GESTURE RECOGNITION AS A SERVICE GRASS

[1]Prof.Anupama HS,[2]Y.Navaneeth Krishnan, [3]Avinash Shankar
[1,2]Department of Computer Science Engineering, RV College of Engineering,Bangalore ,India
[3]Collaboration Information Technology Group, CISCO Systems India Private Limited, Bangalore,India
Email: [1]anupamahs@rvce.edu.in,[2]navaneethrvce@gmail.com,[3]avshanka@cisco.com

***Abstract*—Gesture Recognition is the process of recognizing human gestures via mathematical algorithms. Effective gesture recognition without wearing any devices on the arm is achieved by depth cameras like Leap motion, Nintendo wii or Microsoft Kinect. Gesture Recognition devices be it a depth camera or a normal camera generate tracking data about the hands in the field of view. Gesture Recognition as a service is implemented in three modules. First, the gesture recognition device is connected to any embedded board such as the raspberry pi or the intel Galileo board . A gesture tracking algorithm generates hand tracking data which is uniform for any type of gesture recognition device which is connected to the board. This hand tracking data is then uploaded to a node.js server running in the cloud by means of websockets. The cloud then makes this tracking data publically available via an HTTP server or via the websockets server. Third party applications which need to incorporate gesture recognition as a service send either a GET request or establish a websocket connection to the cloud. The tracking data is received in the form of JSON which can then be parsed to identify the different type of gestures performed.**

Keywords-Gesture, Leapmotion, node.js, cloud, HTTP

I. INTRODUCTION TO GESTURE RECOGNITION

Touch screen technology is now giving way to Gesture based technology. Gesture recognition technology can be achieved accurately using cameras with depth capability such as Microsoft Kinect, Leap Motion or Wii[1]. These devices produce hand tracking data which can be leveraged by developers to create gesture based applications. However the data offered by these gesture recognition devices are not standard and they require the device to be connected to the computer on which the application is running [2].

Gesture recognition involves tracking human hand motion along the x,y and z axis in a 3D interaction box. Gestures can be static (posture or certain pose) which require less computational complexity or dynamic (sequence of postures) which are more complex but suitable for real time environments [3]. Different methods have been proposed for acquiring information necessary for recognition gestures system . Some methods used additional hardware devices such as data glove devices and color markers to easily extract comprehensive description of gesture features . Other methods based on the appearance of the hand using the skin color to segment the hand and extract necessary features , these methods considered easy, natural and less cost comparing with methods mentioned before [4][5].

The gesture recognition as a service generally involves Gesture Recognition locally, generating uniform tracking data across various gesture recognition devices and uploading them to the cloud . The cloud then publishes the tracking data making it publically accessible as a dynamically changing JSON file. The JSON file can then be requested by third party applications and parsed accordingly to make them gesture enabled.

By offering Gesture recognition as a subscription model on the cloud , users can pay and use gesture recognition without having to code individually for different gesture tracking devices , deaf mute people can use the service to convert their sign language into text[37] and also audio into sign language. GRASS was tested on various third party applications like VoIP phone , Robotic Arm, Room Ambience Control , Telepresence control among others.

II. GESTURE RECOGNITION USING LEAP MOTION

Leap Motion is an USB sensor device released in July 2013 by Leap Motion Inc., designed to provide real-time tracking of hands and fingers in three-dimensional space with 0.01 millimetre accuracy. It allows a user to get information about objects located in device's field of view (about 150 degree with distance not exceeding 1 meter). Details of how Leap Motion performs 3D scene capturing have not been revealed by Leap Motion, Inc. However, it is known that hardware consists of three infrared LEDs which are used for scene illumination, while two cameras spaced four centimetres apart capture images with 50– 200 fps framerate, dependant whether USB 2.0 or 3.0 is used.[20][21]

Unlike Microsoft Kinect, Leap Motion does not provide access to raw data in the form of a cloud of points. Captured data is processed by proprietary drivers supplied by vendor and accessible through API. Leap Motion was intended to be a human-computer interface, not general purpose 3D scanner, so it is optimised for recognizing human hands and pointy objects. The main data container, from Leap Motion API is a Frame. Average frame rate while using dual core laptop and USB 2.0 interface, is 50 frames per second[22]. One frame consists of hands. fingers, pointables (objects directly visible by controller), and additional information, like gestures recognized by simple built-in recognition mechanism, frame timestamp, rotation, translation, and scaling data. For purposes of this project, a new data format is created. It contains only information necessary to easily save captured frames to file, and read them later for processing and testing purposes.

