Sensors & Transducers, Vol. 196, Issue 1, January 2016, pp. 1-6

Sensors & Transducers

http://www.sensorsportal.com

Gesture Based Control Using Windows Kinect and Robot Operating System

KUNAL KAUSHIK, K. SRIRAM and M. MANIMOZHI
School of Electrical Engineering, VIT University, Vellore, 632014, Tamil Nadu, India
Tel.: +91-9159897511
E-mail: kunal.kaushik2012@vit.in

Received: 7 November 2015 /Accepted: 7 December 2015 /Published: 30 January 2016

Abstract: This paper deals with using a common gaming sensor Kinect in order to control a wheel chair using hand gestures to help speech disabled person. Lately there have been many attempts to make wheel chairs voice controlled or analog control, but gestures are natural way to communicate having a universal understandable meaning. Using gestures, we can control the speed and the direction of a wheelchair in a more intuitive way as the gestures significantly describe the intensity of the action desired. Various human body organs can be used to give input to the system. *Copyright © 2015 IFSA Publishing, S. L.*

Keywords: Kinect, Skeleton tracking, Gesture recognisation, Robot Operating System, Wheelchair,depth tracking.

1. Introduction

The hand gestures are the most intuitive way of human communication, which can covey message in a short and precise manner which is universal in nature, Now a days due to increase in robot human interaction and machine being so close in vicinity of humans, the need of a new human- machine interaction language which is more natural and easy to communicate is aroused, The earlier attempts were focused on the color detection techniques and voice recognition which had the limitations of fluctuations due to light conditions and need of frequent calibrations in case of color based approach and the difficulty of addressing the intensity of instructions in case of voice controlled models. The low accuracy and tough compositionality were also major drawbacks for these models.

Among the various techniques available for gesture recognition like image processing prove to be computationally very intensive and recognition using an array of ultrasonic sensors might not be accurate for particular complex gestures.

Hence the Kinect provides a cost effective, less computationally intense and precise solution, The Kinect is a sensor developed by Microsoft which includes a RGB plus Depth camera, which prepares a density map of its surroundings in 3-D in form of colored point clouds, where each points have as many as 300,000 points. This data can obtain the real time position of a human body skeleton in 3-D space. Later the 3-D input data is fed to ROS for algorithmic computations and hence giving inputs for gesture recognition

2. Kinect & It's Working

A Microsoft Kinect sensor includes a very high resolution RGB and depth sensing which is lately becoming a trend to recognize gestures and work with human computer interaction. It implements various

tasks such as object detection and reorganization, tracking an object. It also helps in human activity analysis, hand gesture analysis and 3D mapping. It can be used to detect and tell apart different varieties of objects. Thus the Kinect was found to be an effective tool action recognition.

The Kinect camera includes an infrared projector, the color camera, and the Infrared detector. The depth sensor is made up of the IR projector combined with the IR camera, which is a monochrome CMOS sensor. The IR projector is a single IR laser that passes through a number of diffraction gratings to be turned into set of IR dots of a determined pattern.

The IR camera, emitter and different dots make up the vertices of a triangle and simple geometry is used to determine the depth. If a dot found out in an image matches with a dot in the predetermined pattern, rebuild it in 3D using triangulation. Because the pattern the dots make is relatively random, the match between the IR image obtained and the projector pattern can be done in a relatively easy way by comparing small neighborhood's using, for example, normalized cross correlation.

In skeletal tracking, body of a human is represented using different joints representing body parts such as head, neck, shoulders, and arms. Each joint is in turn represented by using its 3D coordinates. The aim is to find out all the three parameters of these joints in real time to facilitate continuous interaction and with as less computation resources as possible. Xia et al. proposes a model based algorithm to detect humans by usage of the depth maps produced using the Kinect sensor.

