

Real-Time Robotic Hand Control Using Hand Gestures

Jagdish Lal Raheja, Radhey Shyam, G. Arun Rajsekhar and P. Bhanu Prasad
*Digital Systems Group, Central Electronics Engineering Research Institute
(CEERI)/Council of Scientific & Industrial Research (CSIR), Pilani, Rajasthan
India*

1. Introduction

Interpretation of human gestures by a computer is used for human-machine interaction in the area of computer vision [3][28]. The main purpose of gesture recognition research is to identify a particular human gesture and convey information to the user pertaining to individual gesture. From the corpus of gestures, specific gesture of interest can be identified[30][36], and on the basis of that, specific command for execution of action can be given to robotic system[31]. Overall aim is to make the computer understand human body language [14], thereby bridging the gap between machine and human. Hand gesture recognition can be used to enhance human-computer interaction without depending on traditional input devices such as keyboard and mouse.

Hand gestures are extensively used for telerobotic control applications [20]. Robotic systems can be controlled naturally and intuitively with such telerobotic communication [38] [41]. A prominent benefit of such a system is that it presents a natural way to send geometrical information to the robot such as: left, right, etc. Robotic hand can be controlled remotely by hand gestures. Research is being carried out in this area for a long time. Several approaches have been developed for sensing hand movements and controlling robotic hand [9][13][22]. Glove based technique is a well-known means of recognizing hand gestures. It utilizes sensor attached mechanical glove devices that directly measure hand and/or arm joint angles and spatial position. Although glove-based gestural interfaces give more precision, it limits freedom as it requires users to wear cumbersome patch of devices. Jae-Ho Shin et al [14] used entropy analysis to extract hand region in complex background for hand gesture recognition system. Robot controlling is done by fusion of hand positioning and arm gestures [2][32] using data gloves. Although it gives more precision, it limits freedom due to necessity of wearing gloves. For capturing hand gestures correctly, proper light and camera angles are required. A technique to manage light source and view angles, in an efficient way for capturing hand gestures, is given in [26]. Reviews have been written on the subsets of hand postures and gesture recognition in [6]and [18]. The problem of visual hand recognition and tracking is quite challenging. Many early approaches used position markers or colored bands to make the problem of hand recognition easier, but due to their inconvenience, they cannot be considered as a natural interface for the robot control [24]. A 3D Matlab Kinematic model of a PUMA [1][4][16] robot, is used for executing actions by hand gesture[39]. It can be extended to any robotic system with a number of specific commands suitable to that system [17].

In this chapter, we have proposed a fast as well as automatic hand gesture detection and recognition system. Once a hand gesture is recognised, an appropriate command is sent to a robot. Once robot receives a command, it does a pre-defined work and keeps doing until a new command arrives. A flowchart for overall controlling of a robot is given in figure 1.

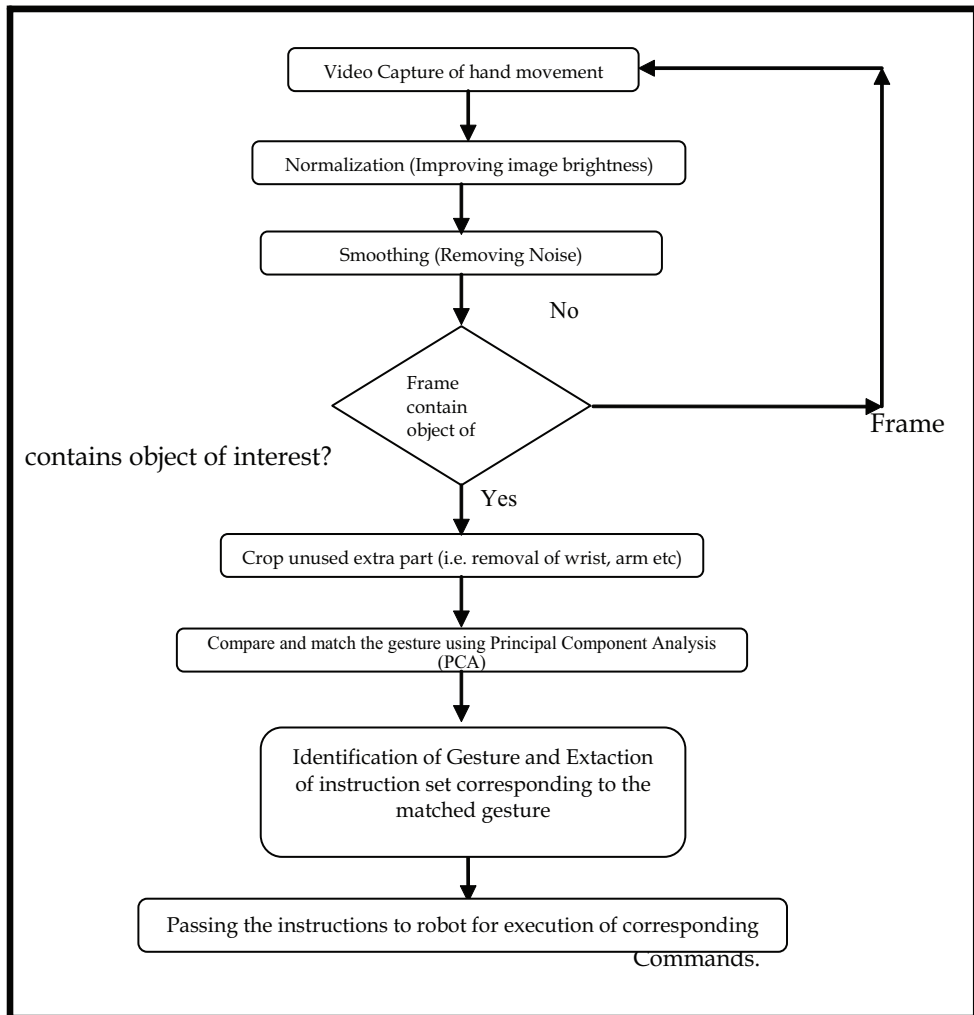


Fig. 1. Flow diagram for robot control using hand gestures

2. Methodology

The proposed methodology involves acquisition of live video from a camera for gesture identification. It acquires frames from live video stream in some given time interval [15]. In this case, the frame capture rate for gesture search is 3 frames per second. The overall proposed

technique to acquire hand gestures and to control robotic system using hand gestures is divided into four subparts:

- Image acquisition to get hand gestures.
- Extracting hand gesture area from captured frame.
- Determining gesture by pattern matching using PCA algorithm.
- Generation of instructions corresponding to matched gesture, for specified robotic action.

