



Eliminating drift of the head gesture reference to enhance Google Glass-based control of an NAO humanoid robot

Xiaoqian Mao¹, Xi Wen¹, Yu Song¹, Wei Li^{1,2,3}
and Genshe Chen⁴

Abstract

This article presents a strategy for hand-free control of an NAO humanoid robot via head gesture detected by Google Glass-based multi-sensor fusion. First, we introduce a Google Glass-based robot system by integrating the Google Glass and the NAO humanoid robot, which is able to send robot commands through Wi-Fi communications between the Google Glass and the robot. Second, we detect the operator's head gestures by processing data from multiple sensors including accelerometers, geomagnetic sensors and gyroscopes. Next, we use a complementary filter to eliminate drift of the head gesture reference, which greatly improves the control performance. This is accomplished by the high-pass filter component on the control signal. Finally, we conduct obstacle avoidance experiments while navigating the robot to validate the effectiveness and reliability of this system. The experimental results show that the robot is smoothly navigated from its initial position to its destination with obstacle avoidance via the Google Glass. This hands-free control system can benefit those with paralysed limbs.

Keywords

Hand-free control, Google Glass, head gesture, NAO humanoid robot, reference drift, sensor fusion, complementary filter

Date received: 28 October 2016; accepted: 30 December 2016

Topic: Special Issue - Multi-Modal Fusion for Robotics

Guest Editor: Huaping Liu

Associate Editor: Huaping Liu

Introduction

Since the 20th century, intelligent robots have been applied into various fields due to the development of human machine interaction (HMI) technology to effectively apperceive the environment for dynamic decision-making and planning, behavior control and execution. More and more scientists are researching on it. For example, Liu et al. proposed a novel object recognition method using visual-tactile fusion information based on a sparse coding method¹⁻³, which greatly improves the object recognition accuracy; Li et al. designed a grasping posture control for a robotic arm based on novel adaptive particle swarm optimization, which made the object grasping of home service robot quicker and more accurate⁴. Nowadays, most of the existing HMI methods rely on the operator's hands. Zhang

et al. adopted a commercially available, standard joystick as the candidate HMI device of the robot⁵; Speers et al. achieved the goal of controlling Robot Operating

¹School of Electrical Engineering and Automation, Tianjin University, Tianjin, China

²State Key Laboratory of Robotics, Shenyang Institute of Automation, Shenyang, Liaoning, China

³Department of Computer and Electrical Engineering and Science, California State University, Bakersfield, CA, USA

⁴Intelligent Fusion Technology, Inc., Germantown, MD, USA

Corresponding author:

Wei Li, Department of Computer and Electrical Engineering and Science, California State University, Bakersfield CA 93311, USA.

Email: wli@csu.edu



Creative Commons CC BY: This article is distributed under the terms of the Creative Commons Attribution 3.0 License

(<http://www.creativecommons.org/licenses/by/3.0/>) which permits any use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (<https://us.sagepub.com/en-us/nam/open-access-at-sage>).

System-enabled robots by sending commands from the lightweight tablet⁶; Ochiai et al. proposed a remote control system to achieve the autonomy of multiple mobile robots using touch panel interface.⁷ However, with the technology development demands, the operation tasks have become more and more complicated. For instance, using a multi-robot system to perform, a complex process demands an operator's attention constantly shift among various tasks to maintain the necessary awareness of a given situation, such as exploring environments and locating targets simultaneously,⁸⁻¹⁰ and manipulators fitted into the 'Jiaolong' deep manned submersible vehicle have to collaborate with each other to complete a complex task.^{11,12} In addition, target capture is an essential and key mission for the tethered space robot, which requires controlling a gripper, a space tether and a space platform at the same time.¹³⁻¹⁵ Using those techniques, a single operator has to adopt multiple cooperated patterns to complete the complex operational tasks. In order to avoid investing extra manpower and material resources, it has become necessary to find a new way to increase sources for generating control signals and efficiently representing their control intentions.

The wearable technology that appeared in 2012 has drawn the attention of researchers and engineers to exploring its application (APP) into the field of HMI, for example, a useful head mounted mouse for people with paralysed limbs,¹⁶ a wearable technology-based electric-powered wheelchairs for disabled people¹⁷ and a gel spat as a device walking assistance for old folks.¹⁸ In the same year, Google Glass, called 'expanding reality' glass, was released by Google Inc. This device with powerful data processing ability and a wearable advantage has been applied to achieve great results in some areas. Jeroudi et al. compared the accuracy among four types of electrocardiogram (ECG): ECG interpretation on a Google Glass screen, a photograph of the ECG taken by Google Glass and displayed on a mobile device, the original paper ECG and a photograph of the ECG taken with a high-resolution camera and displayed on a mobile device¹⁹; Rahimy and Garg recorded a scleral buckling surgery process using the video recording function of Google Glass²⁰; Wille et al. validated that there was no significant difference in subjective strain between wearing Google Glass or using a Tablet-PC when people were asked to follow a toy car assembly instructions while monitoring a virtual gauge²¹; in the field of acoustics, Weppner et al. developed a fully functional APP prototype gPhysics App to perform an educational physical experiment based on the Google Glass platform²²; Kapellmann-Zafra et al. used Google Glass for a guided swarm of robots to transport a large object in an environment with obstacles.²³ Our team, which is one of the pioneers, has reported preliminary research outcomes on the robot control via Google Glass.²⁴⁻²⁶

In the previous research,^{24,25} we adopted the rotation vector sensor to calculate the angle change of the rotation matrix and transfer these angle changes to the control

intentions. A critical problem that gave users a poor experience during our study was that the head reference position had significantly drifted during operations, which probably caused the failure of robot control. In this article, we use a complementary filter to fuse data measured by an accelerometer, a geomagnetic sensor and a gyroscope to update the operator's head reference gestures. Consequently, the reference drift is eliminated and the robot control performance is improved.

