EXPERIMENTAL ENVIRONMENTS FOR COMPUTER VISION AND IMAGE PROCESSING

SERIES IN MACHINE PERCEPTION AND ARTIFICIAL INTELLIGENCE

Editors: **H. Bunke** (Univ. Bern, Switzerland)
P. S. P. Wang (Northeastern Univ., USA)

- Vol. 1: Pattern Recognition and Image Analysis: Selected Papers from the IVth Spanish Symposium
(Eds. *N. Pérez de la Blanca, A. Sanfeliu and E. Vidal*)
- Vol. 2: Theory and Applications of Image Analysis: Selected Papers from the 7th Scandinavian Conference on Image Analysis
(Eds. *P. Johansen and S. Olsen*)
- Vol. 3: Neural Networks in Vision and Pattern Recognition
(Eds. *J. Skrzypek and W. Karplus*)
- Vol. 4: Parallel Image Processing
(Eds. *A. Saoudi, M. Nivat and P. S. P. Wang*)
- Vol. 5: Advances in Structural and Syntactic Pattern Recognition: Proceedings of the International Workshop on Structural and Syntactic Pattern Recognition
(Ed. *H. Bunke*)
- Vol. 6: Active Robot Vision: Camera Heads, Model Based Navigation and Reactive Control
(Eds. *H. I. Christensen, K. W. Bowyer and H. Bunke*)

Forthcoming

- Vol. 7: Advances in Pattern Recognition Systems Using Neural Network Technologies
(Eds. *I. Guyon and P. S. P. Wang*)
- Vol. 8: Thinning Methodologies for Pattern Recognition
(Eds. *C. Y. Suen and P. S. P. Wang*)
- Vol. 9: State of the Art in Digital Mammographic Image Analysis
(Eds. *K. W. Bowyer and S. Astley*)
- Vol. 10: Generic Object Recognition Using Form and Function
(Eds. *K. W. Bowyer and L. Stark*)

EXPERIMENTAL ENVIRONMENTS FOR COMPUTER VISION AND IMAGE PROCESSING

Edited by

H I Christensen

*Laboratory of Image Analysis
Institute of Electronic Systems
Aalborg University, Denmark*

J L Crowley

*Institute National Polytechnique de Grenoble
France*

World Scientific

Singapore • New Jersey • London • Hong Kong

Published by

World Scientific Publishing Co. Pte. Ltd.

P O Box 128, Farrer Road, Singapore 9128

USA office: Suite 1B, 1060 Main Street, River Edge, NJ 07661

UK office: 73 Lynton Mead, Totteridge, London N20 8DH

**EXPERIMENTAL ENVIRONMENTS FOR COMPUTER VISION
& IMAGE PROCESSING**

Copyright © 1994 by World Scientific Publishing Co. Pte. Ltd.

All rights reserved. This book, or parts thereof, may not be reproduced in any form or by any means, electronic or mechanical, including photocopying, recording or any information storage and retrieval system now known or to be invented, without written permission from the Publisher.

For photocopying of material in this volume, please pay a copying fee through the Copyright Clearance Center, Inc., 27 Congress Street, Salem, MA 01970, USA.

ISBN 981-02-1510-X

Printed in Singapore by JBW Printers & Binders Pte. Ltd.