2.1 Image acquisition to get hand gestures

First step is to capture the image from video stream. It is assumed that while capturing video, black background with proper light source is used. To minimize number of light sources, arrangement of the light sources and camera is made as shown in figure 2.

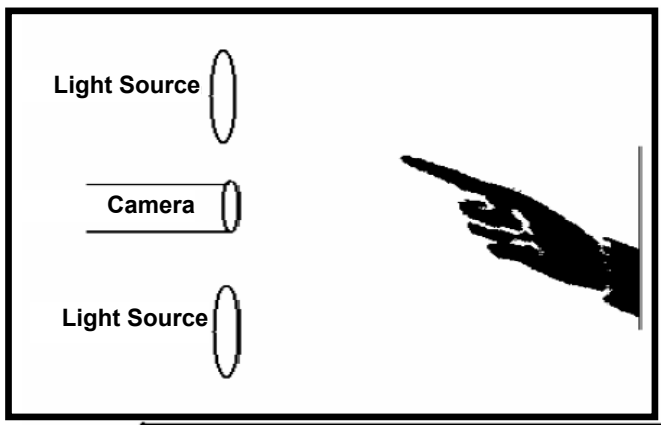


Fig. 2. Arrangement of light sources for image acquisition

From video stream one frame is captured every 1/3 of a second. Our aim is to identify the frame that contains hand gesture shown by human. For this we are searching the frame in which there is no or least movement. Required frame is identified by comparing three continuously captured frames. *Motion parameter* is determined for each frame by counting total pixels of mismatch. If *motion parameter* is less than the specified threshold value, it is considered as a frame having least movement of object i.e. the hand of the person.

Analysis of frames to find the frame of interest is carried out by converting the captured frame into a binary frame [23]. Since binary images have values of one or zero, differences of white pixels between newly captured frame and two previously captured frames are determined and they are added together to find *motion parameter*. XOR function is used to find the differences in white pixels [5][15].

If *frame1*, *frame2*, and, *frame3* are three matrices containing three frames captured in three continuous time slots respectively then:

$$fr1 = frame1 \text{ XOR } frame3 \quad (1)$$

$$fr2 = frame2 \text{ XOR } frame3 \quad (2)$$

$$\text{mismatch_matrix} = \text{fr1 OR fr2} \quad (3)$$

$$\text{Sum} = \sum_{i=1}^r \sum_{j=1}^c \text{mismatch_matrix}[i][j] \quad (4)$$

$$\text{Motion parameter} = \frac{\text{sum}}{r * c} \quad (5)$$

Here r and c are the number of rows and columns in an image frame. Threshold value is set as 0.01. i.e. if total pixels of mismatch are less than 1% of total pixels in a frame, then it is considered as frame of interest. Required frame is forwarded for further processing and this module again starts searching for next frame of interest.

2.2 Extracting hand gesture area from frame of interest

It may happen that the frame, as shown in fig.3, with a gesture contains extra part along with required part of hand i.e. background objects, blank spaces etc. For better results in pattern matching, unused part must be removed. Therefore hand gesture is cropped out from the obtained frame. Cropping of hand gesture from the obtained frame contains three steps:



Fig. 3. Grayscale image

First step is to convert selected frame into HSV color space. As shown in fig.4, a HSV based skin filter was applied on the RGB format image to reduce lighting effects [8][1]. The thresholds for skin filter used in the experiment were as below:

$$\left. \begin{array}{l} 0.05 < H < 0.17 \\ 0.1 < S < 0.3 \\ 0.09 < V < 0.15 \end{array} \right\} \rightarrow \quad (6)$$

The resultant image was segmented to get a binary image of hand[37]. Binary images are bi-level images where each pixel is stored as a single bit (0 or 1). Smoothing was needed, as the output image had some jagged edges as clearly visible in figure 4(c). There can be some noise in the filtered images due to false detected skin pixels or some skin color objects (like wood) in background, it can generate few unwanted spots in the output image as shown in figure 4(e).

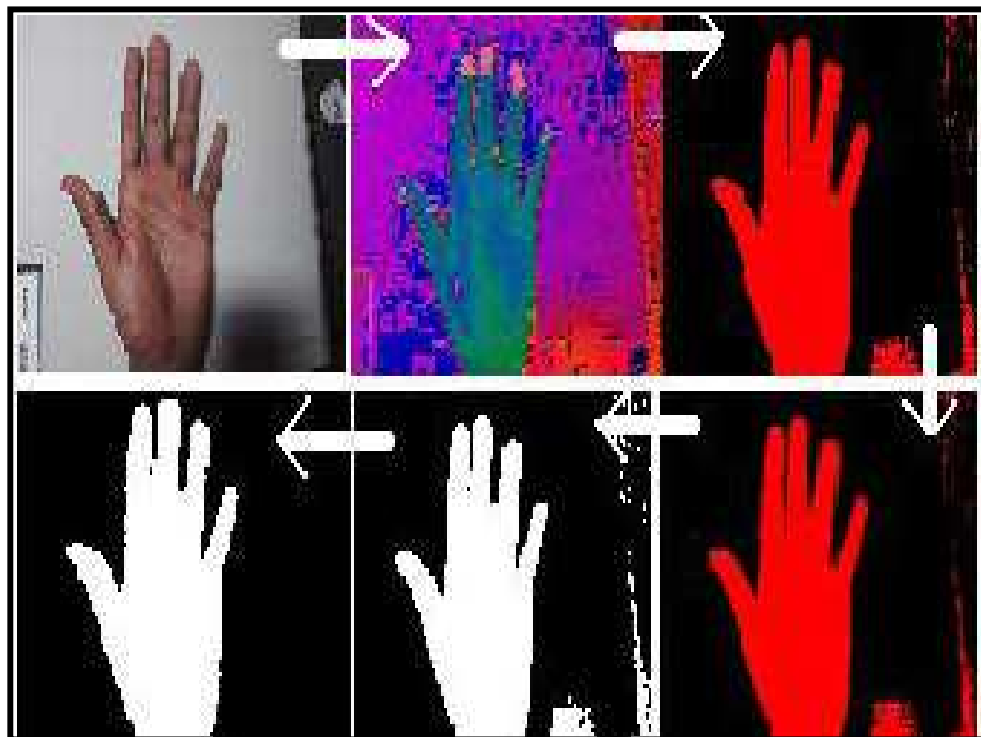


Fig. 4. Binary Image formation (a) Target image (b) HSV conversion (c) Filtered image (d) Smoothened image (e) Binary image (f) Biggest BLOB

To remove these errors, the biggest BLOB (Binary Linked Object) method was applied to the noisy image [24][25]. The error free image is shown in figure 5. The only limitation of this filter was that the BLOB for hand should be the biggest one. In this, masking background would be illuminated, so false detection is not possible from the background [7][10].