Gestures are classified into action gestures and parameterized gestures [16]. The former group, represents gestures, which their identification is based only on detection of gesture occurrence. This means that action gestures have assigned some meaning, and the occurrence of this type of gesture implies execution of pre-defined action. It should also be noted that gestures of this type do not convey any additional parameters which define or describe the action. An example of this type can be gesture showing open hand, which could mean stop playing music. Another example might be a moving index finger in circles, which means rotating previously selected object of the specified number of degrees.

Gestures, which may be included into a parameterized gestures , provide additional parameters related to the context of the gesture. In contrast to previous group, parametrized gestures carry additional meaning or parameters needed to perform the gesture. An example of this kind of gestures might be a similar gesture to the previously mentioned instance of action gestures — making circles through moving finger — but in this case — in addition to returned information about recognized gesture — a value of angle, which should be used to rotate selected object, is also be conveyed.

Action gestures is divided into static and dynamic . The former group relates to gestures, that are not variable in time, therefore hand posture, its position and rotation are fixed during hand gesture performing. It should also be mentioned that these gestures should be independent of the orientation relative to the environment and the occurrence of this gesture is detected only on the basis of hand posture.

Dynamic action gestures refer to the group of gestures which are constant and not parametrized variable over time in terms of hand posture, position and rotation in space. This group is further divided into global, local and mixed [23]. Global were detected based on a fixed hand posture and specified movement, which may relate both to changes in position and rotation. In the case of local, hand posture is only variable in time, while the mixed include gestures, for which both the hand posture, as well as the position and rotation vary with time.

For parameterized gestures division of locally, globally and mixed parameterized gestures is distinguished. Globally parameterized gestures include gestures, whose parameters are determined by values of position and rotation or its changes of the hand. For instance, swipe gesture can be considered as globally parametrized gesture, where the parameter is the length of the swipe motion. Recognition of globally parametrized gestures is performed on the basis of unchanging hand posture or specific changes in the shape of the hand posture over time. Locally parameterized gestures include gestures whose hand posture or its changes are parameterized, for example a distance between index finger and thumb of one hand may be a parameter for scaling gesture. Gestures of this kind should be independent of the position and rotation in space. Recognition should be based on the hand posture, which can be problematic when it will be time-varying.

LeapSDK provides built-in classes representing real-world object seen by the controller[16]. The basic data unit that is got from Leap Motion is a Frame. Frame contains objects like Hands and Pointables (Fingers and Tools), described by features directly related real attributes. Hand is an object representing a regular human hand. It contains Fingers, and is described by three dimensional values, like: position of center of hand, normal vector and direction vector (pointing from the center to the end of fingers). Pointables are objects like Fingers or Tools (which are longer and thinner than Fingers). Both are described by the same set of features: position of tip, pointing direction vector, length and width. All positions are expressed in millimetres, relative to position of controller which is always located in the centre of the 3D space. Manufacturer claims, that the accuracy of device is about 0.01mm. Experiments have shown results better than 0.2mm, using an industrial robot moving an object in controllers' field of view. This is more than enough, as accuracy of positioning human hand is about 0.4mm.

Stability of images acquired and used by Leap Motion to provide the information about hands and finger can be different. Strong sunlight or fast movements may result in obtaining data that is noisy. Sometimes, the noisy information processed by Leap Motion result short-time lost tracking of fingers, or non-existing objects appear in a view. Those short-time malfunctions happen usually last less then five frames and additional pre-processing can be introduced[16].

The pre-processing principle is based on a median filter – it uses a window with defined size w to detect and remove noisy information. For a given frame and every unique finger detected in the window of frames, the algorithm checks if the processed finger is a real finger or a noise information. The neighbourhood of width w can be understood as set of w/2 frames that were recorded earlier that the currently analysed frame and w/2 frames that were recorded later. In this neighbourhood, the occurrences of a finger (fx) are counter and decision is made:

• if fx > w/2 then a finger is believed to be a real finger,

• otherwise, a finger is believed to be a data noise.

