Advances in Machine Learning and Data Analysis

Lecture Notes in Electrical Engineering

Volume 48

For other titles published in this series, go to
http://www.springer.com/series/7818

Sio-Iong Ao • Burghard B. Rieger
Mahyar Amouzegar
Editors

Advances in Machine Learning and Data Analysis

Springer

Editors
Sio-Iong Ao
Harvard School of Engineering
and Applied Sciences
Harvard University
Room 403, 60 Oxford Street
Cambridge MA 02138, USA
siao@harvard.edu

Mahyar Amouzegar
College of Engineering
California State University
Long Beach

Burghard B. Rieger
Universität Trier
FB II Linguistische
Datenverarbeitung
Computerlinguistik
Universitätsring 15
54286 Trier
Germany
publication@iaeng.org

ISSN 1876-1100 e-ISSN 1876-1119
ISBN 978-90-481-3176-1 e-ISBN 978-90-481-3177-8
DOI 10.1007/978-90-481-3177-8
Springer Dordrecht Heidelberg London New York

Library of Congress Control Number: 2009930421

Cover design: eStudio Calamar S.L.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

A large international conference on Advances in Machine Learning and Data Analysis was held in UC Berkeley, CA, USA, October 22–24, 2008, under the auspices of the World Congress on Engineering and Computer Science (WCECS 2008). The WCECS is organized by the International Association of Engineers (IAENG). IAENG is a non-profit international association for the engineers and the computer scientists, which was founded in 1968 and has been undergoing rapid expansions in recent years. The WCECS conferences have served as excellent venues for the engineering community to meet with each other and to exchange ideas. Moreover, WCECS continues to strike a balance between theoretical and application development. The conference committees have been formed with over two hundred members who are mainly research center heads, deans, department heads (chairs), professors, and research scientists from over thirty countries. The conference participants are also truly international with a high level of representation from many countries. The responses for the congress have been excellent. In 2008, we received more than six hundred manuscripts, and after a thorough peer review process 56.71% of the papers were accepted.

This volume contains sixteen revised and extended research articles written by prominent researchers participating in the conference. Topics covered include Expert system, Intelligent decision making, Knowledge-based systems, Knowledge extraction, Data analysis tools, Computational biology, Optimization algorithms, Experiment designs, Complex system identification, Computational modeling, and industrial applications. The book offers the state of the art of tremendous advances in machine learning and data analysis and also serves as an excellent reference text for researchers and graduate students, working on machine learning and data analysis.

Harvard University, USA — Sio-Iong Ao
University of Trier, Germany — Burghard B. Rieger
California State University, Long Beach, USA — Mahyar Amouzegar

Contents

Chapter 1
2D/3D Image Data Analysis for Object Tracking and Classification

Seyed Eghbal Ghobadi, Omar Edmond Loepprich, Oliver Lottner, Klaus Hartmann, Wolfgang Weihs, and Otmar Loffeld

Abstract Object tracking and classification is of utmost importance for different kinds of applications in computer vision. In this chapter, we analyze 2D/3D image data to address solutions to some aspects of object tracking and classification. We conclude our work with a real time hand based robot control with promising results in a real time application, even under challenging varying lighting conditions.

Keywords 2D/3D image data · Registration · Fusion · Feature extraction · Tracking · Classification · Hand-based robot control

1.1 Introduction

Object tracking and classification are the main tasks in different kinds of applications such as safety, surveillance, man–machine interaction, driving assistance system and traffic monitoring. In each of these applications, the aim is to detect and find the position of the desired object at each point in time. While in the safety application, the personnel as the desired objects should be tracked in the hazardous environments to keep them safe from the machinery, in the surveillance application they are tracked to analyze their motion behavior for conformity to a desired norm for social control and security. Man-Machine-Interaction, on the other hand has become an important topic for the robotic community. A powerful intuitive interaction between man and machine requires the robot to detect the presence of the user and interpret his gesture motion. A driving assistance system detects and tracks the obstacles, vehicles and pedestrians in order to avoid any collision in the moving path. The goal of traffic monitoring in an intelligent transportation system is to improve the efficiency and reliability of the transport system to make it safe and convenient

S.E. Ghobadi (✉), O.E. Loepprich, O. Lottner, K. Hartmann, W. Weihs, and O. Loffeld
Center for Sensor Systems (ZESS), University of Siegen, Paul-Bonatz-Str.9-11,
D57068 Siegen, Germany
e-mail:Ghobadi@zess.uni-siegen.de; Loepprich@zess.uni-siegen.de; Lottner@zess.uni-siegen.de; Hartmann@zess.uni-siegen.de; Weihs@zess.uni-siegen.de; Loffeld@zess.uni-siegen.de

S.-I. Ao et al. (eds.), *Advances in Machine Learning and Data Analysis*,
Lecture Notes in Electrical Engineering 48, DOI 10.1007/978-90-481-3177-8_1,

for the people. There are still so many significant applications in our daily life in which object tracking and classification plays an important role. Nowadays, the 3D vision systems based on Time of Flight (TOF) which deliver range information have the main advantage to observe the objects three-dimensionally and therefore they have become very attractive to be used in the aforementioned applications. However, the current TOF sensors have low lateral resolution which makes them inefficient for accurate processing tasks in the real world problems. In this work, we first propose a solution to this problem by introducing our novel monocular 2D/3D camera system and then we will study some aspects of object tracking and classification using 2D/3D image data.