Second step is to extract object of interest from the frame. In our case, object of interest is the part of human hand showing gesture [29]. For this, extra part, other than the hand, is cropped out so that pattern matching can give more accurate results [12]. For cropping extra parts row and column numbers are determined, from where object of interest appears. This is done by searching from each side of binary image and moving forward until white pixels encountered are more than the offset value [25]. Experimental results show that offset value set to 1% of total width gives better result for noise compensation. If size of selected image is $m \times n$ then:

$$Hor_offset = m/100 \quad (7)$$

$$Ver_offset = n/100 \quad (8)$$

Min_col = minimum value of column number where total number of white pixels are more than Hor_offset .

Max_col = maximum value of column number where total number of white pixels are more than Hor_offset .

Min_row= minimum value of row number where total number of white pixels are more than *Ver_offset*.

Max_row= maximum value of row number where total number of white pixels are more than *Ver_offset*.

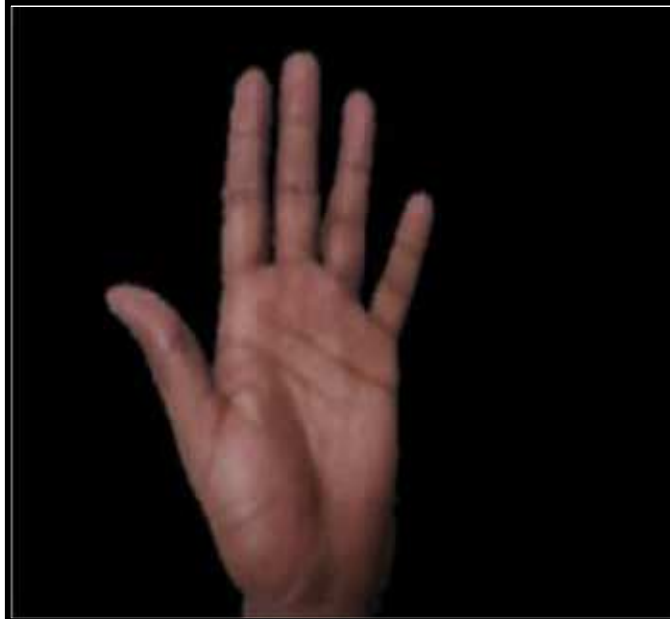


Fig. 5. Error free image

Figure 6 shows *Min_row*, *Max_row*, *Min_col*, and, *Max_col* for the selection of the boundary for cropping of hand.

Third step is to remove parts of hand not used in gesture i.e. removal of wrist, arm etc. As these extra parts are of variable length in image frame and pattern matching with gesture database gives unwanted results. Therefore, parts of hand before the wrist need to be cropped out.

Statistical analysis of hand shape shows that either we pose palm or fist, width is lowest at wrist and highest at the middle of palm. Therefore extra hand part can be cropped out from wrist by determining location where minimum width of vertical histogram is found. Figure 7.c and 7.d show global maxima and cropping points for hand gestures in figure 7.a and 7.b respectively.

Cropping point is calculated as:

Global Maxima = column number where height of histogram is highest (i.e. column number for global maxima as shown in figure 7).

Cropping point = column number where height of histogram is lowest in such a way that cropping point is in between first column and column of *Global Maxima*

If gesture is shown from opposite side (i.e. from other hand), then *Cropping point* is searched between column of *Global Maxima* and last column. Direction of the hand is detected using continuity analysis of object during hand gesture area determination. Continuity analysis

shows that whether the object continues from the column of Global maxima to first column or to last column. i.e. whether extra hand is left side of palm or right side of palm.



Fig. 6. Boundary selection for hand cropping

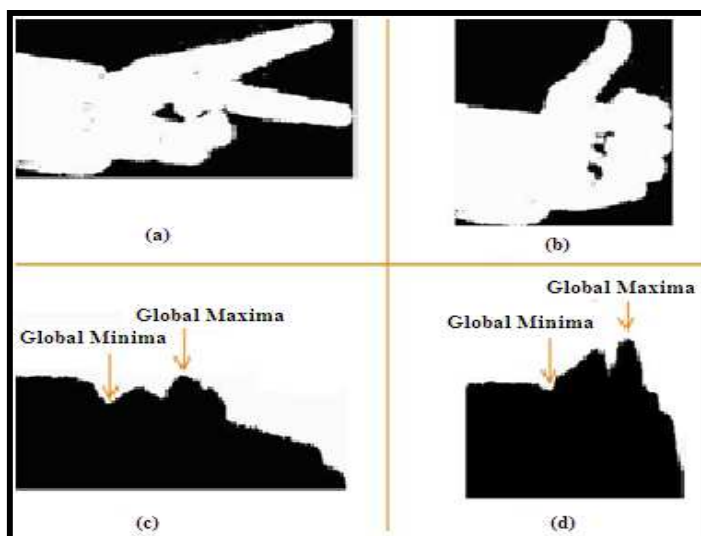


Fig. 7. Histogram showing white pixels in each column, with cropping point for hand gesture

2.3 Determining the gesture by resizing and pattern matching

Cropping results in a frame of 60x80 pixels, so that the gesture can be compared with the database. This particular size of 60x80 is chosen to preserve the aspect ratio of the original image, so that no distortion takes place.