If the finger does not exist in a current frame and described check indicates that it should, then its position is calculated using linear interpolation of finger positions in two closest frames. Otherwise, a finger is simply removed. It is worth noticing, that preprocessing introduces delay in data transmission to the next processing blocks of gesture recognition. Using wider width of preprocessing causes linear grow in delay time. For example, while capturing data with framerate equal to 50 frames per second and a window size equal four, the delay would be equal to: (w/(2+1))/fps=3/50=0.06s. The delay of 0.06s is not affecting the recognition speed, but larger windows may introduce noticeable delay that might be undesirable in the application of the library.

III. Implementation of Gesture recognition As A Service

Gesture Recognition As a Service consists of three main components 1)Localized Gesture Recognition 2) Cloud Normalizer 3) JSON parsing.

A. *System Architecture*

Gesture recognition is made available as a service by provisioning a Linux VM from the cloud. The Linux VM then runs the Tracking Data collector and Tracking Data Publisher . The Tracking data collector collects the gesture recognition device's tracking data and sends it to the tracking data publisher. The tracking data publisher then makes it available in the form of a JSON file which can then be used to integrate with the various modules by parsing as shown in Figure 1.

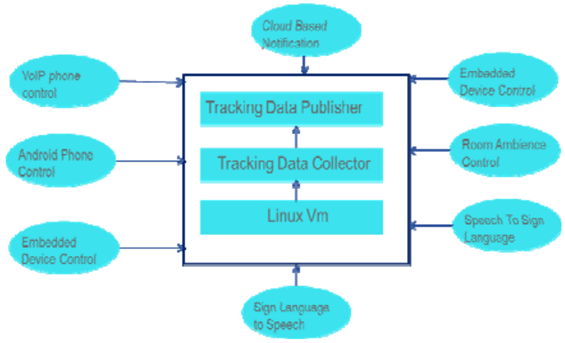

Fig 1- System Architecture of GRASS

B. *Localized Gesture Recognition*

Gesture recognition is the mathematical interpretation of a human hand motion by a computing device

●Purpose: To achieve device specific gesture recognition locally using an embedded board.

●Functionality: To recognize circle gesture, swipe gesture, screen tap gesture and key tap gesture locally on a specific device.

●Input: Hand movement over the gesture recognition device

●Output: Tracking data which is specific to that particular gesture recognition device is uploaded to a server running on the cloud.

Figure 2 shows the Low Level Design of Localized Gesture Recognition. The gesture recognition device sends raw tracking data which consist of numerous parameters (depending on the device used) to the embedded board. The embedded board analyses this data to verify if an hand has been detected by the gesture recognition device, if it has been detected, then it does an HTTP Post to the cloud. The information also contains the information of the hand , including the how many fingers are opened and how many fingers are closed.

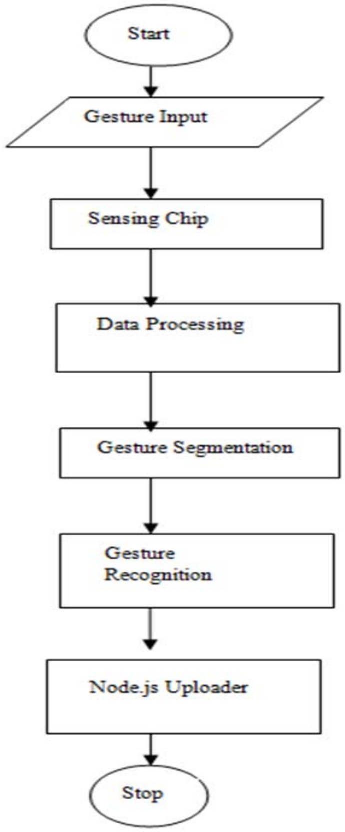

Fig 2- System Architecture of GRASS

C. *Cloud Normalizer*

The Cloud Normalization takes care of gathering the tracking data from the gesture device , normalizes it for any gesture recognition device ,that is it generates the same tracking data for different devices. It then publishes the gesture data making it publically available as a dynamically changing JSON file . Figure 3 shows the cloud normalization flowchart .

- Purpose: To standardize gesture recognition over different gesture recognition devices.

- Functionality: Take the gesture recognition data which is standard to a particular device, normalize it to any gesture recognition device and then publish the data over the cloud as a dynamically changing JSON file.

