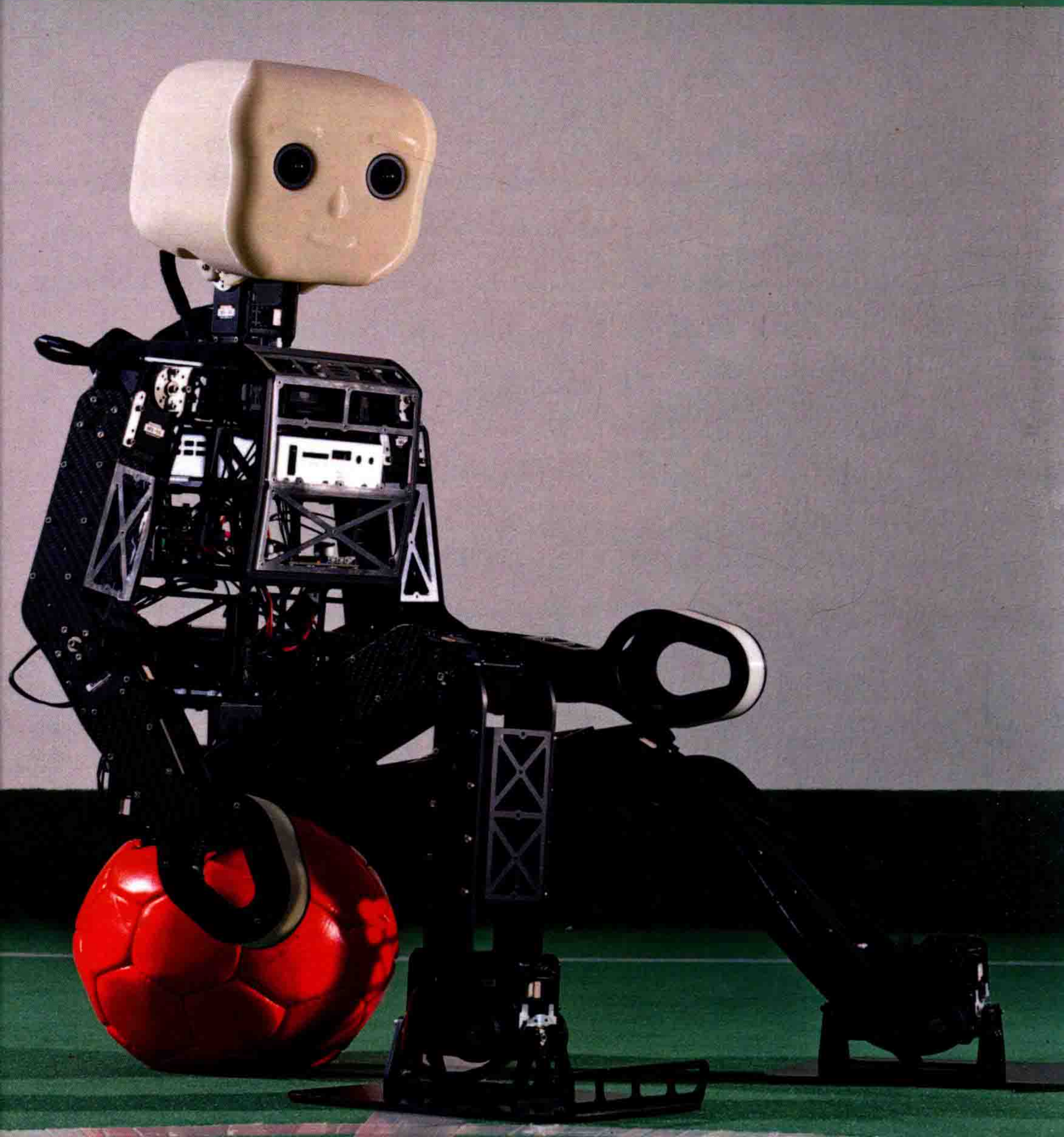
# Computer and Robotic Vision

Contributors

**Hans Utz, Ulrich Kaufmann et al.**

Edited and Compiled by Koros Press Editorial Board



# Computer and Robotic Vision

Contributors

**Hans Utz, Ulrich Kaufmann et al.**

KOROS PRESS

**KOROS PRESS LIMITED**

# Computer and Robotic Vision

Contributors: Hans Utz, Ulrich Kaufmann et al.