Figure 8.b shows cropped hand gesture from figure 8.a. and figure 8.c. shows the result of scaling and fitting 8.b within 60x80 pixels. Figure 8.d represents final cropped and resized gesture of desired dimension.

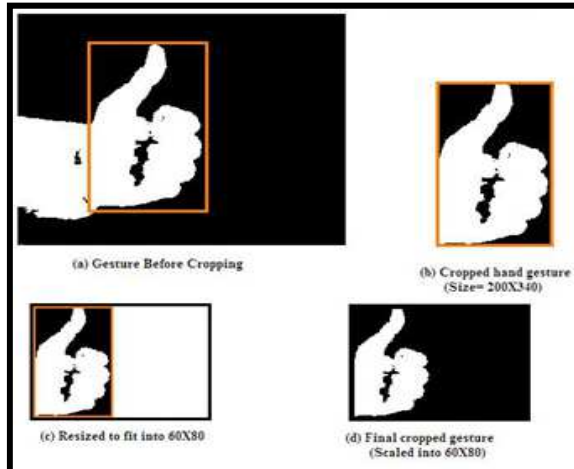


Fig. 8. Scaling cropped hand gesture to fit into desired dimension

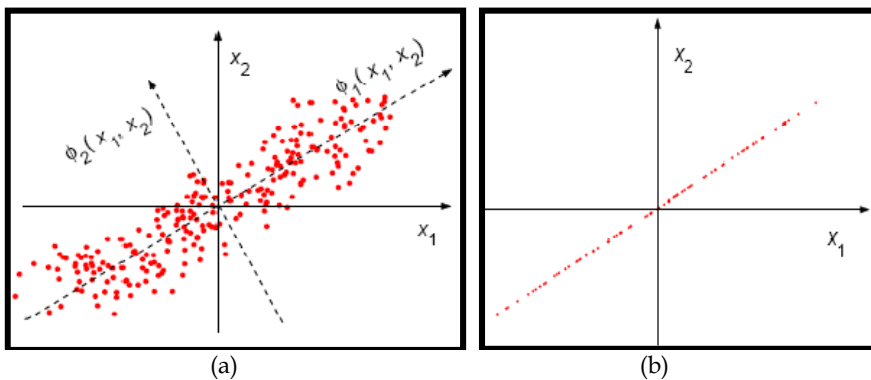
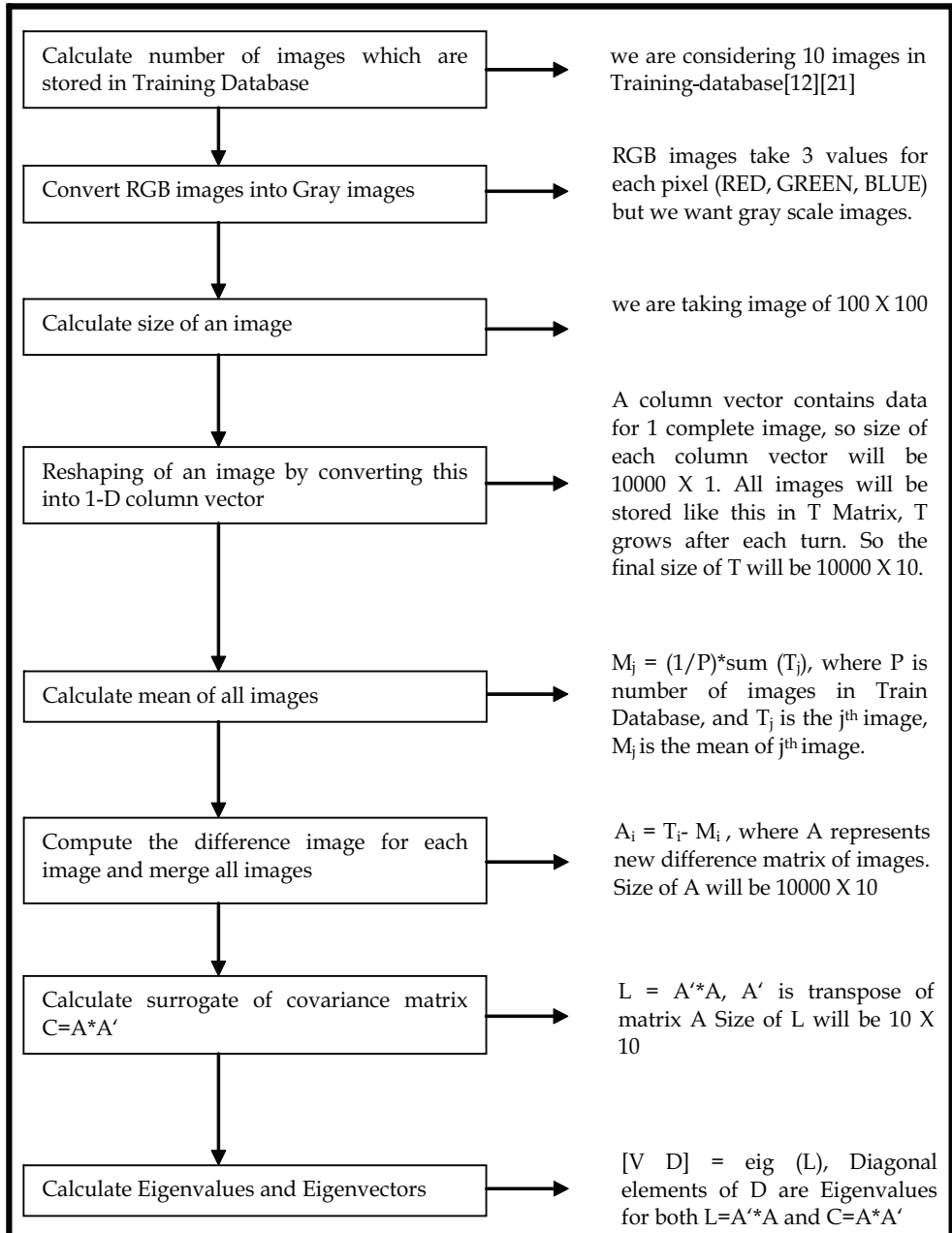


Fig. 9. (a) PCA basis (b) PCA reduction to 1D

(a) The concept of PCA. Solid lines: the original basis; dashed lines: the PCA basis. The dots are selected at regularly spaced locations on a straight line rotated at 30degrees, and then perturbed by isotropic 2D Gaussian noise. (b) The projection (1D reconstruction) of the data using only the first principal component.