Contents

Editorial	vii
<i>H. I. Christensen & J. L. Crowley</i>	
The Khoros Application Development Environment	i
<i>J. Rasure & S. Kubica</i>	
Prototyping, Visualization and Simulation Using the Application Visualization System	33
<i>C. J. Westelius, J. Wiklund & C. F. Westin</i>	
XITE — X-Based Image Processing Tools and Environment	63
<i>T. Lønnestad & O. Milvang</i>	
LABOIMAGE: An Experimental Window-Based Environment for Teaching and Research in Image Analysis	89
<i>A. Jacot-Descombes & T. Pun</i>	
ScilImage: A Multi-Layered Environment for Use and Development of Image Processing Software	107
<i>R. van Balen, D. Koelma, T. K. tenKate, B. Mosterd & A. W. M. Smeulders</i>	
Vision as Process: Integration and Control of a Real Time Active Vision System	127
<i>J. L. Crowley & H. Christensen</i>	
A Modular Control Architecture For Real-Time Synchronous and Asynchronous Systems	157
<i>P. L. Butler & J. P. Jones</i>	
Integrating Visual Procedures for Mobile Perception	183
<i>B. Draper, A. R. Hanson & E. M. Riseman</i>	
The Vision Programmers Workbench (VIPWOB)	195
<i>N. O. S. Kirkeby & H. I. Christensen</i>	
Module Network Tool	225
<i>L. Olsson</i>	
Object-Oriented Communication in Image Processing Systems	253
<i>H. Winroth</i>	
Vista: A Software Environment for Computer Vision Research	281
<i>A. R. Pope & D. G. Lowe</i>	

This page is intentionally left blank

Editorial - Experimental Environments for Computer Vision & Image Processing

Henrik I Christensen† & James L. Crowley*

† Laboratory of Image Analysis
Aalborg University
Fr. Bajers Vej 7
DK-9220 Aalborg East
Denmark

* L.I.F.I.A.
Institute National Polytechnique de Grenoble
46 Ave Felix Viallet
F- 38036 Grenoble
France

Introduction

Rapid advances in computing technology have made it possible to build computer vision systems that are useful for real world applications. This has driven the discipline advance from image understanding to the design and construction of working vision systems. The concern with working systems has brought about a need for experimental environments both testing individual vision techniques, and for prototyping architectures of vision systems. Such environments can greatly reduce the time and effort required to test vision techniques and to prototype vision systems. In this special issue we have the opportunity to present a diverse cross section of such experimental vision environments.

The construction of a complex system is typically a team effort, where such teams may be a group of people at a single institution or a group of researchers collaborating across institutions. In either case, it is necessary that the team work within the framework of a common process decomposition and interface standards, so that individual efforts can fit together. Experimental environments provide such a framework, by specifying, in advance, processes, data structures and communication protocols. Adopting such an environment assures the group that independently developed components can fit together.

Ideally an experimental environment should provide the following facilities to an end user:

Define standards for interfacing to the environment: The system should provide standards for programming, i.e., how software must be produced in order to be integrated into the system. In addition it should also define homogeneous standards for interaction with the system ranging from data formats to user interfaces. Such definitions are in particular critical for systems which are developed by multiple researchers residing at different geographical positions.

Independence of Operating System: A wide set of different platforms are used both in industry and at research institutions, ranging from MS-DOS based PC's, over UNIX work-stations to super computers like the CM-5™. It is consequently desirable to be able to use the same method / environment at all of these.

Enable recycling of methods: It is often possible to structure the processing so that it may be broken down into a set of 'simple' modules that potentially may be used on other applications. I.e., methods

such as edge detection, segmentation, graph matching, visualisation and rendering can often be used in new applications, provided facilities for wrapping of such methods are available.

Support real-time processing: Computer Vision and Image Processing have close relations to applications areas such as robotics. Robotics requires real time processing. In computer vision, the increased interest in active vision implies also that real-time response must be available to accommodate control of the sensory system. It is thus desirable to have facilities that allow real-time processing, though in principle only part of the system have to run on hard real-time (control - interaction critical processes).

Provide an abstract interface to hardware devices: Hardware devices such as frame stores and graphics systems are usually equipped with very different facilities for acquisition of images and presentation of results to end-users. To enable construction of portable software, it is desirable to have software that provides an abstraction. I.e., in principle a simple command like `grab()` is sufficient to provide access to the frame grabber, but to enable tuning of the look-up table, selection of video channels, video formats etc. there is typically an abundance of commands available. A standard interface will, however, simplify training and use of such facilities.