**Published by Koros Press Limited**

**www.korospress.com**

United Kingdom



**Computer and Robotic Vision**

ISBN: 978-1-78163-933-7

British Library Cataloguing in Publication Data
A CIP record for this book is available from the British Library

Printed in the United Kingdom

# Computer and Robotic Vision

# List of Contributors

**Hans Utz**
University of Ulm, Neuroinformatics, Ulm, Germany

**Ulrich Kaufmann**
University of Ulm, Neuroinformatics, Ulm, Germany

**Gerd Mayer**
University of Ulm, Neuroinformatics, Ulm, Germany

**Gerhard K. Kraetzschmar**
Fraunhofer AIS, Sankt Augustin, Germany

**Mohammed Z. Al-Faiz**
College of Information Engineering, Al-Nahrain University, Baghdad, Iraq

**Ahmed F. Shanta**
Department of Computer Engineering, Al-Nahrain University, Baghdad, Iraq

**Zuoliang Cao**
Tianjin University of Technology P.R. China

**Xianqiu Meng**
Tianjin University of Technology P.R. China

**Shiyu Liu**
Tianjin University of Technology P.R. China

**Fernando Perez-Peña**
Computer Architecture and Technology Area, Universidad de Cádiz, School of Engineering, Calle Chile, 1, Cadiz 11002, Spain;

**Arturo Morgado-Estevez**
Computer Architecture and Technology Area, Universidad de Cádiz, School of Engineering, Calle Chile, 1, Cadiz 11002, Spain;

**Alejandro Linares-Barranco**
Robotic and Technology of Computers Lab (RTC), Universidad de Sevilla, ETSI Informática, Avd. Reina Mercedes s/n, Sevilla 41012, Spain

**Angel Jimenez-Fernandez**
Robotic and Technology of Computers Lab (RTC), Universidad de Sevilla, ETSI Informática, Avd. Reina Mercedes s/n, Sevilla 41012, Spain

**Francisco Gomez-Rodriguez**
Robotic and Technology of Computers Lab (RTC), Universidad de Sevilla, ETSI Informática, Avd. Reina Mercedes s/n, Sevilla 41012, Spain;

**Gabriel Jimenez-Moreno**
Robotic and Technology of Computers Lab (RTC), Universidad de Sevilla, ETSI Informática, Avd. Reina Mercedes s/n, Sevilla 41012, Spain;

**Juan Lopez-Coronado**
Automation and System Engineering Department, Polytechnic University of Cartagena, Campus Muralla del Mar, Cartagena, 30202, Spain

**Wei-Cheng Lin**
Instrument Technology Research Center, National Applied Research Laboratories Taiwan

**Chao-Chang A. Chen**
National Taiwan University of Science and Technology Taiwan

**Kuo-Cheng Huang**
Instrument Technology Research Center, National Applied Research Laboratories Taiwan

**Yi-Shin Wang**
National Taiwan University of Science and Technology Taiwan

**Niko Herakovic**
University of Ljubljana, Slovenia

**Alexander B. Konovalov**
Russian Federal Nuclear Center - Zababakhin Institute of Applied Physics, Snezhinsk

**Vitaly V. Vlasov**
Russian Federal Nuclear Center - Zababakhin Institute of Applied Physics, Snezhinsk

**Dmitry V. Mogilenskikh**
Russian Federal Nuclear Center – Zababakhin Institute of Applied Physics, Snezhinsk

**Olga V. Kravtsenyuk**
Institute for Laser Physics of Vavilov State Optical Institute Corporation, St.Petersburg Russia

**Vladimir V. Lyubimov**
Institute for Laser Physics of Vavilov State Optical Institute Corporation, St.Petersburg Russia

**Giuseppe Tortora**
The BioRobotics Institute, Scuola Superiore Sant'Anna, viale R. Piaggio 34, 56025, Pontedera (Pisa) 56025, Italy