There are many different algorithms available for Gesture Recognition, such as principal Component Analysis, Linear Discriminate Analysis, Independent Component Analysis, Neural Networks and Genetic Algorithms.

Principal Component Analysis (PCA)[11] is used in this application to compare the acquired gesture with the gestures available in the database and recognize the intended gesture.



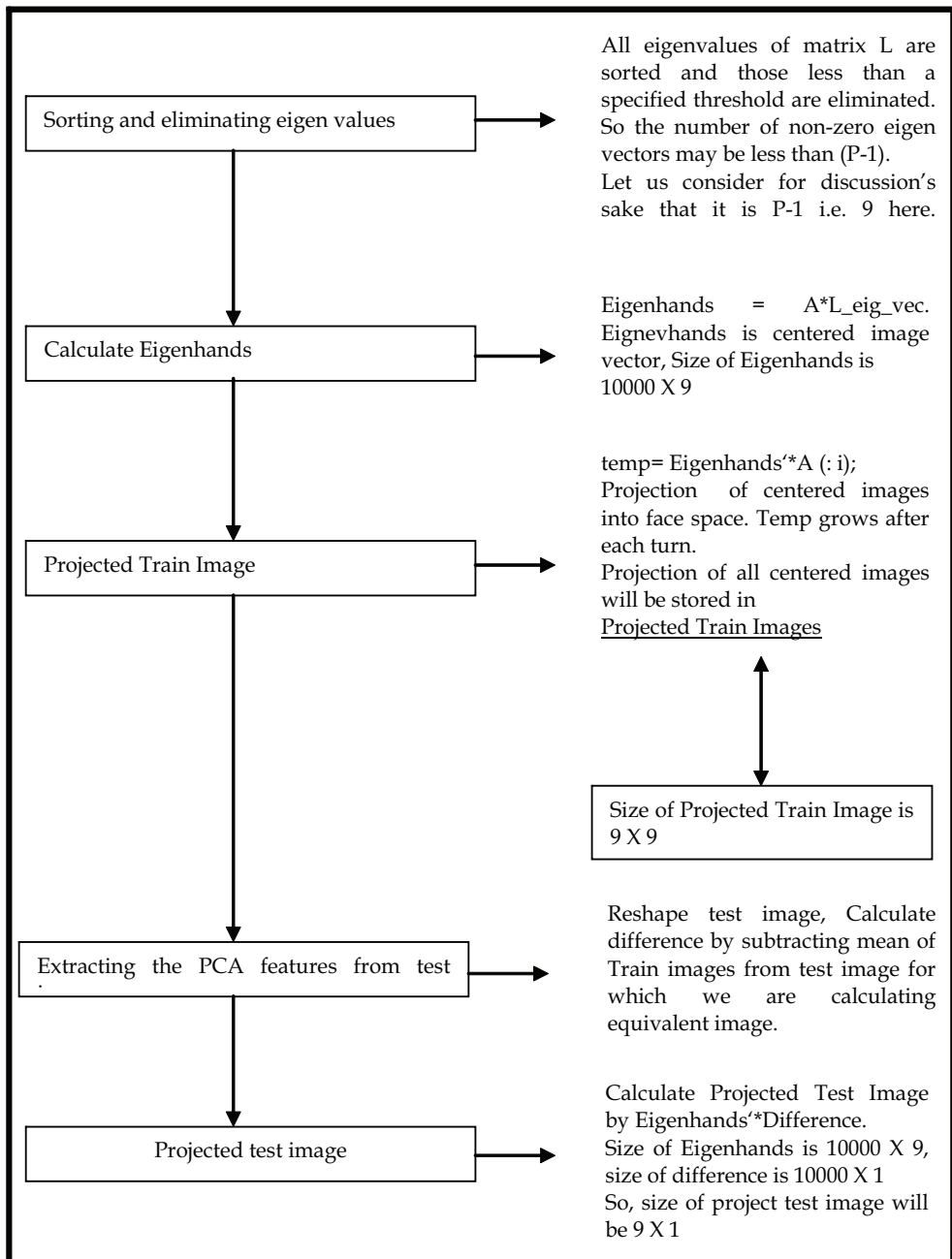


Fig. 10. PCA Algorithm steps for 10 Images

The concept of PCA is illustrated in Figure 9; The graph shows data that originally has two components along the x_1 and x_2 directions. This is now resolved along the Φ_1 and Φ_2 directions. Φ_1 corresponds to the direction of maximum variance and is chosen as the first principal component. In the 2D case currently being considered, the second principal component is then determined uniquely by orthogonality constraints; in a higher-dimensional space the selection process would continue, guided by the variances of the projections.

A comprehensive explanation of the above scheme for gesture recognition[19] within a database of ten images is given in figure 10. For simplicity of calculations and ease of understanding, each figure is taken to be of size of 100X100 pixels. This algorithms was tested on a PC platform.

2.4 Generation of control Instructions for a robot

Different functions corresponding to each meaningful hand gesture are written and stored in database for controlling the robotic Arm. For implementation of robotic arm control, PUMA robotic model has been chosen. Since this model was only available at the time of interface, hand gestures are used to control the various joints of the arm. However, these gestures can be used to control the robotic hand also. Altogether 10 different hand gestures are related to 10 different movements of robotic arm. Movement commands are written as a function in robot specific language. Whenever a gesture is matched with a meaningful gesture from the database, the instruction set corresponding to that gesture is identified and passed to robot for execution. In this way the robotic system can be controlled by hand gesture using live camera. Figure 11 shows movements of robotic hand corresponding to four different specific hand gestures.