'Google Glass-based robot control system' section introduces the Google Glass-based hand-free control system we developed, consisting of a Google Glass and an NAO humanoid robot. 'Head gesture detection' section describes how to detect a head gesture using the complementary filter. 'Controller design' section presents the design procedure of controlling the humanoid robot via the Google Glass. 'Threshold selection and experiment validation' section discusses the threshold selection through an offline experiment and arranges an obstacle avoidance experiment to demonstrate the performance of this system. Finally, 'Discussion' section compares the proposed method with the others based on the rotation vector sensor and discusses further research activities.

Google Glass-based robot control system

Here, we briefly introduce the hand-free control system that consists of a Google Glass and an NAO humanoid robot, as shown in Figure 1.²⁵ On the Google Glass, we developed a detection section using the Java language. This section detects the operator's head movement, that is, head gesture, which represents an operator's intent, converts the intent into the corresponding control command and transmits the command to the robot. This section also has a user interface (UI) to display the command and the head gesture of the operator. On the robot, we designed a robot controller to receive instructions from the Google Glass, make the robot execute the corresponding behavior and feed the live video to the UI on Google Glass which allows the operator to make proper decisions.

Google Glass

Google Glass supports Wi-Fi and Bluetooth and is mainly composed of six sections: a GPS, a camera, a prism, a speaker, a microphone and a battery,²⁷ as shown in Figure 2(a). It runs on the operating system Android 4.4.2 (KitKat) with an OMAP 4430 SoC1.2 GHz Dual Core (ARMv7) processor produced by Texas Instruments and uses the optical head mounted display projection principle to realize 'primary amplification' for displaying information. The micro projector casts light on the reflective screen. The light reflected through the convex lens creates a large virtual screen and displays text and various data in front of the eyes. The projection of the layer is equivalent to a 25-in high-definition screen from 8-feet away, as shown in

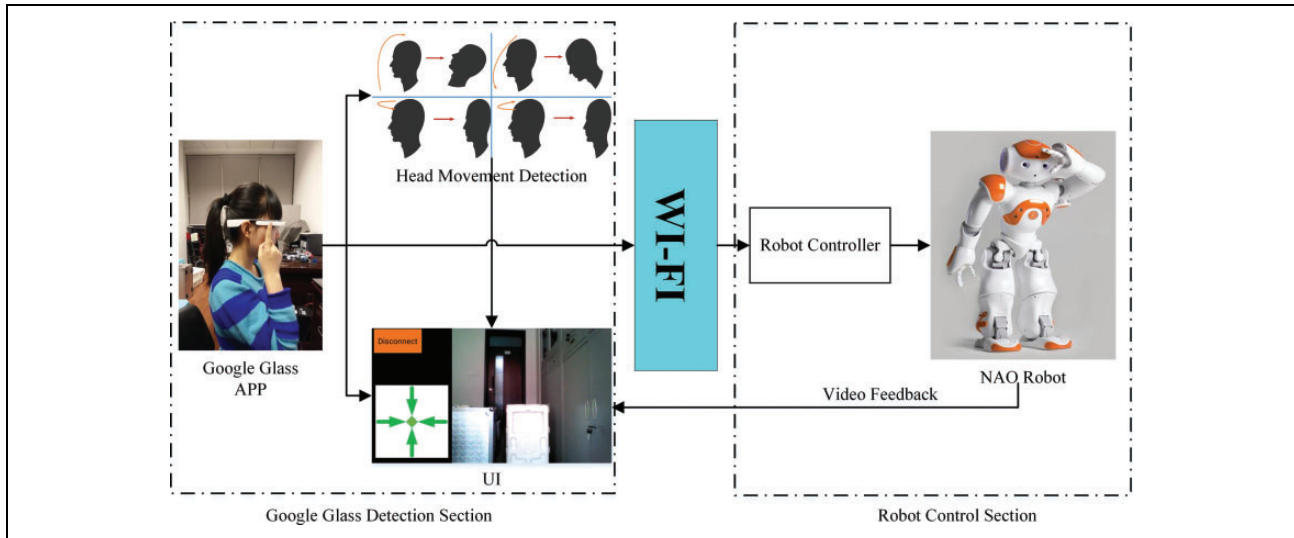


Figure 1. Google Glass-based robot control system.

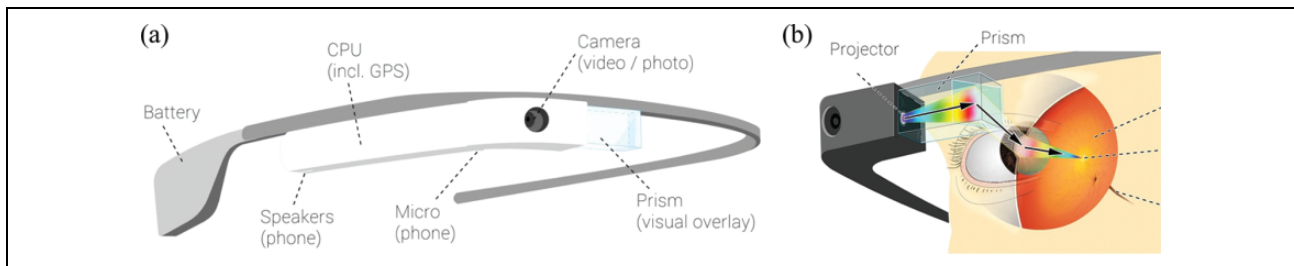


Figure 2. (a) Structure of Google Glass. (b) Superimposed layer diagram.

Figure 2(b). Moreover, a touchpad, located on the external part of the CPU case and on the right side of the device, supports multi-touch interaction. It is possible to identify the gestures of single or double tap and swiping a user's finger in a specific direction.^{27–29}

The main UI on Google Glass is called timeline which is a set of 640×360 pixel cards organized into sections, including 'home', 'past' and 'present and future'. Compared with other wearable devices, Google Glass offers various control options for the user. Moreover, Google Glass supports three types of Glassware, such as Mirror Application Program Interface (API) Glassware, GDK Glassware and Hybrid Glassware.^{27–29} In this study, the GDK Glassware is used to explore an APP to control the NAO humanoid robot.