Support debugging of software which uses complex data-structures and multi-processing: Most debugging facilities available today allow browsing of simple data structures like integers, bytes and array structures but due to a limited ability to analyse the used data structures it is usually cumbersome to analyse data provided within a computer vision system. In addition there are typically very limited facilities available for debugging of systems which include multiple concurrent processes. It is, however, of interest to have such facilities in particular when the systems of interest increases in size.

Use of non-homogeneous hardware platforms: It is seldom convenient or efficient to use the same computer architecture for pixel level manipulation, 2-D description, 3-D geometry and symbolic computing. In particular, the image/signal level processing may often be implemented efficiently in fine-grained parallel architectures, while intermediate processing may best be carried out on a limited number of medium size processing elements such as Transputers™ and the symbolic computing is better carried out on a single computer which has facilities for efficient handling of symbolic entities. To accommodate such variety of computer architectures, it is necessary to have an environment that allows interfacing to all the architectures. In particular, it is desirable that the distribution onto such a non-homogeneous architecture is transparent to end-users and that the distribution information may be made available to developers on request.

Methods for carrying out systematic experiments including techniques for simple recording of set-up: To fully characterise algorithms and investigate issues such as robustness, domain of application, efficiency etc. it is desirable to have simple techniques for carrying out and recording results from sequences of tests. Preferably the environment should convenient interface to standard packages for data and pattern analysis.

Interactive facilities for processing and analysis of data at any level in a system: An essential facility during testing and debugging is the ability to monitor and examine both data and the interaction of system components system. It is thus desirable to have standard facilities for visualisation and analysis of run-time data, and if possible facilities for visualisation of the system architecture and the interaction between different system components.

Availability of a large set of standard methods: To support rapid prototyping it is desirable to have access to a wide set of standard techniques as soon as the environment is installed as this will simplify introduction of the system and allow for fast adaptation. In addition experiments should include comparative studies and re-implementation of competing techniques is a time consuming task which typically is of little merit.

It is not presently possible to obtain an experimental environment that accommodates all of the requirements that have been stated above. However, over the last few years, computer vision researchers have developed different environments to provide partial support for these issues.

In this volume we have tried to compile a wide variety of different experimental environments. The objectives have been two fold:

- 1) To provide an overview of state of the art with respect to experimental environments; and
- 2) To provide an in-depth description of a wide set of environments to enable prospective users to select a tool appropriate for their area of application.

The description includes tools that are applicable for both teaching and research within computer vision and image processing.

Scanning the volume

The papers in this volume may be grouped into three categories:

- 1) General purpose experimental environments which includes a wide set of methods;
- 2) Application specific environments; in particular systems related to robotics; and
- 3) Basic architectures and facilities for construction of computer vision systems.

General Purpose Systems

In the category of general purpose systems the most widely used system is probably the Khoros system, which is described in chapter 1. That system has been developed at University of New Mexico. The environments which has been available for more than 5 years is a rather complete environment for doing experiments in image processing and analysis. The environment includes a large range of methods (in excess of 200 methods are readily available both in software libraries and as stand-alone modules). The environment includes also a visual interface that enables simple inter-connection of modules to complete systems. In this environment. In addition, it is possible to define simple control structures for

doing experiments and repetitive processing. To facilitate implementation of new techniques a wide set of tools for design of software, user interfaces are also available. The system is available in public domain, though its development is governed by the Khoros Consortium (a joint venture between research institutions and commercial companies).

In chapter 2 general purpose environment "Advanced Visualisation System" (AVS) available through the company AVS Inc. is presented. Initially the AVS system was available from the company Stardent as an advanced interface for its line of super-computers. The system is specifically designed for simple construction of applications in terms of an interconnected set of modules. The system was initially designed for processing that lead to high dimensional data structures that subsequently could be visualised. Thus this system contains very impressive tools for visualisation and animation of data. Over the last few years, the AVS system has been ported to a wide range of UNIX work-stations, and its user community has grown larger and more diverse, in particular in the area of image processing and analysis. This has lead to formation of an AVS application center that ports applications or modules to the AVS environment. For example, a majority of the Khoros modules have now been ported. All such modules are available in public domain (presently in excess of 4000 modules are available through not all of them are in the areas of computer vision and image processing). The group at Linköping University has extensive experience with the AVS environment in particular for image processing and visualisation of robotics scenario. In this chapter they describe a range of different application of the AVS environment within these two areas. The chapter gives a good overview of the possibilities available in general visualisation environments.