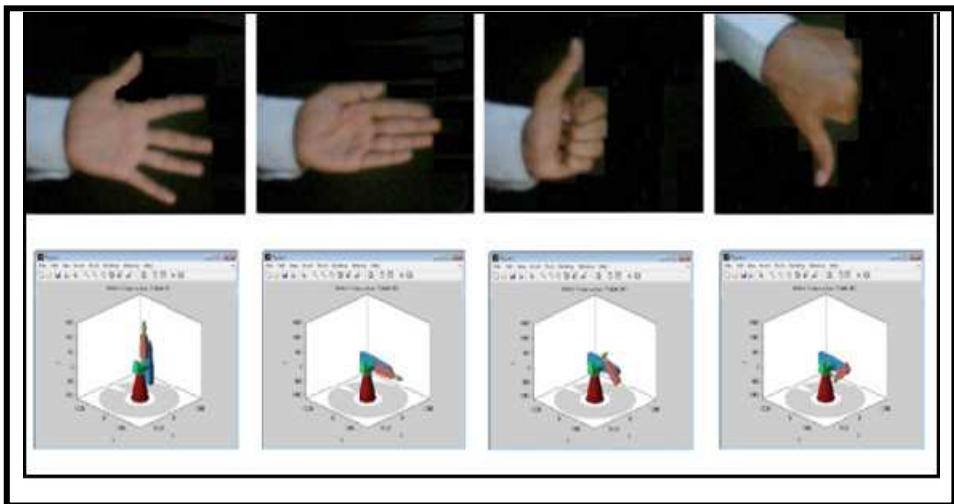


Fig. 11. Movements of PUMA 762 corresponding to some specific hand gestures

However, there are a total of 10 actionable gestures, and one background image for resetting purpose that have been identified as shown in fig. 12.

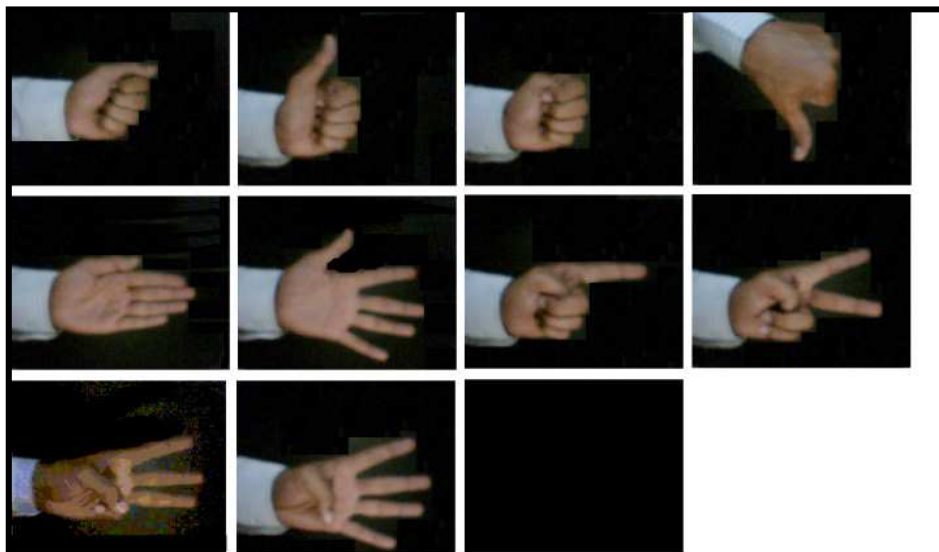


Fig. 12. Set of specified gestures used for robot control

3. Implementation on FPGA processor

We have implemented the algorithm on an FPGA, using a Database of 6 images. Each image size is 60X80 pixels. The reduction in number of images, image size and its resolution are due to FPGA memory constraints. The MATLAB code produces 6x6 projection vectors which are stored in the RAM of the FPGA using a JTAG cable [27]. After formation of the database, the system captures an image and projection vector of 6x1 is produced and that is given to the hardware for further processing. The hardware is responsible for calculating the Euclidian Distance [ED] with a dedicated block which we designate as ED block [34]. Then the minimum distance [MD] is found using a dedicated block named MD block. And the index corresponding to the minimum distance is returned to MATLAB to display the matched image from the Database [33]. The whole process is shown in fig. 13.

The main drawback of this implementation is that only part of the algorithm resides on FPGA and rest of the algorithm still runs on PC. Therefore it is not standalone system and requires interface with the PC[13]. FPGA implementation of PCA algorithm is a great challenge due to the limited resources available in commercial FPGA kits. But it is quite important to have a complete system on a FPGA chip [40]. As far as the technology is concerned, FPGA is most suited for implementation of real time image processing algorithm. It can be readily designed with custom parallel digital circuitry tailored for performing various imaging tasks making them well-suited for high speed real time vision processing applications[35].

4. Conclusion

Controlling a robot arm, in real time, through the hand gestures is a novel approach. The technique proposed here was tested under proper lighting conditions that we created in our

laboratory. A gesture database consisting of binary images of size 60 X 80 pixels is pre-stored, so it takes less time and memory space during pattern recognition. Due to the use of cropped image of gesture, our database becomes more effective as one image is sufficient for one type of gesture presentation. So, neither we need to store more than one image for same gesture at different positions of image, nor have we to worry about positions of hand in front of camera. Experimental results show that the system can detect hand gestures with an accuracy of 95 % when it is kept still in front of the camera for 1 second.