Sung et al. extract the data obtained from the skeleton and uses a supervised learning to recognize activities from RGB and depth images obtained using a Kinect sensor. He used a supervised learning approach, where he extracts features from the skeleton joints, its positions, its orientation and arm movements, he developed a model from labeled instances. Xia et al. proposed an algorithm that recognizes the actions using a histogram of joints in 3D extracted from depth images. Theresia et al proposes an algorithm that recognizes activities with the Kinect using a logic-based approach, he developed a logic model with labeled instances. Wang et al. actionlets ensemble approach to recognize activities, according to Wang, the actionlet is defined as a conjunctive (or AND) structure on the base features, one base feature is defined as a Fourier Pyramid period of one joint of the skeleton. Ong et al presents an approach to extract features from Kinect based on human range of movement. Ong applied K-means clustering on the features extracted based on human range of movement, giving the results of improvement clustering performance. Maierdan et al. proposes a HMM approach for recognize human activities. Maierdan discusses two points in his approach, the first is the HMM application of HMM to recognize human activities, the second is the effect of K-means and fuzzy C-means. Rabiner et al. reviews the theoretical aspects of HMM and show how they have been applied to problems in machine recognition speech.

All these are described works that uses the Kinect skeleton joints tracking to recognize the activity. Some of these papers use a supervised learning algorithm. In long term, the intelligent systems need to learn new activities that are not labeled, that is one of the many reasons there is an interest to decipher human activity discovery using unsupervised learning. Some of these works use K-means clustering to discover human activities and use HMM to recognize the activities.

The approach of this work is to discover human activities when they are seated, for this purpose, we are going to record data of a seated skeleton, the data recollected we are going to pass it to K-means algorithm because of it is an unsupervised learning algorithm, and we want to use it to discover human activities, also we are going to use HMM to recognize what activity the person is doing.

3. Skeleton Tracking

Before application of the action recognition approach, the depth maps obtained by using the Kinect sensor are fed to a skeleton-tracking algorithm. The depth maps mentioned were acquired using the OpenNI. The OpenNI high-level skeleton-tracking module is used for the tracking joints of a person's body. More specifically, the OpenNI tracker detects the position of the following set of joints in the 3D space $G = \{gi, i \in [1, I]\} \equiv \{T$ orso, Neck, Head, Lef t shoulder, Lef t elbow, Lef t wrist, Right shoulder, Right elbow, Right wrist, Left hip, Left knee, Left foot, Right hip, Right knee, Right foot$\}$. The position of joint gi is denoted by vector $pi(t) = [x\ y\ z]\ T$, where t denotes the frame for which the joint position is located and the origin of the orthogonal XY Z co-ordinate system is placed at the center of the Kinect sensor. The OpenNI skeleton-tracking module requires an initial user calibration in order for it to find out approximately several body characteristics of the person. This is done by performing a particular pose in front of the Kinect at the beginning of usage. In recent versions of OpenNI, this initial pose is eliminated by the 'auto-calibration' mode which enables user calibration without implying the person to stay in any particular calibration pose. Since a calibration pose was captured for the employed dataset, the OpenNI's 'auto-calibration' mode is used not in this work**.** Fig. 1 shows he initial calibration pose required to identify the skeleton.

The experimental evaluation revealed that the skeleton-tracking algorithm being applied here is relatively robust. The position of the joints is usually detected accurately, although there were cases where the tracking was not completely correct. An example of the latter is the inaccurate detection of the joint positions when there is a very sudden and intense movement. (e.g. leg movement while a round house kick).

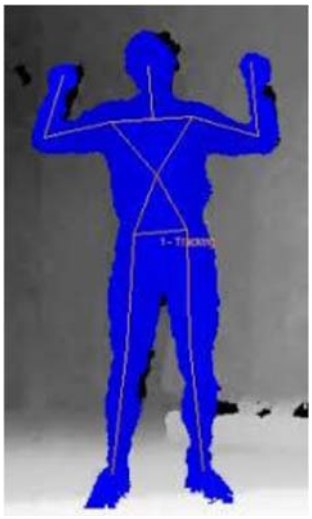

Fig. 1. The initial calibration procedure pose.