Google Glass is also equipped with a variety of sensors, including accelerometer, gyroscope, geomagnetic sensor, direction sensor and rotation vector sensor, which are divided into software-based sensors and hardware-based sensors.³⁰

NAO robot

The NAO humanoid robot, as shown in Figure 1, is 57 cm in height and 4.5 kg in weight and was developed by the

Aldebaran Robotics company in 2006. It is designed with 25 degrees of freedom (DOFs), namely, 11 DOFs for the lower part that includes the legs and pelvis and 14 DOFs for the upper part that includes the trunk, arms and head. Each leg has two DOFs at the ankle, one DOF at the knee and two DOFs at the hip, and each arm features two DOFs at the shoulder and two DOFs at the elbow. Besides, it has one DOF at the wrist and one additional DOF for the grasping ability of the hand. The head can rotate about the yaw and pitch axes. The kinematic details are shown in Figure 3. The NAO uses brushed DC motors with magnetic rotary encoders for position feedback.³¹

The NAO is provided with an open-loop walk engine to generate and stabilize walking and uses the industry-standard zero moment point (ZMP). ZMP is defined as the point on the ground at which the net moment of the inertial and gravity forces have no component along the horizontal axes. The ZMP trajectory is calculated from user specified step parameters and transformed into a center of gravity (CoG) trajectory using an inverted pendulum model. This CoG trajectory is then tracked and maintained throughout single and double support phases using inverse kinematics. The swing leg trajectory is a cycloid.³²

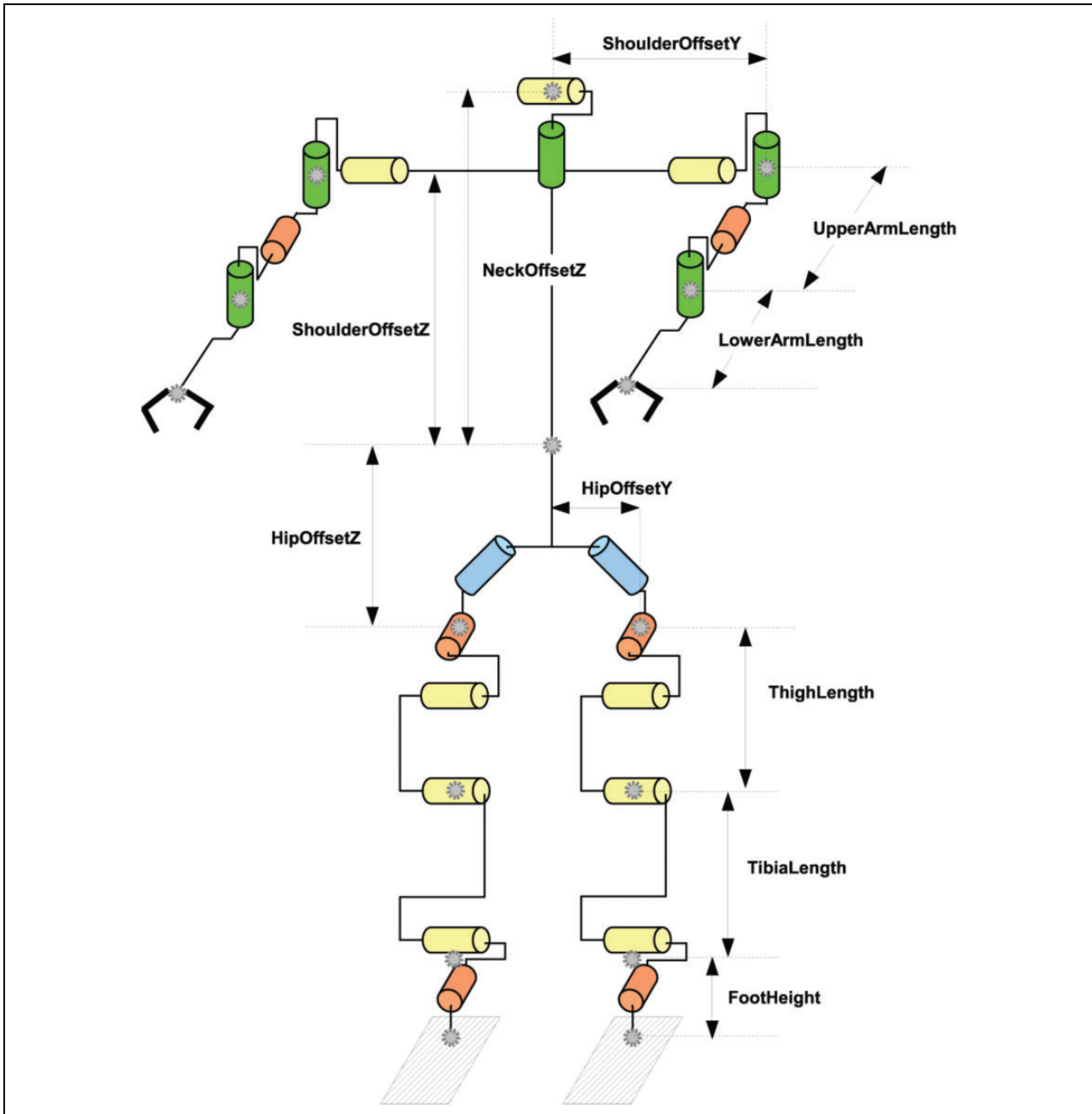


Figure 3. Detailed kinematics of NAO and wrist joint not represented.