1.2 2D/3D Vision System

Although the current optical TOF sensors [13–16] can provide intensity images in addition to the range data, they suffer from a low lateral resolution. This drawback can be obviated by combining a TOF camera with a conventional one. This combination is a tendency in the recent research works because even with regard to the emerging new generation of TOF sensors with high resolution,[1] an additional 2D sensor still results in a higher resolution and provides additional color information. With regard to the measurement range, however, the problem of parallax does not allow to simply position two cameras next to each other and overlay the generated images.

The multimodal data acquisition device used in this work is a recently developed monocular 2D/3D imaging system, named *MultiCam*. This camera, which is depicted in Fig. 1.1, consists of two imaging sensors: A conventional 10-bit CMOS gray scale sensor with VGA resolution and a Photonic Mixer Device (PMD) with a resolution of 64 × 48 pixels. The PMD is an implementation of an optical Time of Flight (TOF) sensor, able to deliver range data at quite high frame rates.

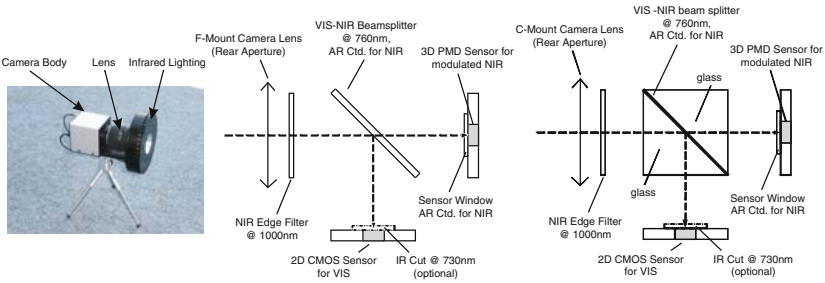

Fig. 1.1 *Left*: 2D/3D vision system (*MultiCam*) developed at ZESS. *Middle*: F-mount optical setup. *Right*: C-mount optical setup

[1] For example PMD-40K (200 × 200 pixels), Swissranger 4000 (176 × 144 pixels) and ZCam-prototype (320 × 480 pixels).

The principles of this sensor will be presented briefly in the next subsection. In addition, a beam splitter (see again Fig. 1.1), a near-infrared lighting system, a FPGA based processing unit as well as an USB 2.0 communication interface represent the remaining main components of this camera. It should be mentioned that the dichroic beam splitter behind the camera lens is used in order to divide the incident light into two spectral ranges: The visible part, which is forwarded to the CMOS chip and the near-infrared part to the TOF sensor [6]. Thus, the *MultiCam* is actually a multi-spectral device.

In fact, through the use of the *MultiCam*, one is able not just to achieve distance data at high frame rates (100 FPS and above) but also high resolution color images provided by the CMOS sensor. The novelty hereby is that a monocular setup is used which avoids parallax effects and makes the camera calibration a lot simpler along with the possibility to synchronize the 2D and 3D images down to several microseconds.

1.2.1 3D-Time of Flight Camera

Basically, the principle of the range measurement in a TOF camera relies upon the time difference Δt that the light needs to travel a distance d as follows

$$\Delta t = \frac{d}{c} \tag{1.1}$$

where c represents the speed of light.

As a lighting source, we use a modulated light signal ($f_{mod} = 20\text{MHz}$), which is generated using a MOSFET based driver and a bank of high speed infrared emitting diodes. The illuminated scene then is observed by an intelligent pixel array (the PMD chip), where each pixel samples the amount of modulated light. To determine the distance d, we measure the phase delay $\Delta\varphi$ in each pixel. Recall that $\Delta\varphi = 2\pi \cdot f_{mod} \cdot \Delta t$ which in turn leads us to

$$d = \frac{c \cdot \Delta\varphi}{2\pi \cdot f_{mod}}. \tag{1.2}$$

Since the maximal phase difference of $\Delta\varphi$ can be 2π, the unambiguous distance interval for range measurement at a modulation frequency of 20 MHz is equal to 15 m. This leads to the maximal (unambiguous) target distance of 7.5 m since the light has to travel the distance twice. In order to be able to use (1.2) for the distance computation in the TOF camera we have to multiply the equation by a factor of 0.5.

To calculate the phase delay $\Delta\varphi$, the autocorrelation function of the electrical an optical signal is analyzed by a phase-shift algorithm. Using four samples A_1, A_2, A_3 and A_4, each shifted by $\pi/2$, the phase delay can be calculated using [1]

$$\Delta\varphi = \arctan\left(\frac{A_1 - A_3}{A_2 - A_4}\right). \tag{1.3}$$

In addition, the strength a of the signal, which in fact can be seen as its quality, along with the gray scale b can be formulated as follows [5]

$$a = \frac{1}{2} \cdot \sqrt{(A_1 - A_3)^2 + (A_2 - A_4)^2}, \tag{1.4}$$

$$b = \frac{1}{4} \cdot \sum_{i=1}^{4} A_i \,. \tag{1.5}$$

The environment lighting conditions in the background should be considered in all optical TOF sensors. There are various techniques dealing with this issue like using optical filters which only pass the band interested in, or applying some algorithm techniques that remove the noise artifacts of ambient light [8]. In our case, the PMD chip used has an in-pixel so-called SBI-circuitry (Suppression of Background Illumination) which increases the sensor dynamics under strong light conditions [1,13].

1.2.2 2D/3D Image Registration and Synchronization

As a prerequisite to profit from the 2D/3D multi-modality, the temporal and spatial relation of the individual sensors' images must be determined.

1.2.2.1 Temporal Synchronization

The detailed disquisition on the temporal synchronization of the individual sensors of the *MultiCam* points out that the camera's internal control unit (FPGA) can synchronize the 2D and the 3D sensor in the temporal domain within the limits of the clock resolution and minimal jitter due to the signal run times in the electronics. The synchronization can either refer to the beginning or to the end of the integration time of a 2D image and a single phase image. While the most common configuration is the acquisition of one 2D image per four phase images such that a new 2D image is available along with a new range image, it is also possible to acquire a new 2D image per phase image if necessary. Figure 1.2 gives an overview of the different possibilities.