**Tommaso Ranzani**
The BioRobotics Institute, Scuola Superiore Sant'Anna, viale R. Piaggio 34, 56025, Pontedera (Pisa) 56025, Italy

**Iris De Falco**
The BioRobotics Institute, Scuola Superiore Sant'Anna, viale R. Piaggio 34, 56025, Pontedera (Pisa) 56025, Italy

**Paolo Dario**
The BioRobotics Institute, Scuola Superiore Sant'Anna, viale R. Piaggio 34, 56025, Pontedera (Pisa) 56025, Italy

**Arianna Menciassi**
The BioRobotics Institute, Scuola Superiore Sant'Anna, viale R. Piaggio 34, 56025, Pontedera (Pisa) 56025, Italy

**Shibin Yin**
State Key Laboratory of Precision Measuring Technology and Instruments, Tianjin University, Tianjin 300072, China

**Yongjie Ren**
State Key Laboratory of Precision Measuring Technology and Instruments, Tianjin University, Tianjin 300072, China

**Jigui Zhu**
State Key Laboratory of Precision Measuring Technology and Instruments, Tianjin University, Tianjin 300072, China

**Shourui Yang**
State Key Laboratory of Precision Measuring Technology and Instruments, Tianjin University, Tianjin 300072, China

**Shenghua Ye**
State Key Laboratory of Precision Measuring Technology and Instruments, Tianjin University, Tianjin 300072, China

**Xungao Zhong**
Department of Automation, Xiamen University, South Siming Road, Xiamen 361005, China

**Xunyu Zhong**
Department of Automation, Xiamen University, South Siming Road, Xiamen 361005, China

**Xiafu Peng**
Department of Automation, Xiamen University, South Siming Road, Xiamen 361005, China

**Mohamed Abdellatif**
Ain Shams University, Faculty of Engineering Egypt

**Kunihito Kato**
Gifu University Japan

**Masayuki Shamoto**
Gifu University Japan

**Kazuhiko Yamamoto**
Gifu University Japan

# Preface

Computer vision is a field that includes methods for acquiring, processing, analyzing, and understanding images and, in general, high-dimensional data from the real world in order to produce numerical or symbolic information, e.g., in the forms of decisions. Robot vision is used for part identification and navigation. Vision applications generally deal with finding a part and orienting it for robotic handling or inspection before an application is performed. The text *Computer and Robotic Vision* presents the basic methodology of computer and machine vision, covering the essential elements of the theory while emphasizing algorithmic and practical design constraints. First chapter introduces the video image processing (VIP) framework, a software framework for multithreaded control flow modeling in robot vision. In second chapter, we describe a humanoid robotic arms controlled by tracking the human skeleton movement in real-time using Kinect upper limbbody tracking. Dynamic omnidirectional vision localization using a beacon tracker based on particle filter has been focused in third chapter. A complete spike-based architecture from a dynamic vision sensor (retina) to a stereo head robotic platform has been presented in fourth chapter. Fifth chapter deals with design and fabrication of soft zoom lens applied in robot vision. An overview of research work related to the field of automated vision systems for assembly and quality control processes has been described in sixth chapter. Methods for post processing in single-step diffuse optical tomography have been proposed in seventh chapter. Eighth chapter illustrates the design and testing of a new robotic tool for retraction tasks under vision assistance for orifice transluminal endoscopic surgery (NOTES). In ninth chapter, a novel robot self-calibration approach has been proposed for a robotic visual inspection system. In tenth chapter, a global-state-space visual servoing scheme has been discussed for uncalibrated model-independent robotic manipulation. Eleventh chapter proposes a new method to integrate the behavior decisions by using potential field theory with fuzzy logic variables. An implementation of humanoid vision has been focused in twelfth chapter.