Additionally, the NAO robot supports Ethernet or Wi-Fi, microphones, cameras and various sensors, such as ultrasonic, tactile/touch, force sensing resistors, gyroscope and accelerometer.³³ These features provide convenience for users to develop a variety of APPs. Based on the Linux operating system, the NAO humanoid robot makes users easily design different types of behaviors with the NAOqi which plays as the API and the framework. Furthermore, a lot of third-party software packages support the programming of the NAO humanoid robot. For instance, Choregraphe and WEBOTS separately offer graphical programming tools and powerful simulation models.³⁴

Head gesture detection

Raw sensor data

The head movement detection is the key to our robot control strategy, which is achieved by calculating the head gesture via a complementary filter, consisting of a low-frequency filter and a high-frequency filter, to fuse the data measured by an accelerometer, a geomagnetic sensor and a gyroscope, as shown in Figure 4. In order to describe a head gesture, a three-dimensional space coordinate system needs to be defined, and the method of space vector transformation is adopted. Then, on the Google Glass, a sensor device

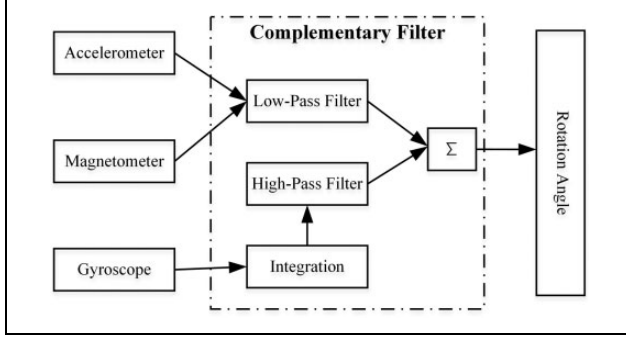


Figure 4. Multi-sensor schematic diagram.

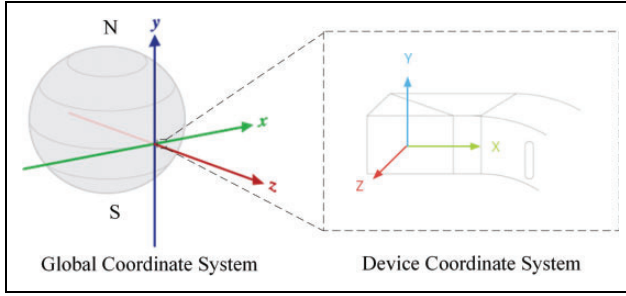


Figure 5. Sensor coordinate system of Google Glass.

coordinate system, called X - Y - Z , and a global coordinate system, called x - y - z , are defined using the Android operating system, whose relationship is shown in Figure 5. We use Euler angles (pitch, yaw and roll) to describe a head gesture which represents the transform relationship between the device coordinate system and the global coordinate system.³⁵

There are two methods for calculating a head gesture, respectively: the first one is based on the accelerometer and geomagnetic sensor, and the second one is based on the gyroscope. The API of the Android operating system provides a `SensorManager` class and a `SensorEventListener` interface which help access these sensors, listen to the sensor data changes and process the measured data. Thus, we are able to obtain the data measured by the accelerometer, geomagnetic sensor and gyroscope to build a rotation matrix R through the sensor data as follows

$$R = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \quad (1)$$

In this research, we use the function `SensorManager.getRotationMatrix(accmagMatrix, null, accValues, magValues)` to read the data from the accelerometer and geomagnetic sensor to calculate the rotation matrix and `SensorManager.getOrientation(accmagMatrix, accmagOrientation)` to derive the Euler angles

$$\begin{cases} \text{Yaw} & = & \arctan\left(\frac{r_{12}}{r_{22}}\right) \\ \text{Pitch} & = & \arcsin(-r_{32}) \\ \text{Roll} & = & \arctan\left(\frac{-r_{31}}{r_{33}}\right) \end{cases} \quad (2)$$

The second way to obtain the rotation matrix is to integrate the data from the gyroscope to calculate the Euler angles. Then, we fuse two groups of raw sensor data to calculate head gestures below.

Fusion of sensor data

As discussed above, the raw data is acquired by the multiple sensors. However, like any normal Android device, the accelerometer and geomagnetic sensor on Google Glass deliver significant noise, while the gyroscope provides a low-noise angle signal due to the integral over time but the integrated gyroscope data may quickly become non-physical, which means the integrated data does not correctly represent the actual orientation of the device. In the frequency domain, the former sensor signals are significantly affected by noise with high frequency, while the latter sensor signal includes a drift in low frequency,^{35,36} as shown in Figure 6. As a result, these raw sensors' data cannot be precisely used to calculate the operator's head gestures.

In order to improve the accuracy of the head gestures, the following sensor fusion algorithm based on a complementary filter is proposed to remove the high-frequency noise and to eliminate the drift in low frequency, separately, and then to combine the processed signals. The complementary filter is a popular sensor fusion method, as shown in Figure 7. This method can remove high-frequency noise and eliminate the drift with low frequency in parallel. Let's suppose the true value of signal $C(s)$ is measured in two different ways, so two measurements of data $C_1(s) = C(s) + \mu_H(s)$ and $C_2(s) = C(s) + \mu_L(s)$ are obtained, respectively, where $\mu_H(s)$ is a high-frequency noise and $\mu_L(s)$ is a low-frequency noise. Assuming that the transfer functions of a high-pass filter and low-pass filter are defined as $G_H(s)$ and $G_L(s)$, respectively, which construct the complementary filter system under the following condition

$$G_H(s) + G_L(s) = 1 \quad (3)$$

The output from the complementary filter shown in Figure 7 is

$$\begin{aligned} \hat{C}(s) &= C_1(s)G_L(s) + C_2(s)G_H(s) \\ &= [C(s) + \mu_H(s)]G_L(s) + [C(s) + \mu_L(s)]G_H(s) \\ &= C(s) + \mu_H(s)G_L(s) + \mu_L(s)G_H(s) \\ &\approx C(s) \end{aligned} \quad (4)$$

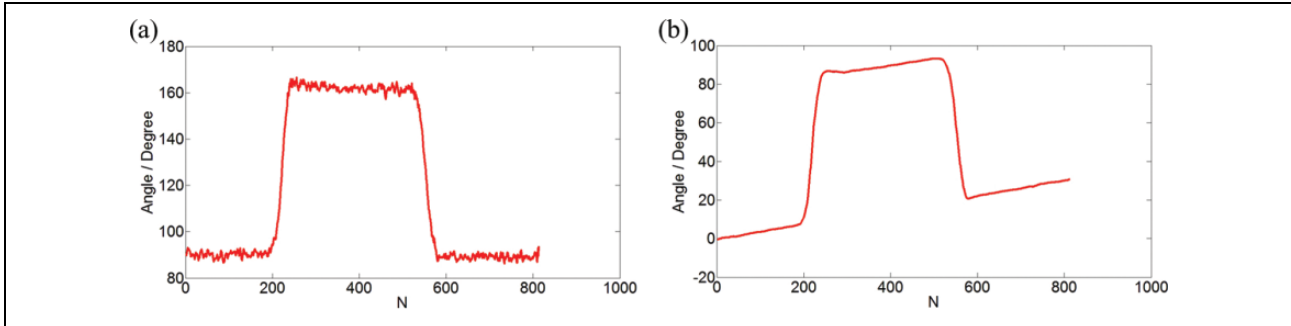


Figure 6. (a) Noise from accelerometer and magnetometer signals. (b) Drift of gyroscope.