- Input: Gesture tracking data which is specific to one particular gesture device.

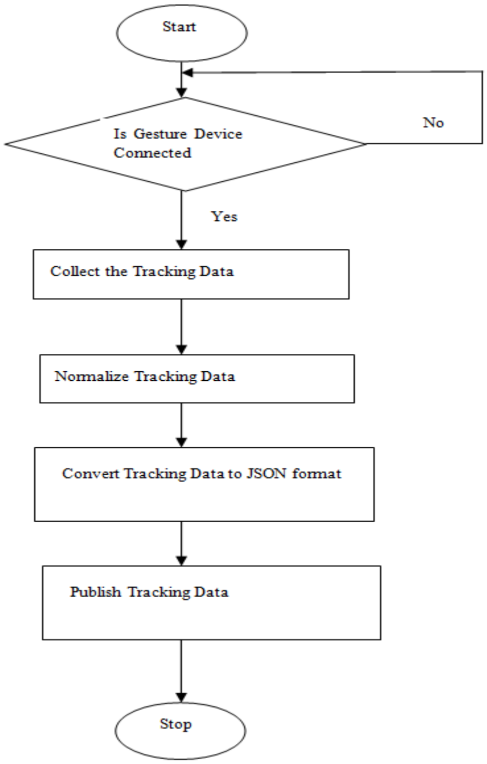

Fig 3- Low Level Design for Cloud Normalizer

D. *JSON Parsing*

JSON Parsing basically occurs at the client where the dynamically changing JSON file is received over HTTP or via a websocket connection and then parsed accordingly to find out gestures performed , number of hands in the interaction range, number of fingers in the interaction range and also the volume of the interaction box [30][31]. From this information various third party functionalities can be invoked based upon the different hand movements.

- Purpose: To incorporate gestures into third party applications by parsing the Gesture based JSON file
- Functionality: To make a connection to the cloud and parse the dynamically changing JSON file to integrate gestures into 3rd party applications.
- Input: JSON file containing gesture information.
- Output: Gesture based 3rd party applications.