# Contents

*List of Contributors* ........ *vii*

*Preface* ........ *xi*

**Chapter 1** **VIP - A Framework-Based Approach to Robot Vision** ........ **1**

Hans Utz, Ulrich Kaufmann, Gerd Mayer & Gerhard K. Kraetzschmar

**Chapter 2** **Kinect-Based Humanoid Robotic Manipulator for Human Upper Limbs Movements Tracking** ........ **15**

Mohammed Z. Al-Faiz, Ahmed F. Shanta

**Chapter 3** **Dynamic Omnidirectional Vision Localization Using a Beacon Tracker Based on Particle Filter** ........ **29**

Zuoliang Cao, Xianqiu Meng and Shiyu Liu

**Chapter 4** **Neuro-Inspired Spike-Based Motion: From Dynamic Vision Sensor to Robot Motor Open-Loop Control through Spike-VITE** ........ **55**

Fernando Perez-Peña, Arturo Morgado-Estevez, Alejandro Linares-Barranco, Angel Jimenez-Fernandez, Francisco Gomez-Rodriguez, Gabriel Jimenez-Moreno and Juan Lopez-Coronado

**Chapter 5** **Design and Fabrication of Soft Zoom Lens |Applied in Robot Vision** ........ **97**

Wei-Cheng Lin, Chao-Chang A. Chen, Kuo-Cheng Huang and Yi-Shin Wang

**Chapter 6** **Robot Vision in Industrial Assembly and Quality Control Processes** ........ **119**

Niko Herakovic

**Chapter 7** **Methods for Postprocessing in Single-Step Diffuse Optical Tomography** ........ **171**

Alexander B. Konovalov, Vitaly V. Vlasov, Dmitry V. Mogilenskikh, Olga V. Kravtsenyuk and Vladimir V. Lyubimov

Chapter 8 **A Miniature Robot for Retraction Tasks under Vision Assistance in Minimally Invasive Surgery** ........ 203
Giuseppe Tortora, Tommaso Ranzani , Iris De Falco , Paolo Dario and Arianna Menciassi

Chapter 9 **A Vision-Based Self-Calibration Method for Robotic Visual Inspection Systems** ........ 221
Shibin Yin , Yongjie Ren, Jigui Zhu , Shourui Yang and Shenghua Ye

Chapter 10 **Robust Kalman Filtering Cooperated Elman Neural Network Learning for Vision-Sensing-Based Robotic Manipulation with Global Stability** ........ 251
Xungao Zhong , Xunyu Zhong and Xiafu Peng

Chapter 11 **Behavior Fusion for Visually-Guided Service Robots** ........ 283
Mohamed Abdellatif

Chapter 12 **An Implementation of Humanoid Vision -Analysis of Eye Movement and Implementation to Robot** ........ 301
Kunihito Kato, Masayuki Shamoto and Kazuhiko Yamamoto

**Citations** ........ 315
**Index** ........ 319

# Chapter 1

# VIP - A FRAMEWORK-BASED APPROACH TO ROBOT VISION

Hans Utz[1]; Ulrich Kaufmann[1] ; Gerd Mayer[1]
& Gerhard K. Kraetzschmar[2]

[1]University of Ulm, Neuroinformatics, Ulm, Germany,
[2]Fraunhofer AIS, Sankt Augustin, Germany,

## ABSTRACT

For robot perception, video cameras are very valuable sensors, but the computer vision methods applied to extract information from camera images are usually computationally expensive. Integrating computer vision methods into a robot control architecture requires to balance exploitation of camera images with the need to preserve reactivity and robustness. We claim that better software support is needed in order to facilitate and simplify the application of computer vision and image processing methods on autonomous mobile robots. In particular, such support must address a simplified specification of image processing architectures, control and synchronization issues of image processing steps, and the integration of the image processing machinery into the overall robot control architecture. This paper introduces the video image processing (VIP) framework, a software framework for multithreaded control flow modeling in robot vision.