If the synchronization of the 2D image relates to the first phase image, the temporal distance between the individual phase images is not equal, as the second phase image is captured only after the end of the 2D sensor's integration time. In contrast to that, synchronizing to the fourth phase image has the advantage of temporally equidistant phase images. In both configurations, it can occur that a change of the scene is represented only by one of both sensors if this change is outside of the actual integration time. With regard to the total time needed for the acquisition of a complete 2D/3D image, these two configurations do not differ from each other. However, the synchronization to the range image rather than to any of the phase

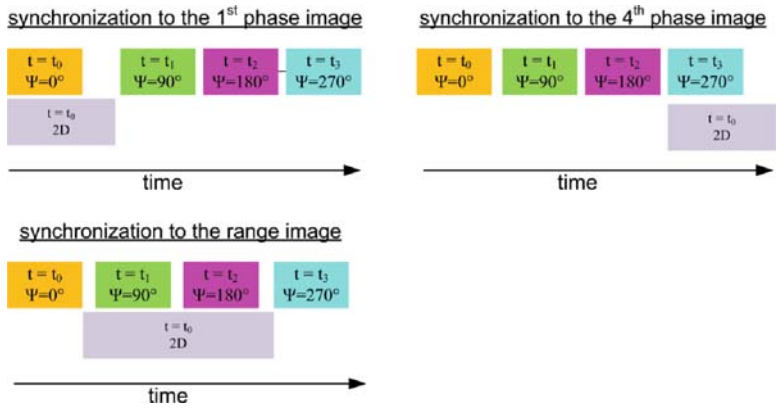

Fig. 1.2 Different possibilities of synchronizing of the 2D image to the 3D data

images is advantageous in that the total acquisition time is kept to a minimum, and in that the temporal equidistance of the phase images is maintained. The discussion on the motion artifacts in [10] gives details on the impacts of the individual configurations.

Binocular setups taking two complete cameras are evidently not as flexible and as precise as the very neatly controlled approach of the *MultiCam*.

1.2.2.2 Spatial Registration

In a 2D/3D vision system, regardless of the kind of setup, a 3D scene is imaged by two two-dimensional matrices of pixels with a degree of overlap which is a-priori unknown but qualitatively assumed to be high without loss of generality. Both sensors operate in a different spectrum (NIR vs. VIS) and have a different modality, i.e., the grey values of the scene represent different physical properties. Due to the operation principle, the sensors operate with different illumination sources meaning that the effects of the illumination must be taken into consideration (corresponding features may be dissimilar due to different lighting conditions). Both sensors have a different resolution with the 2D sensor's resolution being higher. The TOF sensor acquires the distance to an observed point with an accuracy and reproducibility in the range of a centimeter. The relative arrangement of both sensors is not a function of the time but is known in advance with only an insufficient accuracy, meaning that the configuration needs to be calibrated initially.

The aim of the image registration is to establish a spatial transform that maps points from one image to homologous points in a target image as follows

$$[x_1, y_1, z_1]^T = f\left([x_2, y_2, z_2]^T\right). \tag{1.6}$$

The actual transform model depends on the individual configuration. In the following the monocular setup is going to be presented. Considering the special case of the *MultiCam*, the sensors share a common lens, a common extrinsic calibration and the same scene is imaged with the same scale. First, the uncorrected view after sensor alignment is described. This analysis is useful for detecting angle errors which can occur if the angle of 45° (90° respectively) between the beam splitter and the 2D sensor (the PMD sensor respectively) is not exactly adjusted. For this purpose, a test pattern is put in front of the camera and is recorded with both sensors. This test pattern consists of a grid of circles. It is assumed that the reflectivity of this test pattern in the visible spectrum does not differ significantly from the reflectivity in the near-infrared spectrum. In that case, the circles' middle points can be detected reliably with both sensors which results in two sets of matching control points P_{PMD} and P_{2D}. Figure 1.3 shows the average of the displacement between these two sets in units of 2D pixels as a function of the distance between the camera and the pattern for a constant focal length. The average and the standard deviation are computed out of all the circles' middle points in the image. It can be observed that the displacement averages are stable over distance, which means that there is virtually no angle error in the sensor alignment in the observed range. By examining the displacement distribution over the whole image, it can be further concluded that the displacement arrangement does not reveal significant local deviations. Consequently a global rigid transformation model can be used which is independent of the location in the image. Additionally, the uncorrected view shows that the pixel-to-pixel mapping is fixed with a negligible rotational component and which, in particular, is independent from the depth. What remains is a transformation composed only of a two-dimensional translational displacement. Consequently, an iterative closest point algorithm[2] is used to find an optimal solution.

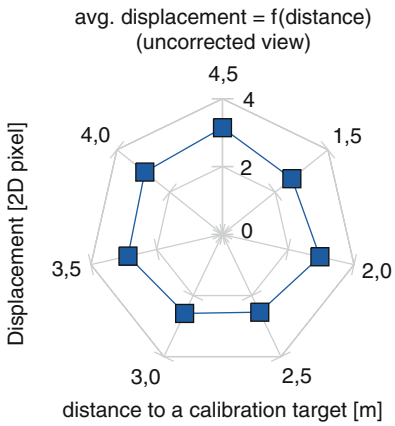

Fig. 1.3 Uncorrected dislocation of PMD and 2D sensor in the *MultiCam*

[2] Levenberg-Marquardt algorithm; the optimization criterion is the sum of squared distances of the individual points.