4. Depth Calculations

To calculate depth at a single point let us assume a point k. As the distance of speckle 2 with camera will be changed, its position in the focal plane will be changed in x direction. This change in distance d is called the disparity of that particular marker. This change in disparity is used to find the depth of the point.

Here, C is the camera that detects the markers, L is the laser source, D is the 3-D disparity, Z_0 is distance of the camera from the Reference plane, Z_k is the distance of the point k form the camera. Fig. 2 depicts the distance calculation for a single marker, with triangulation.

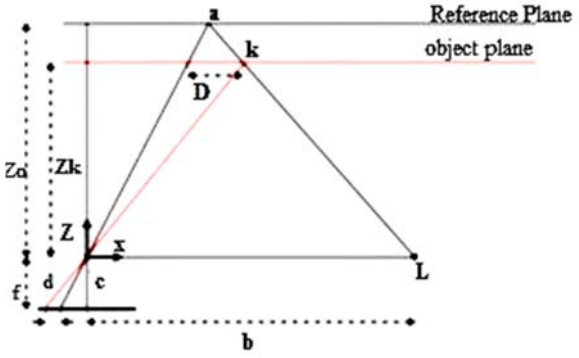

Fig. 2. Depth calculations.

By similarity of triangles,

$$\frac{D}{b} = \frac{Z_0 - Z_k}{Z_0}$$

$$\frac{d}{f} = \frac{D}{Z_k}$$

By solving the above

$$Z_k = \frac{Z_0}{1 + \frac{Z_0}{f} d}$$

The same process is repeated for all the markers. Once, the distances of all the markers are triangulated, then a disparity map, shown below is produced. Fig.3 is an example of obtained disparity map.

Once the depth cloud is attained, an open-source software called NITE is used to interpret the point-clouds, that look similar to human figure, into human skeletons. Hence, the skeletal information is attained. The NITE toolbox publishes 3-D position information about various body parts, like hands, elbows, knees, etc. The tracked skeleton looks similar to the image below. In our case, we take the 3-D positional data of the right hand and train the algorithms.

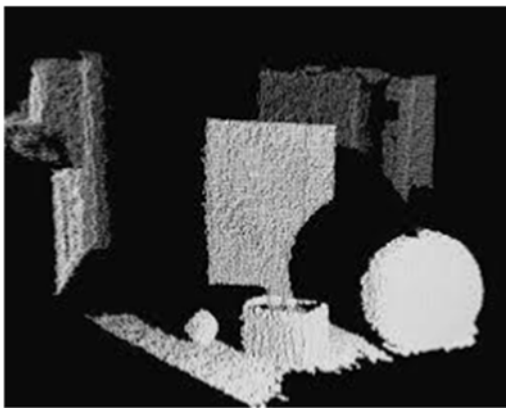

Fig. 3. Disparity Map.

5. Action Recognition

Action recognition Action recognition can be further divided into three subtypes.

5.1. Pose Estimation

The aim of this step is to estimate an updated orthogonal basis of vectors in real time for every frame t that represents the person's pose. The calculation of the later is based on the assumption that the orientation of the person's torso is the quantity that is most characteristic of the subject during the execution of any action and can be used as reference for the same reason. For pose estimation, the position of three joints is taken into consideration: Left shoulder, Right shoulder and Right hip. These make up the joints around the torso area, whose position relative to each other remains unchanged to the greater extent during the execution of any action. The reason behind the consideration of the three before mentioned joints, instead of directly approach to estimating the position of the torso joint and its normal vector, is to obtain a more accurate estimation of the person's pose. It must be seen that the Right hip joint was preferred to the

obvious Torso joint selection. This was done so that the orthogonal basis of vectors to be estimated from joints with bigger distances in between that will be more likely to lead to obtain more accurate pose estimation. However, no significant deviation in action recognition performance was found when the Torso joint was used instead.