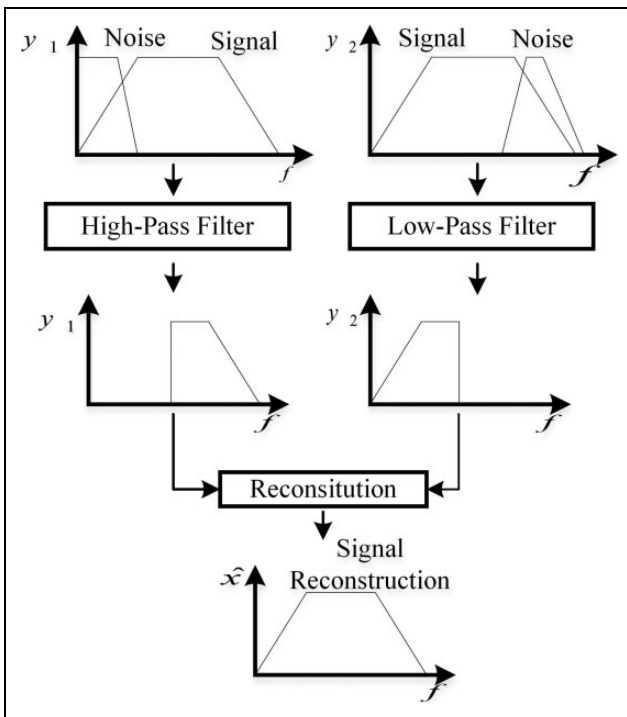


Figure 7. Principle of complementary filter.

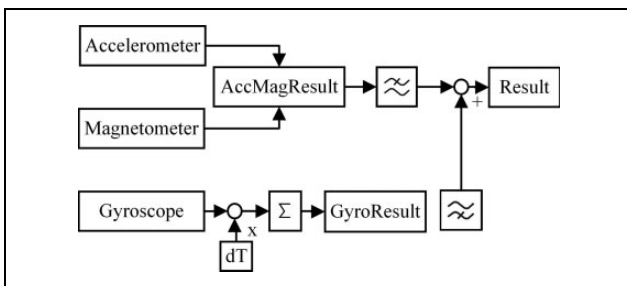


Figure 8. Process of sensor fusion.

In this study, we will develop the sensor fusion system based on the complementary filter to eliminate the angle drift integrated by the gyroscope and the data noise delivered by the accelerometer and geomagnetic sensors, as shown in Figure 8. The latest signal data from the accelerometer and magnetometer are stored in $AM(n)$, while the

latest gyroscope data are stored in $G(n)$. Here, dT is the integral interval for processing the gyroscope data. Then, the output $Y(n)$ from the sensor fusion system is calculated by

$$Y(n) = \text{FusionFactor} \times [Y(n-1) + G(n) dT] + (1 - \text{FusionFactor}) \times AM(n) \quad (5)$$

where the parameter FusionFactor is the weight that affects the response rate and proportion of the sensor data processed by both the high- and low-frequency filters. It can be calculated by

$$\text{FusionFactor} = \frac{\tau}{\tau + dT} \quad (6)$$

where the parameter τ is the time constant for both the low-pass and the high-pass filters. Here, we select the parameter FusionFactor as 0.9. This is because we want to primarily use the gyroscope data and to use a fast gyroscope measurement time dT , so the gyroscope doesn't drift more than a couple of degrees before the next measurement is taken.^{35,36} FusionFactor also plays an important role in achieving a good control quality of this system.

Head gesture detection based on sensor fusion

In this remote system, four control instructions correspond to the four types of head gestures, respectively: head up, head-turning left, head-turning right and head down. Because these head movements mainly change the value of the pitch and yaw, only these two values are investigated. When an operator makes the four kinds of head movements mentioned above and returns to the original position, then all sensor data is recorded during the movements, as shown in Figure 9. The blue and red lines, respectively, represent the pitch and yaw. Through selecting an appropriate threshold, all the head movements can be easily distinguished after the interferences from drift and noise are eliminated. In order to accurately detect a head gesture, we select a reference position which is the head position when the operator looks straight ahead and records the head gesture at this position, denoted as *referPitch* and *referYaw*. Then, any head gesture is configured by new pitch and yaw values detected by the sensors, denoted as *newPitch* and

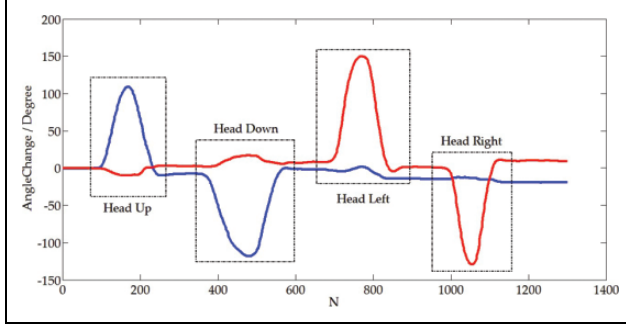


Figure 9. Measurement data of four kinds of head movements.

newYaw. In doing this, a threshold T is selected to identify the head gesture according to the ideal discrimination rule below

$$\begin{cases} \text{newPitch} - \text{referPitch} > T \Rightarrow \text{Head_Up} \\ \text{newPitch} - \text{referPitch} < -T \Rightarrow \text{Head_Down} \\ \text{newYaw} - \text{referYaw} > T \Rightarrow \text{Head_Left} \\ \text{newYaw} - \text{referYaw} < -T \Rightarrow \text{Head_Right} \end{cases} \quad (7)$$

Controller design

Command generation APP

On the Google Glass, we program a robot tele-control APP using GDK Glassware development style. This APP mainly achieves three types of functions as shown in Figure 1, respectively: (1) detecting head gesture and converting and sending instruction; (2) displaying head gesture and (3) live video feedback. They are running on three threads, namely, *Thread_1*, *Thread_2* and *Thread_3*, to reduce interference and ensure the program works smoothly. During the control process, *Thread_1*, named *Head Gesture Detection Thread*, detects the operator's head gesture, converts it to its corresponding control instruction and sends the instruction to the robot. *Thread_2*, named *UI Thread*, displays the operator's head gesture on the UI. *Thread_3*, called *Video Thread*, constantly sends the character 'I' to get live video about the environment and displays it on the UI. The whole process is shown in Figure 10(a).