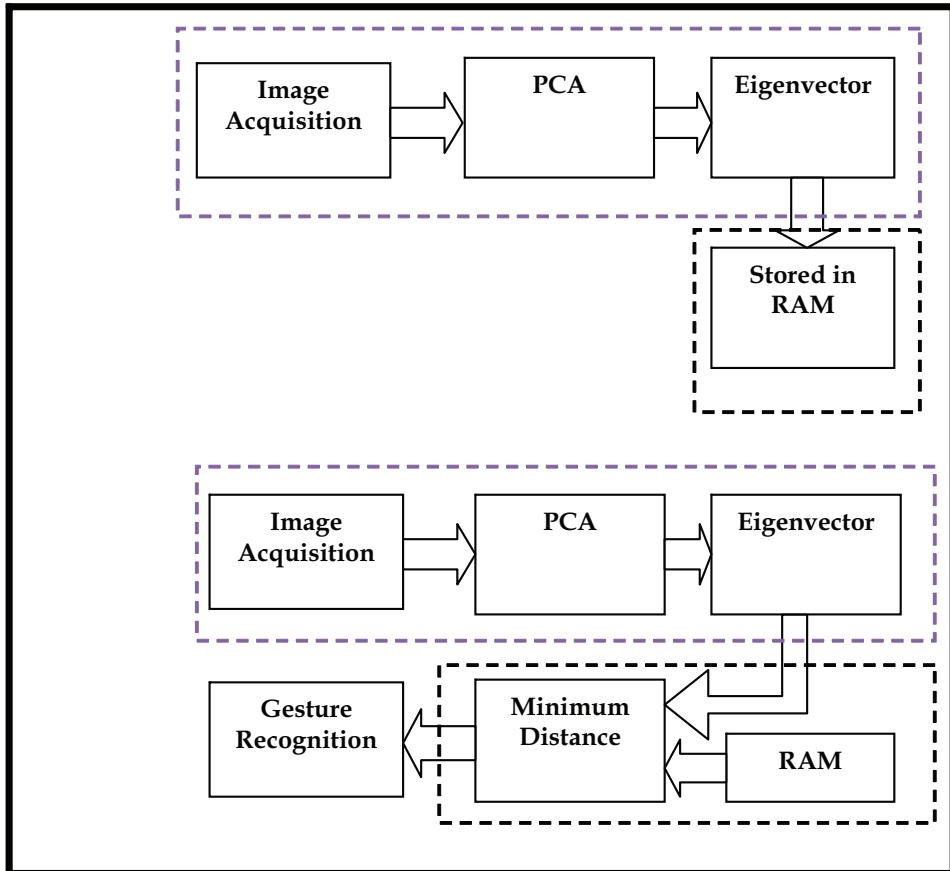


Fig. 13. Recognition process

5. Acknowledgment

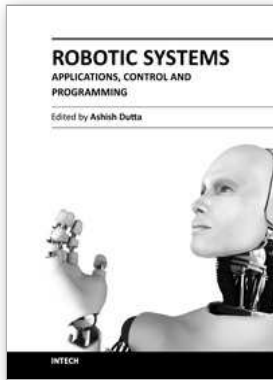
This research is being carried out at Central Electronics Engineering Research Institute (CEERI), Pilani, India as part of a project "Supra Institutional Projects on Technology Development for Smart Systems". Authors would like to thank Director, CEERI for his active encouragement and support.

6. References

- [1] Becerra V.M., Cage, C.N.J., Harwin, W.S., Sharkey, P.M. Hardware retrofit and computed torque control of Puma 560 robot updating an industrial manipulator. *IEEE Control Systems Magazine*, 24(5): 78-82. 2004.
- [2] L. Brthes, P. Menezes, F. Lerasle, and J. Hayet. Face tracking and hand gesture recognition for human robot interaction. In *International Conference on Robotics and Automation*, pp. 1901-1906, 2004. New Orleans, Louisiana.
- [3] Chan Wah Ng, Surendra Ranganath, Real-time gesture recognition system and application. *Image and Vision computing* (20): 993-1007, 2002.
- [4] Corke, P.I. Operational Details of the Unimation Puma Servo System. Report, CSIRO Division of Manufacturing Technology. Australia, 1993.
- [5] Crow, F. Summed-area tables for texture mapping. *SIGGRAPH '84: Proceedings of the 11th annual conference on Computer graphics and interactive techniques*. pp. 207-212.
- [6] J. Davis, M. Shah. Recognizing hand gestures. In *Proceedings of European Conference on Computer Vision, ECCV: 331-340*, 1994. Stockholm, Sweden.
- [7] Daggi Venkateshwar Rao, Shruti Patil, Naveen Anne Babu and V Muthukumar, Implementation and Evaluation of Image Processing Algorithms on Reconfigurable Architecture using C-based Hardware Descriptive Languages, *International Journal of Theoretical and Applied Computer Sciences*, Volume 1 Number 1 pp. 9-34, 2006.
- [8] Dastur, J.; Khawaja, A.; Robotic Arm Actuation with 7 DOF Using Haar Classifier Gesture Recognition, *Second International Conference on Computer Engineering and Applications (ICCEA)*, vol.1, no., pp.27-29, 19-21 March 2010.
- [9] K G Derpains, A review of Vision-based Hand Gestures, Internal Report, Department of Computer Science. York University, February 2004.
- [10] Didier Coquin, Eric Benoit, Hideyuki Sawada, and Bogdan Ionescu, Gestures Recognition Based on the Fusion of Hand Positioning and Arm Gestures, *Journal of Robotics and Mechatronics* Vol.18 No.6, 2006. pp. 751-759.
- [11] Fang Zhong, David W. Capson, Derek C. Schuurman, Parallel Architecture for PCA Image Feature Detection using FPGA 978-1-4244-1643-1/08/2008 IEEE.
- [12] Freeman W., Computer vision for television and games. In *Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, page 118, 1999. Copyright© MEITCA.
- [13] Haessig D., Hwang J., Gallagher S., Uhm M., Case-Study of a Xilinx System Generator Design Flow For Rapid Development of SDR Waveforms, *Proceeding of the SDR 05 Technical Conference and Product Exposition*. Copyright © 2005 SDR Forum.
- [14] Jae-Ho Shin, Jong-Shill Lee, Se-Kee Kil, Dong-Fan Shen, Je-Goon Ryu, Eung-Hyuk Lee, Hong-Ki Min, Seung-Hong Hong, Hand Region Extraction and Gesture Recognition using entropy analysis, *IJCSNS International Journal of Computer Science and Network Security*, VOL.6 No.2A, 2006 pp. 216-222.
- [15] Jagdish Raheja, Radhey Shyam and Umesh Kumar, Hand Gesture Capture and Recognition Technique for Real-time Video Stream, In *The 13th IASTED International Conference on Artificial Intelligence and Soft Computing (ASC 2009)*, September 7 - 9, 2009 Palma de Mallorca, Spain.