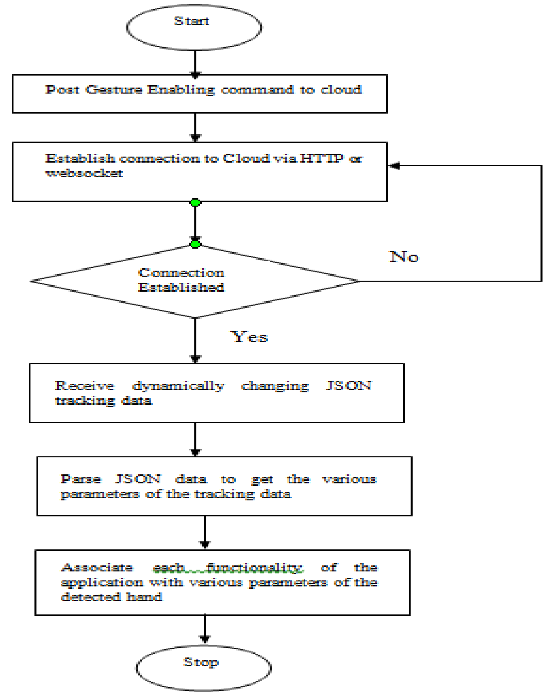

Fig 4- Low Level Design for JSON Parsing

IV. RESULTS AND SCREENSHOTS

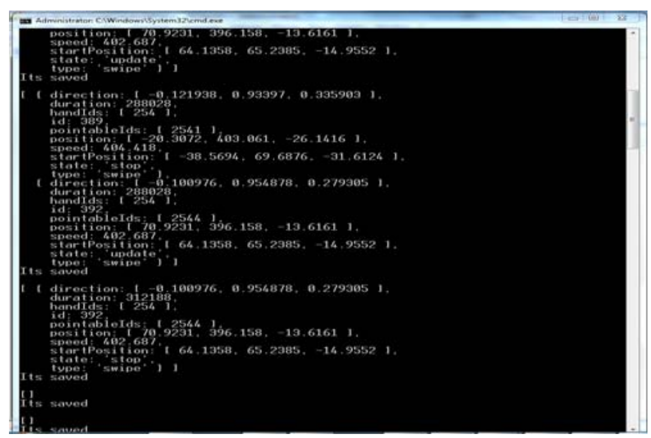

Fig 1- Output on connecting to the GRASS services

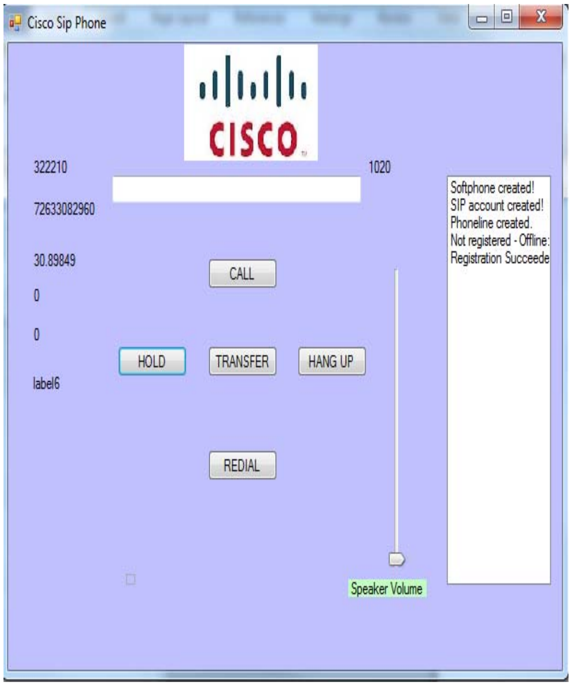

Fig 2- Gesture Based VoIP phone

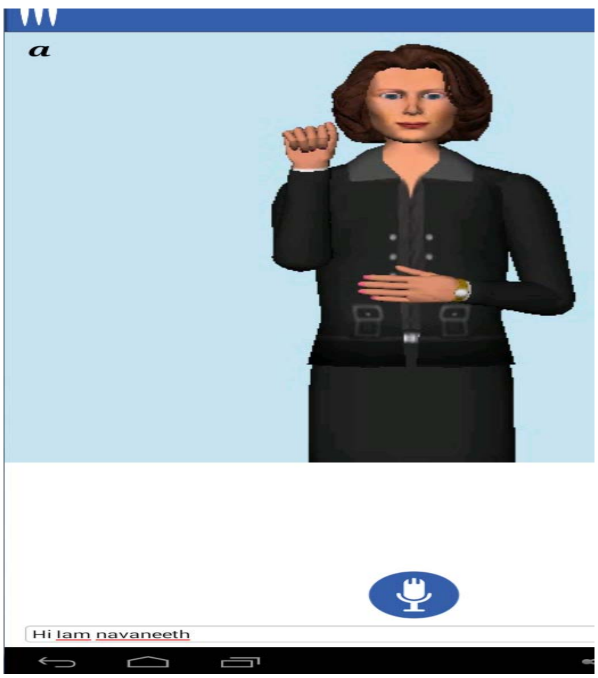

Fig 3- Speech to sign language converter using GRASS

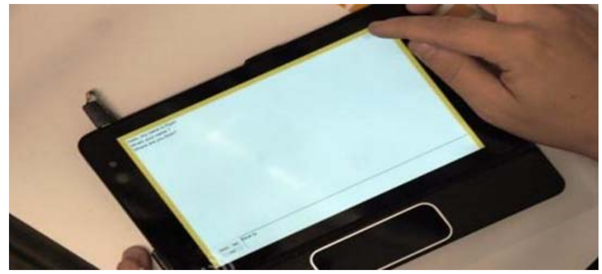

Fig 4- Sign Language to speech convertor

REFERENCES

[1] Preventis, A. ; Stravoskoufos, K. ; Sotiriadis, S. ; Petrakis, “Interact: Gesture Recognition in the Cloud” *Utility and Cloud Computing (UCC), 2014 IEEE/ACM 7th International Conference Publication*, pp: 501- 502, 2014.

[2] G. R. S. Murthy, R. S. Jadon. “A Review of Vision Based Hand Gestures Recognition,” *International Journal of Information Technology and Knowledge Management*, vol. 2(2), pp. 405- 410, 2009.