5.2. Action Representation

For getting efficient action recognition, a proper representation is required that will handle the differences in appearance, human body type and execution of actions among the individuals satisfactorily. For that the angles of the joints and relative position are used, which proved to be more discriminative than using normalized coordinates of the joints. Building on the fundamental idea of the previous section, all the angles are computed using the Torso joint as reference, i.e. the Torso joint position is used as the origin of the spherical coordinate system. For computing the proposed action representation, only a subset of the supported joints is used. This is because the trajectory of some joint contains redundant or noisy information mainly. To this end, only the joints0 corresponding to the upper and lower body limbs were taken into account after experimental evaluation, namely the joints Left shoulder, Left elbow, Left wrist, Right shoulder, Right elbow, Right wrist, Left knee, Left foot, Right knee and Right foot. The velocity vector is approximated by the displacement vector between two successive frames, i.e. vi(t) = i(t)−pi(t−1). The estimated spherical angles and angular velocities for frame t constitute the frame's observation vector. Collecting the computed observation vectors for all frames of a given action segment forms the respective action observation sequence h that will be used for performing HMM-based recognition, as will be described in the next part.

5.3. HMM Based Recognition

Markov Models is stochastic model describing the sequence of possible events in which the probability of each event depends only on the state attends in the previous event. This model is too restrictive to be applicable to current problem and thus the concept of Markov model is extended to form Hidden Markov Model (HMM). HMM is doubly embedded stochastic process with the underlying stochastic process i.e. not observable (it is Hidden) but can only be observed through set of stochastic process that produce the sequence of observations. HMMs are employed in this work for performing action recognition, due to their suitability for modeling pattern recognition. In particular, a set of J HMMs is employed, where an individual HMM is introduced for every supported action aj. Each HMM receives as input the action observation sequence h (as described above) and at the evaluation stage returns a posterior probability P (aj|h), which represents the observation sequence's fitness to the particular model. The developed HMMs were implemented using the software libraries of Hidden Markov Model Toolkit (HTK).

6. Setup

The current project being discussed has a Kinect sensor is placed in front of a wheelchair and is attached to a laptop running on LINUX and to the laptop attached is the Arduino board to control the motors of a wheelchair. The Arduino board is precoded to perform certain actions based on the inputs received from the laptop. The Kinect sensor captures the depth and RGB images and sends them to the laptop for processing. The laptop is upon receiving the images employs the above discussed algorithms to track the joints and determines the lengths which the right and left hands rise or fall. It then sweeps the values of the obtained lengths into limits both above and below which the Arduino can understand and move the motors accordingly. ROS system is used for the interface between laptop and the Arduino.

7. Process

The process is initialized by standing infront of the Kinect in a particular pose in our experiment but auto calibration can be used when used for disables person in the chair. Fig. 4 here shows the process flow model of Skeletal Frame recognition by Kinect.

Initially Infrared Rays (IR) are emitted from the IR transmitter of Kinect sensor. Emitted rays are been received by Kinect receiver. Since the Kinect sensors monitoring for the human joints, it does not show any significant data until the human joints are recognized. If any object other than the skeleton joints are recognized it discards the frame and restarts the scanning of the next frame until joints are recognized. Black frame indicates that neither the object is been detected nor the skeletal joints are detected. This kind of image results into blackening of frame and the white spots on the black frame are due to noises present in the environment. Once the Joints are been recognized/detected Kinect uses HMM algorithm for joint estimation and predicts the future movements.

Fig.5 is the process flow model of Stage 2 which includes Calculation and implementation. This recognized joint information is read by the laptop and swept into limits and is sent to Arduino using Robot Operation System (ROS). There the signals are converted into PWM pulses by the programmed PWM pulse generator present on Arduino board. The generated PWM pulses which serve as input to the servo motors, are been made to perform the desired movement according to the action that has been captured. Since this is real time the entire process is been continuously repeated for each frame.