An alternative to the AVS environment is the Explorer™ product available through Silicon Graphics Inc. (SGI). This system has a high degree of similarity to AVS, though it has not yet been developed to the same degree of sophistication. The Explorer system is delivered free of charge whenever an SGI work-station is acquired, and thus has a rapidly growing user community.

In Norway, a long term effort has been directed at providing a standard image processing and analysis environment to be used at many of the national research institutions. Initially the package was termed "blab", but it has recently been renamed to "XITE". This environment provides an interactive system that allows experimentation with image level processing, as described in chapter 3. The modules which have been developed and maintained by Oslo University, include contributions from a wide range of Nordic institutions. The environments may run on a diverse set of platforms and it includes an extensive set of techniques. These include good facilities for analysis of texture and several dedicated modules for medical image processing. A pipeline-style architecture with sequential processing has been adopted for this package. This tools is also available in public domain.

Another interactive tool which has been designed for use in teaching and research in computer vision is the LaboImage package from University of Geneva, which is described in chapter 4. In this package, a register based approach is adopted. I.e. images or intermediate results are stored in a predefined set of registers. Each of the methods takes input from one or more of the registers and produces output to

another register. The package includes a wide range of standard image processing and analysis tools. A major emphasis in the design of this tool has been its suitability for use in teaching and training situations. It contains a large set of comparative methods that allow users to study these methods. Many of the text book examples are present. The system has also been used for a number of industrial studies. Examples of the use of the system for inspection are provided in the chapter.

In the Netherlands, there has been a long and substantial effort in image analysis. The effort has included work on development of a suitable environment to be used both in research and teaching, as describe in chapter 5. A wide range of work-stations have been used in this effort. Thus this package is designed and implemented to run on a variety of different platforms, ranging from low-end Apple Macintosh computers to high-end UNIX work-stations. The environment supports distributed processing across platforms and includes an interactive & visualisation tool which has a certain similarity to the Khoros system. Given the system is based on a long term effort it includes a large collection of methods for image processing and analysis tools, including support for reconstruction of 3-D information. This tool also provides a standard set of rules for interfacing and use which has motivated its use in a number of multi-partner projects. The paper includes a brief description of how it has been used both in national and international projects.

Application Specific Environments

The European Commission's Basic Research programme has funded a consortium to develop techniques for the design and control of continuously operating vision systems. This project, which is named "Vision as Process", brings together six partners from five different countries. The experimental environment described in chapter 6 was developed as a skeleton system to enable different partners to integrate their individual contributions into an operating prototype and to study issues related to system architectures and to control. The SAVA skeleton system is designed to integrate information over time, and provides interpreted control to permit individual modules to adapt their functionality based on symbolic reasoning.

The full system architecture includes modules for practically any level of abstraction in computed vision from image acquisition and processing to symbolic interpretation of the scene. The system has been designed to be distributed over a network of computers, including the possibility for dedicated hardware for image processing. In this paper both a flexible system architecture and a standard module architecture used at all levels in the system is presented.

At Oak Ridge National Laboratory there has been a long term effort in construction of autonomous robots for operational use in a nuclear power plant. The effort includes use of a wide range of sensors including sonar, laser range sensors and traditional video cameras. To process such a large amount of information while ensuring real-time operation it is necessary to employ substantial and dedicated computational resources. The robot used, named Hermies-III, has consequently been equipped with a set of real-time computer boards, a hypercube computer, a UNIX work-station and an Apple Macintosh. To provide a suitable environment that enables integration across all the platforms while ensuring real-time

operation the MICA system described in chapter 7 was developed. This system has been used substantially with the Hermies robot and an impressive performance in terms of response time and versatility has been demonstrated. It should be noted that despite the fact that it was developed for use in conjunction with robotics research, it is sufficiently general to be used in computer vision research in general.