Controller for NAO robot

A controller, which runs on the NAO humanoid robot, is developed using the C++ language under the Linux environment and includes two functions running on different threads: instruction receiving and behaviour performing. In order to achieve instruction receiving, the socket TCP protocol is applied to build communications between the Google Glass and the NAO Robot. Here, the robot acts as the server port, while the Google Glass is the client port. The server receives the instruction from the client in the form of input stream. At the same time, the controller parses six types of character instructions: 'F', 'S', 'L',

'R', 'I' and 'D', and their definitions are listed in Table 1. During the entire control process, the robot needs to continuously listen on the bound port in order, so that it is able to obtain the instruction from the client on time, as shown in Figure 10(b).

Threshold selection and experiment validation

It is critical to select a threshold to cut off invertible activities of the operator head, hence we generate head gesture data without control of the real robot. Then, we arrange an online experiment to validate the performance, in which an operator needs to navigate the NAO robot from its initial to its destination with obstacle avoidance.

Threshold selection

In this part, in order to find out a reliable threshold, we use the offline experiment to analyze all the data while the operator is moving his/her head. During the experiment, the operator wearing the Google Glass follows instructions and completes corresponding head movements. These instructions' including *Head_Up*, *Head_Down*, *Head_Left* and *Head_Right* are randomly generated and displayed on the glass screen. A total of 50 instructions are executed and the interval between instructions is 5 s. Figure 11 shows the recorded data of angle changes plotted in a chart.

Figure 11 shows that reset errors are usually between $\pm 10^\circ$ because of an invertible activity of the operator's head and hardly accurately returning the reference position as indicated by the yellow lines. The minimum angle change yielded by the operator head movement is about $\pm 40^\circ$ as indicated by the two black lines. So, a threshold of 10° – 40° can be selected which can identify all head movements with the accuracy of 100%. Moreover, we find that the *Head_Up* and *Head_Down* movements' change both of the pitch and yaw values as enclosed in the black rectangle in Figure 11. So, we modify the discrimination rules by considering the condition: When the yaw value triggers a rule, it is essential to check if the pitch value satisfies the rule. In this study, we use the rules in equation (8) to distinguish operator head movements. In order to obtain more measurement margins, we set threshold T as $\pm 25^\circ$, as indicated by the green lines

$$\begin{cases} \text{newPitch} - \text{referPitch} > T & \Rightarrow \text{Head_Up} \\ \text{newPitch} - \text{referPitch} < -T & \Rightarrow \text{Head_Down} \\ \begin{cases} \text{newYaw} - \text{referYaw} > T \\ |\text{newPitch} - \text{referPitch}| < T \end{cases} & \Rightarrow \text{Head_Left} \\ \begin{cases} \text{newYaw} - \text{referYaw} < -T \\ |\text{newPitch} - \text{referPitch}| < T \end{cases} & \Rightarrow \text{Head_Right} \end{cases} \quad (8)$$

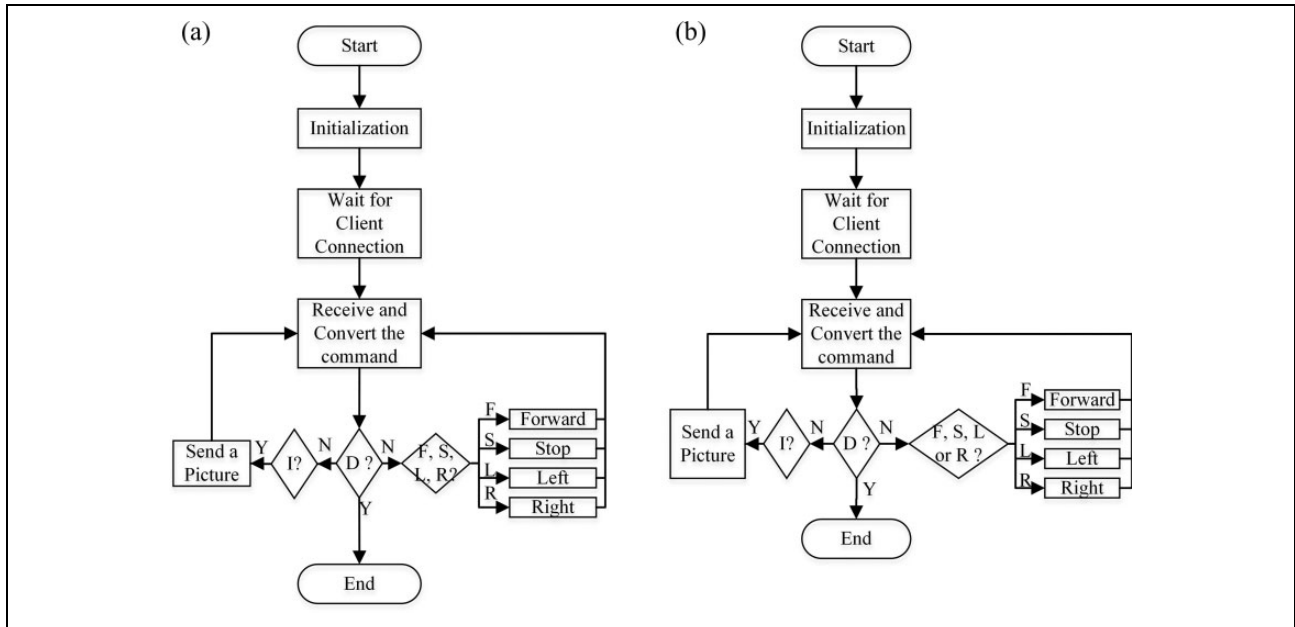


Figure 10. (a) Controller of Google Glass. (b) Controller of NAO robot.