[3] S. Mitra, and T. Acharya, “Gesture Recognition: A Survey”, *IEEE Transactions on systems, Man and Cybernetics, Part C: Applications and reviews*, vol. 37 (3), pp. 311-324, 2009.

[4] Harshith. C, Karthik. R. Shastry, Manoj Ravindran, M. V. V. N. S Srikanth, Naveen Lakshmikhanth, "Survey on various gesture recognition Techniques for interfacing machines based on ambient intelligence", *International Journal of Computer Science & Engineering Survey (IJCSES)* ,Vol. 1, No. 2, pp 65-68, 2010.

[5] Piyush Kumar, Jyoti Verma and Shitala Prasad,”Hand Data Glove: A Wearable Real-Time Device for Human-Computer Interaction”, *International Journal of Advanced Science and Technology*, Vol.43, pp. 550-551, June,2012.

[6] Dipietro, L., Sabatini, A.M., Dario, P.:” A survey of glove-based systems and their applications”. *IEEE Transactions on Systems, Man, and Cybernetics,* Part C: Applications and Reviews, pp. 461–482, 2008.

[7] C. Yu, X. Wang, H. Huang, J. Shen and K. Wu, “Vision-Based Hand Gesture Recognition Using Combinational Features”, *IEEE Sixth*

International Conference on Intelligent Information Hiding and Multimedia Signal Processing, pp. 543-546, 2010.

[8] Maria Karam and M. C. Schraefel. A taxonomy of gestures in human computer interactions, Southhampton University,Technical report, 2005.

[9] Rama B. Dan ,P. S. Mohod,"Survey on Hand Gesture Recognition Approaches", *International Journal of Computer Science and Information Technologies*, Vol. 5 (2) , , pp. 2050-2052, 2014.

[10] Leder, R.S. ; Azcarate, G. ; Savage, R. ; Savage, S. ; Sucar, L.E. ; Reinkensmeyer, D. ; Toxtli, C. ; Roth, E. ; Molina, "Nintendo Wii remote for computer simulated arm and wrist therapy in stroke survivors with upper extremity hemipariesis" *Virtual Rehabilitation,* pp 74-78, 2008.

[11] Aysun Bascetincelik, " WiiRobot: Controlling Robots with Wii Gestures", Master's Thesis, Department of Computer Science, Brown University, 2009

[12] K. Abe, H. Saito, S. Ozawa: Virtual 3D Interface System via Hand Motion Recognition From Two Cameras. *IEEE Trans. Systems, Man, and Cybernetics*, Vol. 32, No. 4, pp. 536-540, July 2002.

[13] A. Ali and J. K. Aggarwal, "Segmentation and recognition of continuous human activity", *Proceedings of IEEE Workshop on Detection and Recognition of Events in Video*, vol 1(5), pp. 28 – 35, 2001.

[14] Haitham Hasan and S. Abdul-Kareem, Static hand gesture recognition using neural networks. *Artificial Intelligence Review*, pp 1–35, 2012.

[15] Biswas, K.K., Basu, S., "Gesture Recognition using Microsoft Kinect", Proceedings of the *IEEE International Conference on Automation, Robotics and Applications* (ICARA), vol2(5), pp.40-45, 2011.

[16] Leap Motion | Mac & PC Motion Controller for Games, Design, & More. 2014. Available at: http://www.leapmotion.com. [Accessed January 2015].

[17] Mladen A.Vouk,"Cloud Computing-Issues, Research and Implementations", *Journal of Computing and Information Technology*, vol 4, pp 235-246, 2012.

[18] Dimpi Rani, Rajiv Kumar,"A Comparative Study of SaaS,PaaS and IaaS in Cloud Computing",*International Journal of Computer Science, Engineering and Information Technology*,Volume 4 Issue 6 , pp 250-253,2014.

[19] Mohamed Magdy Mosbah, "Current Services in Cloud Computing: A Survey ," *International Journal of Computer Science, Engineering and Information Technology* (IJCSEIT), Vol.3, No.5, October 2013.

[20] Junchen Shen ; Yanlin Luo ; Xingce Wang ; Zhongke Wu ; Mingquan Zhou," GPU-Based Realtime Hand Gesture Interaction and Rendering for Volume Datasets Using Leap Motion" , *Cyberworlds (CW), 2014 International Conference on* pp: 85 - 92, 2014.