1.3 Multimodal Data Fusion and Segmentation

The TOF camera delivers three data items for each pixel at each time step: intensity, range and amplitude of the received modulated light. The intensity image of the TOF camera comparable to the intensity images in CCD or CMOS cameras relies on the environment lighting conditions, whereas the range image and the amplitude of the received modulated light are mutually dependent.

None of these individual data can be used solely to make a robust segmentation under variant lighting conditions. Fusing these data provides a new feature information which is used to improve the performance of the segmentation technique.

In this paper we have used the basic technique for the fusing of the range and intensity data which has already been used in other fields like SAR imaging. We observed that the range data in our TOF sensor is dependent on the reflection factor of the object surface (how much light is reflected back from the object). Therefore, there is a correlation between the intensity and range vector sets in a TOF image. These two vector sets are fused to derive a new data set, so-called "phase", which indicates the angle between two intensity and range vector sets. The details of this technique is presented in our previous works [12]. Another type of fusion which has also been used in our work is to weight the value of the range for each pixel using the modulation amplitude which adjusts the range level in the regions where the range data might get wrong.

However, using *MultiCam*, we can acquire low resolution TOF images with their corresponding features derived from fusion; and high resolution 2D Images. For segmentation, same as in [12], first we apply the unsupervised clustering technique to segment the low resolution TOF images. Next, we map the 3D segmented image to 2D image. Due to the monocular setup of *MultiCam*, mapping the 3D range image to the 2D image is a trivial and fast task which consequently makes the segmentation of high resolution 2D image computationally cheap. This kind of segmentation has two main advantages over 2D segmentation. On the one hand 3D range segmentation is more reliable and robust in the natural environment where lighting conditions might change and on the other hand due to the low resolution of 3D image, segmentation is faster. An example of such a segmentation is shown in Fig. 1.4.

1.4 Object Tracking and Classification

One of the approach for object identification in tracking process is to use a classifier directly to distinguish between different detected objects. In fact, if the classification method is fast enough to operate at image acquisition frame rate, it can be used directly for tracking as well. For example, supervised learning techniques such as Support Vector Machines (SVM) and AdaBoost can be directly employed to classify the objects in each frame because they are fast techniques which can work at real time rate for many applications.

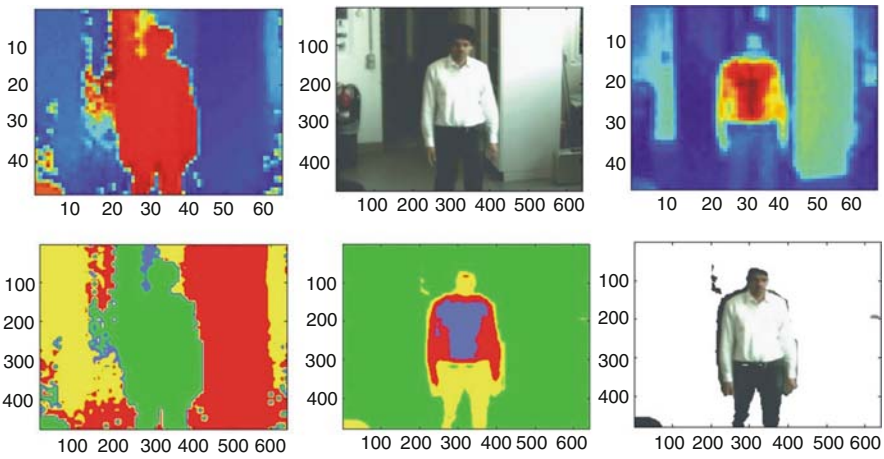

Fig. 1.4 Segmentation in 2D/3D images. *Top Left*: low resolution range image from TOF sensor. *Top Middle*: high resolution 2D image. *Top Right*: modulation amplitude image. *Bottom Left*: 3D rescaled segmented image. *Bottom Middle*: rescaled segmented image using fusion of range and modulation amplitude data. *Bottom Right*: 2D segmented image result from mapping

In this section, we describe tracking with classifier more in detail by applying a supervised classifier based on AdaBoost to 2D/3D videos in order to detect and track the desired object. After segmentation of the image which was described in the previous section, in the next step the Haar-Like features are extracted and used as the input data for the AdaBoost classifier. Haar-like features which have been used successfully in face tracking and classification problems encode some information about the object to be detected. For a much more in depth understanding the reader is referred to [11].

However, there are two main issues in real time object detection based on Haar-Like features and using AdaBoost technique. The first issue is that background noise in the training images degrades detection accuracy significantly, esp. when it is a cluttered background with varying lighting condition which is the case in many real world problems. The second issue is that computation of all sub-windows (search windows) in an image for every scale is too costly if the real time constraints are to be met. The fundamental idea of our algorithm is to address the solution to these problems using fusion of 3D range data with 2D images. In order to extinguish the background issue from object recognition problem, the procedure of object detection is divided into two levels. In the low level we use range data in order to: (i) Define a 3D volume where the object of interest is appearing (Volume of Interest) and eliminate the background to achieve robustness against cluttered backgrounds and (ii) Segment the foreground image into different clusters. In the high level we map the 3D segmented image to its corresponding 2D color image and apply Viola-Jones method [11] (searching with Haar-Like features) to find the desired object in the image. Figure 1.5 shows some examples of this procedure for hand detection and tracking.