- [16] Katupitiya, J., Radajewski, R., Sanderson, J., Tordon, M. Implementation of PC Based Controller for a PUMA Robot. Proc. 4th IEEE Conf. on Mechatronics and Machine Vision in Practice. Australia, p.14-19. [doi:10.1109/MMVIP.1997.625229], 1997.
- [17] R. Kjeldsen, J. Kinder. Visual hand gesture recognition for window system control, in International Workshop on Automatic Face and Gesture Recognition (IWAAGR), Zurich: pp. 184-188, 1995.
- [18] Mohammad, Y.; Nishida, T.; Okada, S.; Unsupervised simultaneous learning of gestures, actions and their associations for Human-Robot Interaction, Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on, vol., no., pp.2537-2544, 10-15 Oct. 2009.
- [19] Moezzi and R. Jain, "ROBOGEST: Telepresence Using Hand Gestures". Technical report VCL-94-104, University of California, San Diego, 1994.
- [20] M. Moore, A DSP-based real time image processing system. In the Proceedings of the 6th International conference on signal processing applications and technology, Boston MA, August 1995.
- [21] Nolker, C.; Ritter, H.; Parametrized SOMs for hand posture reconstruction, Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on, vol.4, no., pp.139-144 vol.4, 2000
- [22] Nolker C., Ritter H., Visual Recognition of Continuous Hand Postures, IEEE Transactions on neural networks, Vol 13, No. 4, July 2002, pp. 983-994.
- [23] K. Oka, Y. Sato and H. Koike. Real-Time Fingertip Tracking and Gesture Recognition. IEEE Computer Graphics and Applications: 64-71, 2002.
- [24] Ozer, I. B., Lu, T., Wolf, W. Design of a Real Time Gesture Recognition System: High Performance through algorithms and software. IEEE Signal Processing Magazine, pp. 57-64, 2005.
- [25] V.I. Pavlovic, R. Sharma, T.S. Huang. Visual interpretation of hand gestures for human-computer interaction, A Review, IEEE Transactions on Pattern Analysis and Machine Intelligence 19(7): 677-695, 1997.
- [26] Quek, F. K. H., Mysliwiec, T., Zhao, M. (1995). Finger Mouse: A Free hand Pointing Computer Interface. Proceedings of International Workshop on Automatic Face and Gesture Recognition, pp.372-377, Zurich, Switzerland.
- [27] A.K. Ratha, N.K. Jain, FPGA-based computing in computer vision, Computer Architecture for Machine Perception, CAMP '97. Proceedings Fourth IEEE International Workshop on, pp. 128-137, 20-22 Oct 1997.
- [28] Ramamoorthy, N. Vaswani, S. Chaudhury, S. Banerjee, Recognition of Dynamic hand gestures, Pattern Recognition 36: 2069-2081, 2003.
- [29] Sawah A.E., and et al., a framework for 3D hand tracking and gesture recognition using elements of genetic programming, 4th Canadian conference on Computer and robot vision, Montreal, Canada, 28-30 May, 2007, pp. 495-502.
- [30] Sergios Theodoridis, Konstantinos Koutroumbas, Pattern Recognition, Elsevier Publication, Second Edition, 2003.
- [31] Seyed Eghbal Ghabadi, Omar Edmond Loepprich, Farid Ahmadov Jens Bernshausen, Real Time Hand Based Robot Control Using Multimodal Images, IAENG International Journal of Computer Science, 35:4, IJCS_35_4_08, Nov 2008.

- [32] Shin M. C., Tsap L. V., and Goldgof D. B., Gesture recognition using bezier curves for visualization navigation from registered 3-d data, *Pattern Recognition*, Vol. 37, Issue 5, May 2004, pp.1011–1024.
- [33] N. Shirazi and J. Ballagh, "Put Hardware in the Loop with Xilinx System Generator for DSP", *Xcell Journal online*, May 2003,
<http://xvwww.xilinx.com/publications/xcellonline/xcell47/xcsysgen47.htm>
- [34] Stephen D. Brown, R.J. Francis, J. Rose, Z.G. Vranesic, *Field Programmable Gate Arrays*, 1992.
- [35] Starner T., Pentland. A., Real-time American Sign Language recognition from video using hidden markov models. In *SCV95*, pages 265–270, Ames Street, Cambridge, 1995.
- [36] Triesch J., Malsburg. C.V.D., Robotic gesture recognition. In *Gesture Workshop*, pages 233–244, Nara, Japan, 1997.
- [37] Tomita, A., Ishii. R. J. Hand shape extraction from a sequence of digitized grayscale images. *Proceedings of twentieth International Conference on Industrial Electronics, Control and Instrumentation, IECON '94*, vol.3, pp.1925–1930, Bologna, Italy, 1994.
- [38] Verma R., Dev A., Vision based Hand Gesture Recognition Using finite State Machines and Fuzzy Logic, *International Conference on Ultra-Modern Telecommunications & Workshops*, 12-14 Oct, 2009, pp. 1-6, in St. Petersburg, Russia.
- [39] Walla Walla University, robotic lab PUMA 762.
<http://www.mathworks.com/matlabcentral/fileexchange/14932>
- [40] Xilinx Inc., *System Generator for DSP, Users Guide*
<http://www.xilinx.com/products/software/svstgen/appdocs/userguide.htm>
- [41] Ying Wu, Thomas S Huang, Vision based Gesture. Recognition: A Review, *Lecture Notes In Computer Science*; Vol. 1739, *Proceedings of the International Gesture Workshop on Gesture-Based Communication in Human-Computer Interaction*, 1999.



Robotic Systems - Applications, Control and Programming

Edited by Dr. Ashish Dutta

ISBN 978-953-307-941-7

Hard cover, 628 pages

Publisher InTech

Published online 03, February, 2012

Published in print edition February, 2012

This book brings together some of the latest research in robot applications, control, modeling, sensors and algorithms. Consisting of three main sections, the first section of the book has a focus on robotic surgery, rehabilitation, self-assembly, while the second section offers an insight into the area of control with discussions on exoskeleton control and robot learning among others. The third section is on vision and ultrasonic sensors which is followed by a series of chapters which include a focus on the programming of intelligent service robots and systems adaptations.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Jagdish Lal Raheja, Radhey Shyam, G. Arun Rajsekhar and P. Bhanu Prasad (2012). Real-Time Robotic Hand Control Using Hand Gestures, *Robotic Systems - Applications, Control and Programming*, Dr. Ashish Dutta (Ed.), ISBN: 978-953-307-941-7, InTech, Available from: <http://www.intechopen.com/books/robotic-systems-applications-control-and-programming/real-time-robotic-hand-control-using-hand-gestures->

INTECH

open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2012 The Author(s). Licensee IntechOpen. This is an open access article distributed under the terms of the [Creative Commons Attribution 3.0 License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.