[21] Metz, R. "Leap Motion's Struggles Reveal Problems with 3-D Interfaces," in: *MIT Technology Review*. Cambridge, MA: MIT,2013.

[22] Han,Nicholas gold, "Lessons Learned In Exploring Leap Motion Sensor For Gesture Based Instrument design", *Proceedings of the International Conference on New Interfaces for Musical Expression*, pp. 485-489, 2014.

[23] Nicholas Gillian and Joseph Paradiso. The gesture recognition toolkit. In New England Machine Learning Day, London, 2012.

[24] Chih-Chung Chang and Chih-Jen Lin. "LIBSVM: A library for support vector machines". *ACM Transactions on Intelligent Systems and Technology*, pp. 2:27:1–27:27, 2011.

[25] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: "A library for large linear classification ". *J. Mach. Learn.* Res., pp.1871–1874, June 2008.

[26] Rafiqul Zaman Khan, Noor Adnan Ibraheem," Hand Gesture Recognition a Literature Survey", *International Journal of Artificial Intelligence*

& Applications (IJAIA), Vol.3, No.4, pp. 23-25, July 2012.

[27] Hamido Fujita, Roberto Revetria, "New Trends in Software Methodologies, Tools and Techniques*", International Conference on New Trends in software Methodology, Tools and Techniques* (SoMeT_12), Genoa, Italy , pp. 3-20, Sept. 2012.

[28] Ching-Hua Chuan ; Regina, E. ; Guardino, C."American Sign Language Recognition using Leap Motion", *International Council for Machine Learning Applications* pp: 541- 544, 2014.

[29] W. Stevens, G. Myers, L. Constantine, "Structured Design", *IBM Systems Journal*, vol. 13 no. 2, 1974, pp. 115-139.

[30] C.Cui and H.Ni., "Optimized simulation on XML with JSON". *Communication Technology* (Chinese) 42(8), pp. 108-111. 2009.

[31] T. Zhang, Q. Huang, Y. Mao and X. Gao. Algorithm of object serialization based on JSON. *Computer Engineering and Applications*(Chinese) 43(15), pp 98-100, 2007.

[32] Shivkumar Hasmukhrai Trivedi," Software Testing Techniques", *International Journal of Advanced Research in Computer Science and Software Engineering*, Volume 2, Issue 10, pp 100-108October 2012.

[33] Haniyeh Kazemitabar, Sameha Ahmed, Kashif Nisar, Abas B Said, Halabi B Hasbullah, "A Survey on Voice over IP over Wireless LANs", *World Academy of Science, Engineering and Technology*, pp 71-75, 2010.

[34] Pisharady, P.K., Vadakkepat, P., Loh, A.P.: "Attention based detection and recognition of hand postures against complex backgrounds". *International Journal of Computer Vision* 101, pp. 403–419,2013.

[35] Md. Anisur Rahman , Alimul Haque Khan , Dr. Tofayel Ahmed , Md. Mohsin Sajjad," Design, Analysis and Implementation of a Robotic Arm-The Animator", *American Journal of Engineering Research (AJER),* vol 2, Issue-10, pp-298-307, 2013.

[36] Stowers, M. Hayes, and A. Bainbridge-Smith, "Quadrotor Helicopters for Visual Flight Control," in *Proceedings of Electronics, New Zealand Conference 2010*, Hamilton, New Zealand, pp. 21-26, 2010.

[37] Bassily, D. ; Georgoulas, C. ; Guettler, J. ; Linner, T. ; Bock, T." Intuitive and Adaptive Robotic Arm Manipulation using the Leap Motion Controller" *ISR/Robotik 2014; 41st International Symposium on Robotics; Proceed Publication,* pp: 1- 7,2014.

[38] Ashbrook, D. and Starner, T. MAGIC: A Motion Gesture Design Tool. *Proceedings* of *CHI '10, ACM (2010)New York*, pp 2159-2168, 2010.

[39] Erik Eloff & Daniel Torstensson," An Investigation into the Applicability of Node.js as a Platform for Web Services", LinKopings University, Department of Computer Science and Information Science,Master's Thesis,2008.

[40] Ward, J.A., Lukowicz, P., Gellersen, H.W.: "Performance metrics for activity recognition". *ACM Transactions on Intelligent Systems and Technology* , pp .6:01– 6:23, 2011