Table 1. Definition of commands.

Command	Definition
F	Controlling the robot to go forward
S	Controlling the robot to stop
L	Controlling the robot to turn left
R	Controlling the robot to turn left
I	Sending a locale image taken by the NAO robot's camera to the Google Glass
D	Disconnecting communication between Google Glass and NAO robot

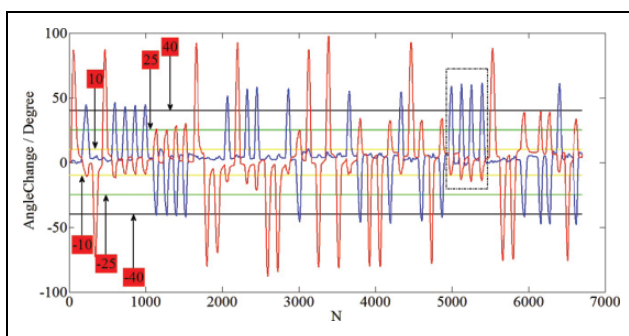


Figure 11. Chart of experimental data.

Robot navigation with obstacle avoidance

In order to validate the effectiveness of the designed system, we navigate the robot from its initial position to its destination with an obstacle avoidance experiment. First, we set a connection between Google Glass and the NAO humanoid robot. Second, we configured some initial parameters (generally, the natural position of the robot head).

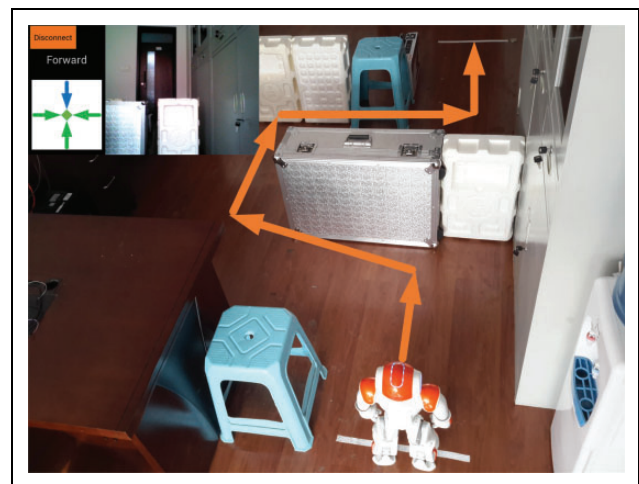


Figure 12. Experimental scene.

Third, wearing the Google Glass, the operator was able to navigate the NAO humanoid robot to pass through several narrow paths and finish obstacle avoidance with the assistance of the feedback of live videos.

The experimental scene and navigation mission are shown in Figure 12. For an ideal case, the controller just needs five kinds of commands to guide the NAO robot through the tangential paths, namely, one straight forward going, two left turnings, two right turnings and one stop at the destination. While in realistic cases, more control commands are needed in the process of walking because of the limitation of the humanoid robot's flexibility. In our experiment, the number of commands is eight forward going, two left turning, five right turning and one stop, as shown in Table 2. This is because the operator needed more

Table 2. The number of the commands.

Case	Command			
	F	S	L	R
Best case	5	1	2	2
Case based on multi-sensor fusion	8	1	2	5

commands to adjust the deviation of the robot when finishing the navigation task.

Discussion

The method proposed in this article is a novel scheme based on multi-sensor fusion of Google Glass. It is the integration of wearable technology and robot control. Thus, the operator has a feeling of interaction with the expanding reality glass and experiences hand-free control missions. Meanwhile, a new channel to convey the control intention is mined, and the source of control signal is enriched. It makes it possible to accomplish a variety of control missions at the same time. Moreover, there are some significant advantages for this method as follows:

- (i) Compared with the traditional control system which relied on the computer monitor to watch the video feedback, the image of Google Glass is directly projected on the operator's retina. Therefore, this system fits the human's habits better, gives the operator a stronger sense of immersion, alleviates the burden of control and improves the control efficiency.
- (ii) Compared with the joystick control system without video feedback, the controlled robot does not have to be in sight.
- (iii) Compared with the control system relying on a handheld mobile device, the control system is able to navigate the humanoid robot with the operator's hands released and relieve hand fatigue due to operating equipment for a long period.
- (iv) Compared with the control system based on a rotation vector sensor, the head gesture reference drift does not occur, and the user experience has been greatly improved.
- (v) The system based on Google Glass is more portable due to the wearable feature, and the switching of different control commands is smoother and more natural.
- (vi) When facing a complex task, the operator can easily achieve robot remote control without the need for his/her hands. At the same time, it is understood that his/her hands can do other jobs.
- (vii) It is significant for people with paralysed limbs that this system can help them do many tasks which would normally depend upon their limbs.

This research intends not only to test the possibility of using a Google Glass to remotely control a humanoid robot but also inspires and accumulates data on how to develop a hands-free control system for those with paralysed limbs. In the future, we will explore the human machine interface based on head gestures with the integration of other interaction methods, for example, voice controls to make robot control easier and more convenient.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the National Natural Science Foundation of China under the grant 61473207.