# INTRODUCTION

Perception of the environment is a key component for autonomous mobile robots, which relies on various kinds of sensors providing data that allow to extract relevant information. Video cameras are nowadays low-cost components yet a very rich source of information, but the computer vision methods necessary to extract the desired information are often computationally expensive. Integrating computer vision methods into a robot control architecture requires great care and substantial effort by the programmer, especially in highly dynamic environments which require an adequate level of reactivity. System designers must carefully balance their desire to apply computationally costly image processing methods for maximally exploiting the information contained in camera images with the necessity to maintain timely reaction to critical sensor events and provide reasonable degree of robustness. Better software support is needed in order to facilitate and simplify the application of computer vision and image processing methods on autonomous mobile robots. In particular, a framework for robot vision must address a simplified specification of image processing architectures, control and synchronization issues of image processing steps, and the integration of the image processing architecture into the overall robot control architecture. This paper introduces the video image processing (VIP) framework, a software framework for multithreaded control flow modeling in robot vision and discusses its support for the development of robot vision applications. It is particularly well-suited for robot vision applications in highly dynamic environments, such as robot soccer playing like in the RoboCup middle-size robot league (Utz, H. & Kaufmann, U. & Mayer, G., 2005). The remainder of this paper is organized as follows: The next section discusses challenges for robot vision and related work. The VIP framework is introduced in Section 3 and a discussion of the provided support for the development process of robot vision applications is presented in Section 4. The application of the framework is illustrated by a short example in Section 5, and the paper concludes with application examples, conclusions, and possible future work.

# IMAGE PROCESSING ON MOBILE ROBOTS

Vision systems for autonomous mobile robots must unify the requirements and demands of two very challenging disciplines: i) computer vision and image processing, and ii) robotics and embedded system. While the state of the art in computer vision algorithms is quite advanced, many computer vision methods are intrinsically computationally expensive. Even an efficient implementation of such methods cannot fix this problem. Therefore, the resource demands of computer vision methods are in conflict with the requirements posed by robotics and embedded systems, which demand very short execution cycles for the control loops which read out and process sensor data, interpret and fuse them, and determine appropriate actions for the actuators. Particularly the real-time requirements of robotics seem to rule out most sophisticated computer vision methods, which is one of the reasons why some computer vision experts get discouraged to work on robot vision. As a consequence, robot vision software systems are often inflexible and hard to maintain, because they tend to contain hard-coded quick hacks, which for efficiency reasons try to exploit micro optimizations like performing multiple operations simultaneously, or because they are heavily model-based or incorporate application-specific heuristics. In order to mediate between the partially contradictory requirements of advanced vision processing in a real-time constraint environment, proper conceptual support for vision processing architectures is required. Such conceptual support should encapsulate the vision application within its application domain. For a better understanding of the different requirements that need to be supported, we briefly review some characteristics of the two problem domains.

## Computer Vision and Image Processing

The basic concept of computer vision is the application of operators to image data, such as logical and arithmetical operations (conjunctions, multiplications), color conversions, morphological functions (erosion, dilation), filtering functions (Gaussian filters, convolutions), or linear transforms (Fourier or wavelet transforms). Often operations transform more than one input image into a new output image. For example, the Canny edge detector (Canny, J. F., 1986) needs two images which are obtained by convolving a horizontal and a vertical

Sobel operator respectively. Other operators, such as optical flow require as input a sequence of images obtained at different instances of time (Horn, B. K. P. & Schunck, B. G., 1980). In principle, an image operation can be viewed as a mapping of one or more input images into a new one. More sophisticated operations cover more general input/output mappings, i.e. the result of an image processing operation does not have to be another image, but may be any other kind of data structure, such as a color histogram, a similarity measure between two images, or any other statistic measure on the image. In this case, the definition of a filter is extended to a mapping of one or multiple input images into a new image, or one or multiple classification values associated with the image. Sequences of such image operators reveal features within the image that can be used to identify regions of interest (ROIs). Some filters do not work on the whole image, but only on parts of it. ROIs are used either to speed up the processing loop, or to make sure that the result of successive operations is not influenced or noisified by image areas which are already known to be irrelevant (e.g. in object classification). Various kinds of feedback loops, such as integration over time, can speed up processing and improve classification results (Mayer, G. & Melchert, J. & Utz, H. & Kraetzschmar, G. & Palm, G., 2005). Because regions of interest can change between individual processing steps, they are associated to images just like the above mentioned classification values.