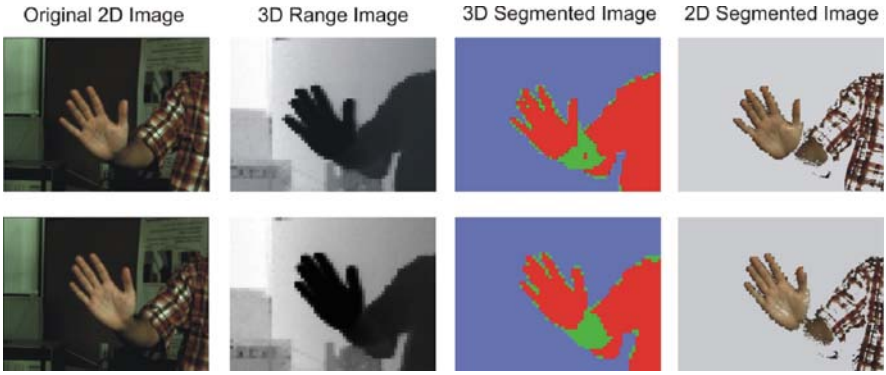

Fig. 1.5 Solution to the background issue in object detection using Viola-Jones method. Using range data the cluttered background is removed and the foreground image is segmented and mapped to 2D image. Viola-Jones technique is applied to 2D segmented image to find the object of interest

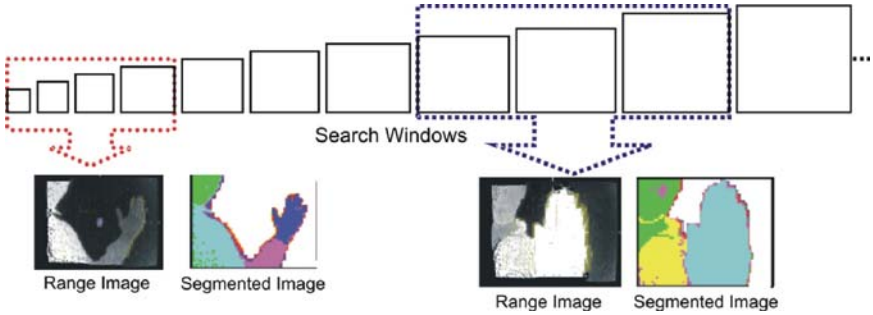

Fig. 1.6 Selection of search windows using range information for hand detection. *Left*: Hand is far from the camera and therefore the image is searched with small search windows. *Right*: Hand is close to the camera and therefore the image is scanned with large search windows to find the hand in the image

The second issue (Computation of all search windows in an image for every scale is too costly.) can be addressed by using the range information directly. After segmentation, the distance of the segmented object from the camera can be easily derived from 3D range image. By having the information about the distance of object from the camera, its size can be roughly estimated and a set of search windows which could fit to the size of the object is selected and therefore there is no need to use all possible size of search windows to find the object. This reduces the computational cost of the Viola-Jones technique to a great extent which is a significant point in real time applications. An example of selecting search windows for hand detection is illustrated in Fig. 1.6.

1.5 Real Time Hand Based Robot Control Using 2D/3D Images

Nowadays, robots are used in the different domains ranging from search and rescue in the dangerous environments to the interactive entertainments. The more the robots are employed in our daily life, the more a natural communication with the robot is required. Current communication devices, like keyboard, mouse, joystick and electronic pen are not intuitive and natural enough. On the other hand, hand gesture, as a natural interface means, has been attracting so much attention for interactive communication with robots in the recent years [2–4, 9]. In this context, vision based hand detection and tracking techniques are used to provide an efficient real time interface with the robot. However, the problem of visual hand recognition and tracking is quite challenging. Many early approaches used position markers or colored gloves to make the problem of hand recognition easier, but due to their inconvenience, they can not be considered as a natural interface for the robot control. Thanks to the latest advances in the computer vision field, the recent vision based approaches do not need any extra hardware except a camera. These techniques can be categorized as: model based and appearance based methods [7]. While model based techniques can recognize the hand motion and its shape exactly, they are computationally expensive and therefore they are infeasible for a real time control application. The appearance based techniques on the other hand are faster but they still deal with some issues such as: complex nature of the hand with more than 20 DOF, cluttered and variant background, variation in lighting conditions and real time computational demand. In this section we present the results of our work in a real time hand based tracking system as an innovative natural commanding system for a Human Robot Interaction (HRI).

1.5.1 Set-Up

Set-up mainly consists of three parts: (1) A six axis, harmonic driven robot from *Kuka* of type KR 3 with attached magnetic grabber. The robot itself has been mounted onto an aluminium rack along with the second system component. (2) A dedicated robot control unit, responsible for robot operation and communication by running proprietary software from *Kuka*© company. (3) The main PC responsible for data acquisition from 2D/3D imaging system (*MultiCam*) and running the algorithms. Communication between the robot control unit and the application PC is done by exchanging XML-wrapped messages via TCP/IP. The network architecture follows a strict client server model, with the control unit as the client connecting to the main PC, running a server thread, during startup.

1.5.2 Control Application

In order to make the communication system more convenient for the user, all the necessary commands to control the robot, such as moving the robot in 6 directions ($x^+, x^-, y^+, y^-, z^+, z^-$) or (de)activating the grabber (palm-to-fist or vice versa) are done by using a self developed GUI based application illustrated in Fig. 1.7. As a first step, we track the user's hand movement in a predefined volume covered by the *MultiCam*, followed by mapping its real world position into a virtual space which is represented by a cuboid of defined size and correlates with the *MultiCam's* view frustum. Hand movement is visualized by placing a 3D hand-model in the according location within the cuboid. Depending on the hand's distance from the cuboid's center, a velocity vector is generated, along with some other state information, and sent to the robot's control unit which is in charge of sending the appropriate information to the robot itself. By placing the virtual hand in the cuboid's center, the system can be put in a susceptible mode for special commands. For that matter, a rudimentary gesture classification algorithm has been implemented which is able to distinguish between a fist and a palm. We use defined fist to palm transition sequences (e.g., a palm-fist-palm transition) in order to perform a robot reset, put the system in predefined modes and to (de)activate the magnetic grabber which in turn enables the whole system to handle ferric objects (Fig. 1.8).

