand gesture classification](#),» *Image and Vision Computing*, vol. 32, n° 8, pp. 522-532, 2014.
- [47] C. Tang, Y. Ou, G. Jiang, Q. Xie y Y. Xu, «[Hand tracking and pose recognition via depth and color information](#),» de *ROBIO 2012 - Conference Digest IEEE International Conference on Robotics and Biomimetics*, Guangzhou, 2012.
- [48] M. Tang, «[Recognizing Hand Gestures with Microsoft's Kinect](#),» *Computer*, vol. 14, pp. 303-313, 2011.
- [49] P. Trindade, J. Lobo y J. P. Barreto, «[Hand gesture recognition using color and depth images enhanced with hand angular pose data](#),» de *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, Hamburg, 2012.
- [50] M. Van Den Bergh y L. Van Gool, «[Combining RGB and ToF cameras for real-time 3D hand gesture interaction](#),» de *2011 IEEE Workshop on Applications of Computer Vision, WACV 2011*, Kona, HI, 2011.