## Robot Vision

Performing operations such as described in the previous section on the video image stream supplied by one or more cameras on an autonomous mobile robot imposes further constraints on the processing model.

Timeliness Constraints: Robots are situated in a physical world. For tasks like obstacle detection and object tracking the image processing operations must be calculated many times per second and preferably at full frame rate, which is typically 30Hz. The system designer needs to repeatedly assess the performance of the vision system and to ensure its efficiency. Whenever possible, image processing operations should be executed in parallel in order to fully exploit the available resources, such as dual-CPU boards and hyperthreading and multicore processor technologies. More complex image operations, which need not be applied at full frame-rate, should be executed asynchronously in order to ensure that the performance of other image evaluations is not jeopardized. Adequate processing models are required to support such designs.

Fixed Frame Rate Image Streams: New images usually arrive at a fixed frame rate. As the value of the obtained information rapidly decreases in a dynamic environment, a sensor-triggered evaluation model is required.

Parameterization: Most image operators need to be parameterized in order to tune the quality of their results. Examples are the width of a Gaussian filter or the number of buckets for an orientation histogram. The calibration and optimization of parameters is an important part of the development process. Also, the configuration of the filter graph has to be altered frequently during development, which can be significantly facilitated by a flexible and configurable development environment for robot vision systems.

Development Model: Robot vision is performed on live image streams, which are recorded by a moving robot platform. This must be addressed in the development model. For many vision applications, development starts on sets of test images and recorded test image streams. If the application domain implies nondeterminism, or if the robot's actions affect the quality of sensor data by inducing effects like motion blur, the vision system needs to be tested extensively in the real-world environment. This requires effective and stringent support for the test-debug cycle, for inspection, and for adaptation of parameters in the running application.

## Related Work

The widely known vision-related architectures and the associated relevant literature can be roughly divided into three categories: subroutine libraries, command languages, and visual programming languages.

Subroutine libraries are most widespread. They concentrate on the efficient implementation of image operators. Therefore, they typically provide a set of functions, each of which implements a different image processing operation. Well-known examples are the SPIDER system (Tamura, H. & Sakane, S. & Tomita, F. & Yokoya, N., 1983) and NAG's IPAL package (Carter, M. K. & Crennell, K. M. & Golton, E. & Maybury, R. & Bartlett, A. & Hammarling, S. & Oldfield, R., 1989), which are written in C or Fortran. More recent libraries include the LTI-Lib (ltilib) or VXL (vxl), which are both open-source, written in C++, and provide a wide range of image operations covering image processing methods, visualization tools, and I/O functions. The commercial Intel Performance Primitives (ipp) are an example for highly (MMX and SSE) optimized processing routines with a C-API. What all these libraries have in common is their lack of adequate support for flow control. Aside of yet another collection of mutex or semaphore helper classes and possibly some kind of thread abstraction, there is no special flow control support available. Command languages for image processing are commonly implemented as scriptable command line tools. In case of the imlib3d package (imlib3d), the image processing operators can be called from the Unix command line. The CVIPtools (Umbaugh, S. E., 1998) are delivered with an extended Tcl command language. Both packages provide programming constructs for conditionals and iteration. While the programmer has complete control over the system in a very flexible way, she also carries full liability over the dynamics of the image processing cycle. Additionally, the scripting-based approach does not make it any easier to meet the required performance constraints of our typical application domains. Visual programming languages currently present probably the most sophisticated solution approach. They allow the user to connect a flowchart of the intended processing pipeline using the mouse. They combine the expressiveness and the flexibility of both libraries and command languages. Often, they provide not only a wide spectrum of image processing functions and statistical tools, but also a complete integrated development environment. Many of the available systems are commercial products, with Visi-Quest (formerly known as Khoros/Cantata) being one of the most advanced ones. According to the information available from their web site, it supports distributed computing capabilities for deploying applications across a heterogeneous network, data transport abstractions (file, mmap, stream, shared memory) for efficient data movement, and some basic utilities for memory allocation and data structure I/O. To the best of our knowledge there is no image processing framework that combines all of the features described above, like processing parts of the filter tree on demand and in a flexible yet powerful way. Such