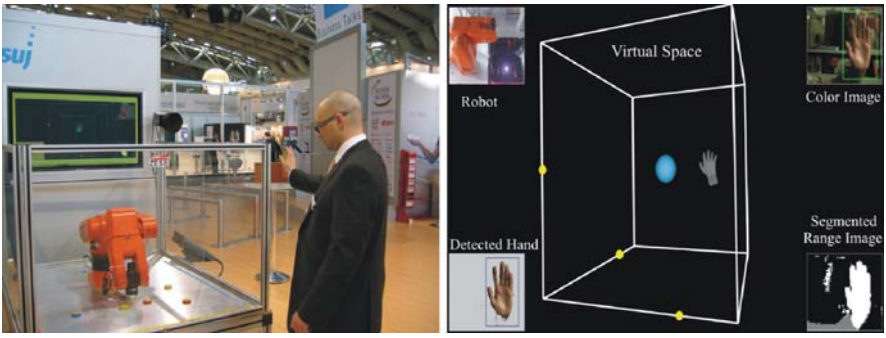

Fig. 1.7 *Left*: Hand based robot control using *MultiCam*, Hannover Fair 2008, *Right*: Graphical User Interface (GUI)

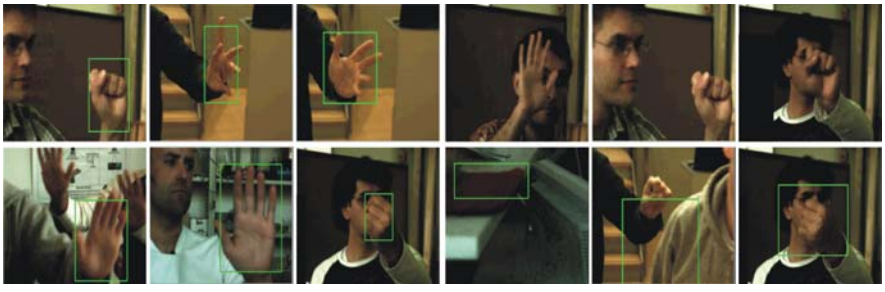

Fig. 1.8 Some results, *left*: example of correctly detected images (True Positive), *right*: example of wrongly detected images (*First row*: missed hand, *Second row*: misclassified)

Table 1.1 Confusion table for hand detection system

	Hand	Non-hand
Hand	2,633	87
Non-hand	224	2,630
Sum	2,857	2,717

1.5.3 Experimental Results

For the Hannover Fair 2008, a simple task had been defined to be performed by the visitors and to put the system's performance under the test as follows: Commanding the robot to move in six directions using moving the hand with any kind of posture in the corresponding directions, picking up a metal object with the magnet grabber using palm to fist gesture, moving the object using the motion of the hand and finally dropping it in the defined areas with palm to fist gesture. It turned out that the system handling has been quite intuitive, since different people have been able to operate the robot instantly. In terms of reliability the whole system worked flawlessly during the complete time exposed at the fair. For training of the classifier we took 1037 positive hand images from seven people, and 1,269 negative images from non-hand objects in our lab environment. Using OpenCV we trained our classifier with 20 stages and window size of 32×32. Although the classifier was trained under the lab conditions, it worked quite well under the extreme lighting conditions at the fair.

In order to analyze the performance of the system, we recorded the results of hand detection from our GUI in the video format while different users were commanding the robot. Likewise, we moved the camera and took the videos from the environment. These videos are labeled as "Positive" and "Negative" data. While positive stands for the hand, the negative represents the non-hand objects. The data were acquired using a PC with dual core 2.4 GHz CPU. The exposure time for 3D sensor was set at 2ms while for 2D sensor it was about 10 ms. Under these conditions, we had about 15 detected images (including all algorithms computational time) per second. The confusion matrix derived from these videos with 2857 hand images and 2717 non-hand images is shown in Table 1.1. As it can be calculated from this table, the system has a Hit Rate of 0.921, False Positive Rate of 0.032 and the recognition accuracy of 94.4%.

1.6 Conclusion

In this work we study some aspects of object detection and tracking using 2D/3D Images. These images are provided by a monocular 2D/3D vision system, so-called *MultiCam*. The principle of this camera system as well as the registration and fusion of 2D/3D data are discussed. This work is concluded with some results of a real time hand based robot control application which was demonstrated at Hannover fair in Germany in 2008.

Acknowledgments This work has been funded by German Research Foundation (DFG) under contract number LO 455/10-2 which is gratefully appreciated.