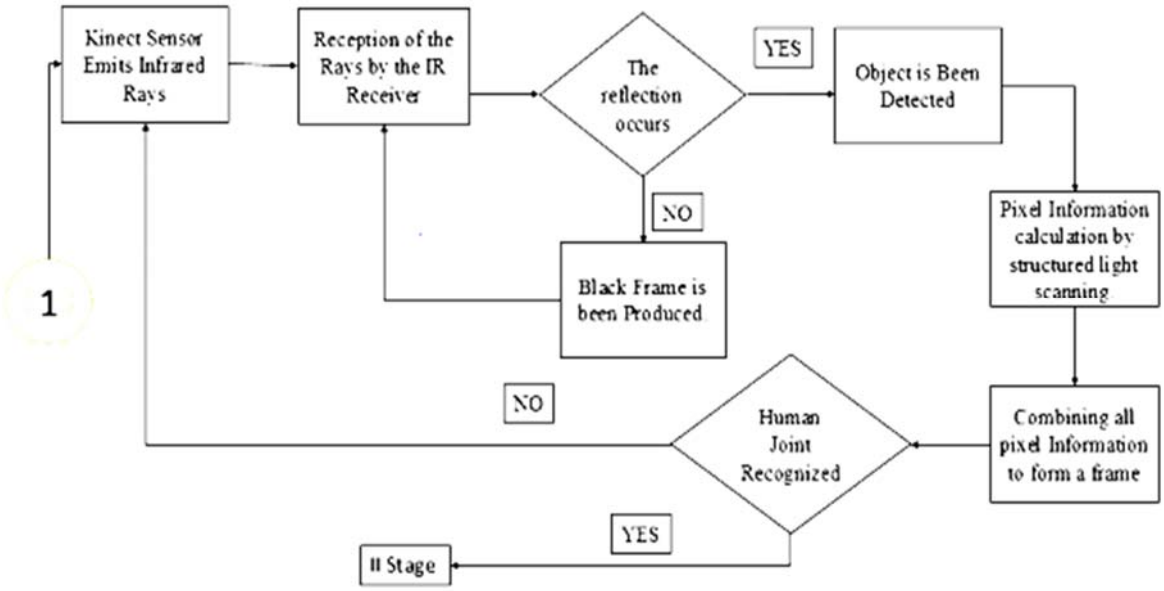

Fig. 4. Stage 1. Skeletal Frame recognition by Kinect.

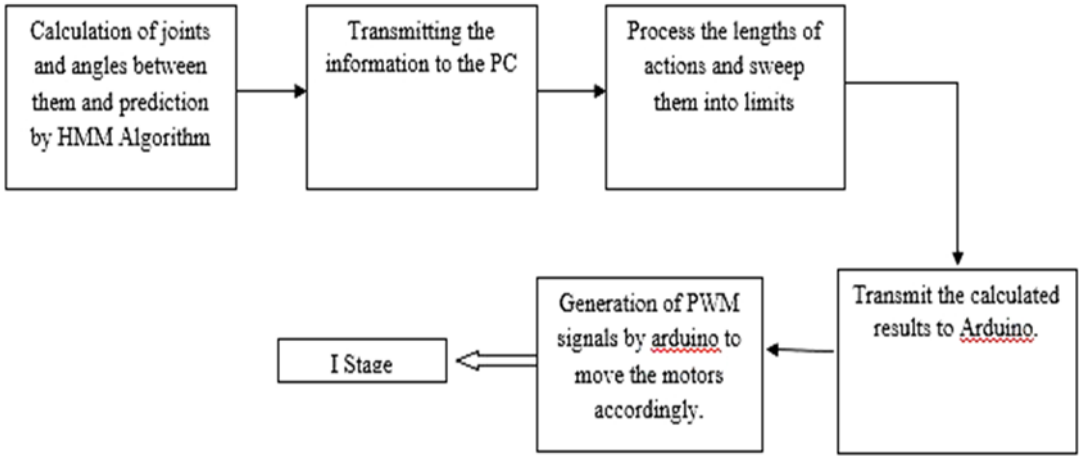

Fig. 5. Stage 2. Calculation and implementation.

8. Wheelchair Operations

The motors run continuously in forward or backward direction as per the input given by use of right hand. For the angular movements only one motor will be moving while other will be in stationary position.