References

1. Liu HP, Yu YL, Sun FC, et al. Visual-tactile fusion for object recognition. *IEEE Trans Autom Sci Eng* 2016; 1–13. DOI: 10.1109/TASE.2016.2549552.
2. Liu HP, Guo D and Sun FC. Object recognition using tactile measurements: kernel sparse coding methods. *IEEE Trans Instrum Meas* 2016; 65(3): 656–665.
3. Liu HP, Liu YH and Sun FC. Robust exemplar extraction using structured sparse coding. *IEEE Trans Neural Netw Learn Syst* 2015; 26(8): 1816–1821.
4. Li THS, Lin CJ, Kuo PH, et al. Grasping posture control design for a home service robot using an ABC-based adaptive PSO algorithm. *Int J Adv Robot Syst* 2016. DOI: 10.5772/64044.
5. Zhang SH, Wang DX, Zhang YR, et al. The human machine interface implementation for the robot assisted endoscopic surgery system. In: *Proceeding of international workshop on robot and human interactive communication*, 2002, pp. 442–447. DOI: 10.1109/ROMAN.2002.1045662.
6. Speers A, Forooshani PM, Dicke M, et al. Lightweight tablet devices for command and control of ROS-enabled robots. In: *Proceeding of international conference on advanced robotics*, 2013, pp. 1–6. DOI: 10.1109/ICAR.2013.6766481.
7. Ochiai Y, Takemura K, Ikeda A, et al. Remote control system for multiple mobile robots using touch panel interface and autonomous mobility. In: *Proceeding of international conference on intelligent robots and systems*, 2014, pp. 3272–3277. DOI: 10.1109/IROS.2014.6943017.
8. Chien SY, Lewis M, Mehrotra S, et al. Task switching for supervisory control of multi-robot teams. *IEEE Transactions on Human-Machine Systems* 2016.
9. Wang HD, Chien SY, Lewis M, et al. Human teams for large scale multirobot control. In: *Proceeding of international conference on systems, man and cybernetics*, 2009, pp. 1269–1274. DOI: 10.1109/ICSMC.2009.5345912.

10. Ball D and Wyeth G. Multi-robot control in highly dynamic, competitive environments. *Robocup Robot Soccer World Cup VII 2004*; 3020: 385–396.
11. Zhang J, Li W, Yu JC, et al. Virtual platform of a manned submersible vehicle carrying an underwater manipulator. In: *Proceeding of the 5th IEEE international conference on cyber technology in automation, control, and intelligent systems*, Shenyang, China, 8–12 June 2015, pp. 1126–1131.
12. Liu KZ, Zhu PQ, Zhao Y, et al. Research on the control system of human occupied vehicle “Jiaolong”. *Sci China Bull* 2013; 58: 40–48.
13. Huang PF, Wang DK, Meng ZJ, et al. Impact dynamic modelling and adaptive target capturing control for tethered space robots with uncertainties. *IEEE/ASME Transactions on Mechatronics* 2016; 21(5): 2260–2271.
14. Huang PF, Zhang F, Xu XD, et al. Coordinated coupling control of tethered space robot using releasing characteristics of space tether. *Adv Space Res* 2016; 57: 1528–1542.
15. Cartmell MP and McKenZie DJ. A review of space tether research. *Prog Aerosp Sci* 2008; 44: 1–21.
16. <http://www.boundlessat.com/SmartNav-4>
17. Gulrez T, Tognetti A, Yoon WJ, et al. A hands-free interface for controlling virtual electric-powered wheelchairs. *Int J Adv Robot Syst* 2016; pp. 1–10. DOI: 10.5772/62028.
18. Yi L, Maeda Y and Hashimoto M. Lightweight, soft variable stiffness gel spats for walking assistance. *Int J Adv Robot Syst* 2015, 12(12): 175.
19. Jeroudi OM, Christakopoulos G, Christopoulos G, et al. Accuracy of remote electrocardiogram interpretation with the use of Google Glass technology. *Am J Cardiol* 2015; 115: 374–377.
20. Rahimy E and Garg SJ. Google glass for recording scleral buckling surgery. *Jama Ophthalmology* 2015; 133: 710–711.
21. Wille M, Wischniewski S, Scholl P.M, et al. Comparing google glass with Tablet-PC as guidance system for assembling tasks. In: *Proceeding of international conference on wearable and implantable body sensor networks workshops*, 2014, pp. 38–41. DOI: 10.1109/BSN.Workshops.2014.11.
22. Weppner J, Hirth M, Lukowicz P, et al. Physics education with Google Glass gPhysics experiment app. In: *ACM international joint conference on pervasive and ubiquitous computing: adjunct publication*, Seattle, WA, USA, 13–17 September 2014, pp. 279–282.
23. Kapellmann-Zafra G, Chen JN and Gro BR. Using Google Glass in human Robot swarm interaction. In: *Proceedings towards autonomous robotic systems*, Sheffield, UK, 26 June–1 July 2016; pp. 196–201.
24. Wen X, Song Y, Li W, et al. Rotation vector sensor-based remote control of a mobile robot via google glass. In: *Robot intelligence technology and applications 4*, Springer International Publishing, 2015; pp. 581–588. DOI: 10.1007/978-3-319-31293-4_47.
25. Wen X, Song Y, Li W, et al. Rotation vector sensor-based remote control of a humanoid robot through a Google Glass. In: *Proceeding of international workshop on advanced motion control*, Auckland, New Zealand, 22–24 April 2016, pp. 211–215.
26. Song Y, Wen X, Li W, et al. Google Glass-based remote control of a mobile robot. In: *Sensors and systems for space applications IX*, Baltimore, Maryland, USA, 13 May 2016.
27. <https://developers.google.com/glass/design/>
28. Lins CN, Teixeira JM, Roberto RA, et al. Development of interactive applications for Google Glass. *Tendências e Técnicas em Realidade Virtual e Aumentada* 2014; 4: 167–188.
29. Scheffel J and Kockesen G. Wearable web technology: google glass and the mirror API.
30. http://wear.techbrood.com/guide/topics/sensors/sensors_motion.html
31. Gouaillier D, Hugel V, Blazevic P, et al. The NAO humanoid: a combination of performance and affordability. *Computer Science* 2008; abs/0807.3223.
32. Kulk J and Welsh J. A low power walk for the NAO robot. In: *Proceeding of the 2008 Australasian Conference on Robotics & Automation (ACRA-2008)* (eds J Kim and R Mahony), 2008.
33. <http://www.aldebaran-robotics.com/en/>
34. <http://www.robotappstore.com>
35. Milette G and Stroud A. Professional android sensor programming. *Wrox Press Ltd* 2012.
36. Wang T and Niu YF. Sensor fusion technology to achieve by complementary filter. *Computer Knowledge and Technology* 2014; 5950–5954.