References

1. Moeller, T. Kraft, H. and Frey, J.: Robust 3D Measurement with PMD Sensors, PMD Technologies GmbH, www.pmdtec.com
2. Wang, C.C. and Wang, K.C.: Hand Posture Recognition Using Adaboost with SIFT for Human Robot Interaction, International Conference on Advanced Robotics, 2007.
3. Cerlinca, T.L., Pentiuc, S.G. and Cerlinca, M.C.: Hand Posture Recognition for Human-Robot Interaction. Proceedings of the 2007 workshop on Multimodal interfaces in semantic interaction,2007.
4. Malima, A., Ozgur, E. and Cetin, M.: A Fast Algorithm for Vision-based Hand Gesture Recognition for Robot Control. IEEE Conference on Signal Processing and Communications Applications, 2006.
5. Ghobadi, S.E., Hartmann, K., Weihs, W., Netramai, C., Loffeld, O. and Roth, H.: Detection and Classification of Moving Objects-Stereo or Time-of-Flight Images, IEEE conference on Computational Intelligence and Security, 2006, China.
6. Lottner, O., Hartmann, K., Weihs, W. and Loffeld, O.: Image Registration and Calibration aspects for a new 2D/3D camera, EOS Conference on Frontiers in Electronic Imaging, June 2007, Munich, Germany.
7. Fang, Y., Wang, K., Cheng, J. and Lu, H.: A Real-Time Hand Gesture Recognition Method. 2007 IEEE International Conference on Multimedia and Expo.
8. Gokturk, S.B., Yalcin, H. and Bamji, C.: A Time of Flight Depth Sensor, System Description, Issues and Solutions, on IEEE workshop on Real-Time 3D Sensors and Their Use in conjunction with IEEE Conference on Computer Vision and Pattern Recognition, CVPR, Washington, USA 2004.
9. Rogalla, O., Ehrenmann, M., Zoellner, R., Becher, R. and Dillmann, R.: Using Gesture and Speech Control for Commanding a Robot Assistant. 11th IEEE International Workshop on Robot and Human Interactive Communication, 2002.
10. Lottner, O., Sluiter, A., Hartmann, K. and Weihs, W.: Movement Artefacts in Range Images of Time-of-Flight Cameras, EOS DOI: 10.1109/ISSCS.2007.4292665, 2007 Romania.
11. Viola, P. and Jones, M.: Rapid Object Detection using a Boosted Cascade of Simple Features. Conference on Computer vision and Pattern Recognition, 2001.
12. Ghobadi, S.E., Loepprich, O., Hartmann, K. and Loffeld, O.: Hand Segmentation Using 2D/3D Images, IVCNZ 2007 Conference, Hamilton, New Zealand, 5–7. December, 2007.
13. PMD-Technologie; www.pmdtec.com
14. Swissranger; C.C.S. d'Electronique SA, http://www.mesa-imaging.ch
15. Canesta, Inc., http://www.canesta.com/
16. 3DV Systems, ZCam; http://www.3dvsystems.com/

Chapter 2
Robot Competence Development by Constructive Learning

Q. Meng, M.H. Lee, and C.J. Hinde

Abstract This paper presents a constructive learning approach for developing sensor-motor mapping in autonomous systems. The system's adaptation to environment changes is discussed and three methods are proposed to deal with long term and short term changes. The proposed constructive learning allows autonomous systems to develop network topology and adjust network parameters. The approach is supported by findings from psychology and neuroscience especially during infants cognitive development at early stages. A growing radial basis function network is introduced as a computational substrate for sensory-motor mapping learning. Experiments are conducted on a robot eye/hand coordination testbed and results show the incremental development of sensory-motor mapping and its adaptation to changes such as in tool-use.

Keywords Developmental robotics · Biologically inspired systems · Constructive learning · Adaptation

2.1 Introduction

In many situations such as home services for elderly and disabled people, artificial autonomous systems (e.g., robots) need to work for various tasks in an unstructured environment, system designers cannot anticipate every situation and program the system to cope with them. This is different from the traditional industrial robots which mostly work in structured environments and are programmed each time for a specific task. Autonomy, self-learning and organizing, and adapting toenvironment changes are crucial for these artificial systems to successfully

Q. Meng (✉) and C.J. Hinde
Department of Computer Science, Loughborough University, LE11 3TU, UK
e-mail: q.meng@lboro.ac.uk

M.H. Lee
Department of Computer Science, University of Wales, Aberystwyth, SY23 3DB, UK

S.-I. Ao et al. (eds.), *Advances in Machine Learning and Data Analysis*,
Lecture Notes in Electrical Engineering 48, DOI 10.1007/978-90-481-3177-8_2,

fulfil various challenging tasks. Traditional controllers for intelligent systems are designed by hand, and they do not have such flexibility and adaptivity. General cognitivist approach for cognition is based on symbolic information processing and representation, and does not need to be embodied and physically interact with the environment. Most cognitivist-based artificial cognitive systems rely on the experience from human designers.

Human beings [1] and animals face similar problems during their development of sensor-motor coordination, however we can tackle these problems without too much effort. During human cognitive development, especially at the early stages, each individual undergoes changes both physically and mentally through interaction with environments. These cognitive developments are usually staged, exhibited as behavioural changes and supported by neuron growth and shrinking in the brain. Two kinds of developments in the brain support the sensory-motor coordination: quantitative adjustments and qualitative growth [19]. Quantitative adjustments refer to the adjustments of the synapse connection weights in the network and qualitative growth refers to the changes of the topology of the network. Inspired by developmental psychology especially Piaget's sensory-motor development theory of infants [12], developmental robotics focuses on mechanisms, algorithms and architectures for robots to incrementally and automatically build their skills through interaction with their environment [21]. The key features of developmental robotics share similar mechanisms with human cognitive development which include learning through sensory-motor interaction; scaffolding by constraints; staged, incremental and self-organizing learning; intrinsic motivation driven exploration and active learning; neural plasticity, task transfer and adaptation. In this paper, we examine robot sensory-motor coordination development process at early stages through a constructive learning algorithm. Constructive learning which is inspired by psychological constructivism, allows both quantitative adjustments and qualitative network growth to support the developmental learning process. Most static neural networks need to predefine the network structure and learning can only affect the connection weights, and they are not consistent with developmental psychology. Constructive learning is supported by recent neuroscience findings of synaptogenesis and neurogenesis occurring under pressures to learn [16, 20]. In this paper, a self-growing radial basis function network (RBF) is introduced as the computational substrate, and a constructive learning algorithm is utilized to build the sensory-motor coordination development. We investigate the plasticity of the network in terms of self-growing in network topology (growing and shrinking) and adjustments of the parameters of each neuron: neuron position, the size of receptive field of each neuron, and connection weights. The networks adaptation to systems changes is further investigated and demonstrated by eye/hand coordination test scenario in tool-use.