The researchers at University of Massachusetts, Amhurst has a long standing tradition of building computer vision systems. Initial work was directed towards construction of knowledge based systems, which later has been complemented by use of such systems in the context of robotics. A characteristic of the effort has been use and integration of a broad range of features to facilitate a robust and efficient processing, analysis and reasoning. To enable this it was at an early stage determined that a suitable basis for processing using such a strategy was use of a common symbolic representation method (termed an Intermediate Symbolic Representation (ISR)). This representation has formed a fundamental basis for both the VISIONS systems, and the commercial product KBVision™. The representation has for natural reasons evolved over the years to accommodate increasing demands in terms of representational power, versatility and efficiency. In chapter 8 the latest version of the ISR (ISR-3) is described together with a system based on this representation. The system has been specially designed to accommodate use in autonomous mobile navigation as part of the ARPA AUV project.

Basic Architectures for Computer Vision

A dominant problem in computer vision research is the immense amount of labour used for implementation of methods for experimental research, another problem is integration across different sites. To make such work more efficient a Danish effort was initiated five years ago. The aim was design of a Vision Programmer Work-Bench (VIPWOB). Basic requirements was the ability to build systems for all aspects of computer vision, while allowing use of existing software with a minimum or no change at all, and interfacing and use of special hardware. To propitiate these requirements the simple flexible, and yet powerful software architecture described in chapter 9 was devised. The architecture imposes very few restrictions on the programmer, enables use of existing software and supports very broad categories of system architectures. The system has now been in use for a little more than 3 years and it has been used for a wide range of different applications.

The interaction between different computer vision processes can typically be formulated in term of objects with associated features such as inheritance, methods etc. To fully exploit the functionality of object oriented communication it is desirable to have a tool that allow simple communication of such objects between modules. In addition it is desirable to have standard module architectures that impose a standard processing model, but the architecture should be sufficiently flexible to allow introduction of arbitrary methods into the control structure. To enable simple inter-connection of modules which are based on such a system philosophy the Module Network Tool (MNT) has been designed and implemented at the Royal Institute of Technology in Sweden. The tool is sufficiently general to support arbitrary module

and system configurations and distribution of such across a set of work-stations, as described in chapter 10.

A basic problem in distribution of image processing techniques across a range of work-stations is maintenance of consistency between such modules. If an object oriented approach is adopted for the programming and/or communication it is further desirable to have simple mechanisms for transfer not only of data but also of associated methods and properties. In the framework of UNIX networking a standard for exchange of simple data types has been developed by Sun Microcomputers. This standard and the associated implementation is called XDR (External Data Representation). To accommodate use of a similar technique, but for object oriented communication an External Object Representation (XOR) has been developed at the Royal Institute of Technology in Sweden. Through use of this methods it is simple to construct and interconnect modules which uses Object Oriented Programming. The system has been designed so that it may be used with the traditional C-programming language, with C++ and the LISP package CLOS. The technique, which is described in chapter 11, is in highly interesting as it in principle may be integrated into any of the earlier described experimental environments.

Summary

A wide set of different systems are presented in this volume. In general they provide a good overview of the techniques available to researchers for performing experimental work. It should be noted that one major effort, the ARPA Image Understanding Environment, is not presented here. Unfortunately, it was not possible to obtain contributions from any of the researchers involved in that effort. It is, however, felt that a comprehensive and adequate presentation is provided. We hope the readers will find the volume timely and suitable companion in their experimental work.

We would like to thank the reviewers for their help in bringing together a set of high quality papers in a minimum of time. The editorial effort has been supported by the CEC DG-III Basic Research Office through the project "Vision as Process-II" (EP-7108 VAP-II) and the Danish Technical Research Council through the "MOBS program", this support is gratefully acknowledged.