For turning right the right motor will stop and left will rotate the wheelchair up to a certain extent and similarly for turning left , the left motor will stop allowing the right motor to rotate wheelchair in left direction.

9. Observations and Results

The observations are recorded considering the initial calibration position as a zero reference. The variations in the speed was recorded using tachometer and the angular movements the optical encoder in the feedback.

Table 1. Speed control using right hand gestures.

S.no	Right hand movement's variation in upward direction (cm)	Speed in forward direction (km/h)	Right hand movement's variation in downward direction (cm)	Speed in reverse direction (km/h)
1.	0	0	0	0
2	5	2.3	5	1.3
3	10	4.1	10	2.4
4	15	6.9	15	4.6
5	20	9.1	20	5.9
6	25	11.6	25	7.6
7	30	13.6	30	9.2

Table 2. Direction control using left hand gestures.

S.no	Left hand movement in upward direction (cm)	Left rotation (degrees)	Left hand movement in downward direction (cm)	Right rotation (degree)
1.	0	6.7	0	6.9
2	5	13.6	5	14.5
3	10	20.8	10	19.4
4	15	32.6	15	33.8
5	20	40.7	20	39.8
6	25	51.6	25	50.7
7	30	58.3	30	59.4

10. Conclusions

The efficiency obtained by this system is good enough for a practical implementation. The hand gesture recognition on a repeated trials yielded as much as 86 % similar outcomes as its preceding trials which makes its behaviour in real life circumstances reliable and trusted.

The cost of this design is nearly 30-40 % cheaper than the other viable options as LIDAR, which makes it industrially feasible.

References

[1]. Agarwal A., Triggs, B., 3D human pose from silhouettes by relevance vector Regression, *in Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, Vol. 2, 2004, pp. 882-888.

[2]. Dnyaneshwar R. Uttaarwar, Motion Computing using Microsoft Kinect, in *Proceedings of the National Conference on Advances on Computing*, 2013.

[3]. Rodríguez, N. D., Wikström, R., Lilius, J., CuÃl'llar, M. P., Flores, M. D. C., Understanding Movement and Interaction: An Ontology for Kinect-Based 3D Depth Sensors, in Ubiquitous Computing and Ambient Intelligence. Context-Awareness and Context-Driven Interaction, *Springer*, 2013, pp. 254-261.

[4]. J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, Real-Time Human Pose Recognition in Parts from a Single Depth Image, in CVPR, *IEEE*, 2011.

[5]. L. Xia, C.-C. Chen, and J. K. Aggarwal, Human Detection Using Depth Information by Kinect, in *Proceedings of the IEEE Computer Society Conference on Computer Vision (HAU3D) and Pattern Recognition Workshops (CVPRW),* Colorado Springs, CO, 2011, pp. 15-22.

[6]. Xia, L., Chen, C. C., Aggarwal, J. K., View invariant human action recognition using histograms of 3d joints, in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW),* 2012, pp. 20-27.

[7]. Sung, J., Ponce, C., Selman, B., Saxena, A., Human Activity Detection from RGBD Images, in *Proceedings of the AAAI Plan, Activity, and Intent Recognition Workshop*, 2011.

[8]. Maiike Johanna Theresia Veltmaat, Recognizing Acitivities with the Kinect A logic-based approach for the support room, MSc Thesis, *Radboud University Nijmegen,* The Netherlands, 2013.

[9]. J. Wang, Z. Liu, Y. Wu, J. Yuan, Mining Actionlet Ensemble for Action Recognition with Depth Cameras, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 1290 - 1297.

[10]. Ong, W. H., Palafox, L., Koseki, T., Investigation of Feature Extraction for Unsupervised Learning in Human Activity Detection, *Bulletin of Networking, Computing, Systems and Software,* 2, 1, 2013, pp. 30.

[11]. Maierdan, M., Watanabe, K., Maeyama, S., Human behavior recognition system based on 3-dimensional clustering methods, in *Proceedings of the 13th IEEE International Conference on Control, Automation and Systems (ICCAS),* 2013, pp. 1133-1137.