2.2 Sensory-Motor Mapping Development Via Constructive Learning

In order to support the development of sensor-motor coordination, a self-growing RBF network is introduced due to its biological plausibility. There exists very strong evidence that humans use basis functions to perform sensorimotor transformations [15], Poggio proposed that the brain uses modules as basis components for several of its information processing subsystems and these modules can be realized by generalized RBF networks [13, 14].

There are three layers in the RBF network: input layer, hidden layer and output layer. The hidden layer consists of radial basis function units (neurons), the size of receptive field of each neuron varies and the overlaps between fields are different. Each neuron has its own centre and coverage. The output is the linear combination of the hidden neurons.

A RBF network is expressed as:

$$\mathbf{f}(\mathbf{x}) = \mathbf{a}_0 + \sum_{k=1}^{N} \mathbf{a}_k \phi_k(\mathbf{x}) \tag{2.1}$$

$$\phi_k(\mathbf{x}) = \exp\left(-\frac{1}{\sigma_k^2} \|\mathbf{x} - \boldsymbol{\mu}_k\|^2\right) \tag{2.2}$$

where $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \cdots, f_{N_o}(\mathbf{x}))^T$ is the vector of system outputs, N_o is the number of outputs and $\mathbf{X}$ is the system input. $\mathbf{a}_k$ is the weight vector from the hidden unit $\phi_k(\mathbf{x})$ to the output, N is the number of radial basis function units, and $\boldsymbol{\mu}_k$ and σ_k are the kth hidden unit's center and width, respectively.

2.2.1 Why Constructive Learning?

According to Shultz [19, 20], in addition to that constructive learning is supported by biological and psychological findings, there are several advantages of constructive learning over static learning: first, constructive-network algorithms learn fast (in polynomial time) compared with static learning (exponential time), and static learning maybe never solve some problems as the designer of a static network must first find a suitable network topology. Second, constructive learning may find optimal solutions to the bias/variance tradeoff by reducing bias via incrementally adding hidden units to expand the network and the hypothesis space, and by reducing variance via adjusting connection weights to approach the correct hypothesis. Third, static learning cannot learn a particular hypothesis if it has not been correctly represented, a network may be too weak to learn or too powerful to generalize. Constructive learning avoids this problem because its network growth enables it to represent a hypothesis that could not be represented previously with limited network power.

2.2.2 Topological Development of the Sensory-Motor Mapping Network

During the development of sensory-motor mapping network, two mechanisms exist: topological changes of the mapping network and network parameter adjustments. The qualitative growth of the sensory-motor mapping network depends on the novelty of the sensory-motor information which the system obtained during its interaction with the environment in development, the growth is incremental and self-organizing. The sensory-motor mapping network starts with no hidden units, and with each development step, i.e., after the system observes the consequence of an action, the network grows or shrinks when necessary or adjusts the network parameters accordingly. The network growth criteria are based on the novelty of the observations, which are: whether the current network prediction error for the current learning observation is bigger than a threshold, and whether the node to be added is far enough from the existing nodes in the network: $\|\mathbf{e}(t)\| = \|\mathbf{y}(t) - \mathbf{f}(\mathbf{x}(t))\| > e_1$, $\|\mathbf{x}(t) - \boldsymbol{\mu}_r(t)\| > e_3$. In order to ensure smooth growth of the network the prediction error is checked within a sliding window: $\sqrt{\sum_{j=t-(m-1)}^{t} \frac{\|\mathbf{e}(j)\|^2}{m}} > e_2$, where, $(\mathbf{x}(t), \mathbf{y}(t))$ is the learning data at tth step, and $\boldsymbol{\mu}_r(t)$ is the centre vector of the nearest node to the current input $\mathbf{x}(t)$. m is the length of the observation window. If the above three conditions are met, then a new node is inserted into the network with the following parameters: $\mathbf{a}_{N+1} = \mathbf{e}(t)$, $\boldsymbol{\mu}_{N+1} = \mathbf{x}(t)$, $\sigma_{N+1} = k\,\|\mathbf{x}(t) - \boldsymbol{\mu}_r(t)\|$, where, k is the overlap factor between hidden units.

The above network growth strategy does not include any network pruning, which means the network size will become large, some of the hidden nodes may not contribute much to the outputs and the network may become overfit. In order to overcome this problem, we use a pruning strategy as in [8], over a period of learning steps, to remove those hidden units with insignificant contribution to the network outputs.

Let o_{nj} be the jth output component of the nth hidden neuron, $o_{nj} = a_{nj} \exp(-\frac{\|\mathbf{x}(t)-\boldsymbol{\mu}_n\|^2}{\sigma_n^2})$, $r_{nj} = \frac{o_{nj}}{max(o_{1j}, o_{2j}, \cdots, o_{Nj})}$.

If $r_{nj} < \delta$ for M consecutive learning steps, then the nth node is removed. δ is a threshold.

2.2.3 Parameter Adjustments of the Sensory-Motor Mapping Network

There are two types of parameters in the network, the first type of parameter is the connection weights; the second is parameters of each neuron in the network: the position and the size of receptive field of each neuron. A simplified node-decoupled EKF (ND-EKF) algorithm was proposed to update the parameters of each node independently in order to speed up the process. The parameters of the