

AISB '03
Cognition
in Machines
& Animals

**Proceedings of the
AISB '03 Symposium on
Biologically-Inspired Machine Vision,
Theory and Application**

**7th – 11th April 2003
The University of Wales, Aberystwyth**

EPSRC



Prifysgol Cymru
Aberystwyth
The University of Wales



The AISB'03 Convention
COGNITION in MACHINES and ANIMALS

7th - 11th April 2003
University of Wales, Aberystwyth

www.aishb.org.uk

Proceedings of the Symposium on
Biologically-Inspired Machine Vision,
Theory and Application

Published by

**The Society for the Study of Artificial
Intelligence and the Simulation of
Behaviour**

www.aisb.org.uk

ISBN 1 902956 33 1

Printed at the University of Wales, Aberystwyth, SY23 3DB, Wales

The proceedings of the five symposia in the AISB'03 Convention are available from SSAISB. They are entitled as follows:

The Second International Symposium on Imitation in Animals and Artifacts
ISBN 1 902956 30 7

The Third Symposium on Adaptive Agents and Multi-Agent Systems
ISBN 1 902956 31 5

Scientific Methods for the Analysis of Agent-Environment Interaction
ISBN 1 902956 32 3

Biologically-Inspired Machine Vision, Theory and Application
ISBN 1 902956 33 1

Artificial Intelligence and Creativity in Arts and Science
ISBN 1 902956 34 X

The AISB'03 Convention

Cognition in Machines and Animals

The Society for the Study of Artificial Intelligence and the Simulation of Behaviour (SSAISB) is the UK's national society in the field of Artificial Intelligence. It is one of the oldest AI societies in Europe and has a long track record of supporting the UK research community. The society has a tradition of holding annual meetings and a convention format has proved successful, most recently at the Universities of Edinburgh (1999), Birmingham (2000), York (2001), and Imperial College, London (2002). We are delighted to host the 2003 convention at the University of Wales, Aberystwyth.

Recent advances in artificial intelligence, cognitive science, neuroscience and robotics have stimulated new interest in the relationships between biologically-based and computer-based cognitive systems. There is a current upsurge of activity both in biologically-targeted computer modelling and in adapting and experimenting with biological concepts for the design and development of novel computing technologies. For example, the BBSRC and EPSRC recently started a joint programme on Adaptive and Interactive Behaviour of Animals and Computational Systems (AIBACS). These topics are providing much inspiration for new research and innovation and the general theme of the 2003 convention reflects this stimulus and has attracted a range of high quality paper submissions and presentations.

In the early years of the SSAISB there was considerable interdisciplinary excitement about the general phenomenon of cognition, and lively meetings were attended by both computer scientists and psychologists. Perhaps the success of AI technology across a wide range of hard applications in the '80s and '90s caused the research focus to drift away somewhat from that most fascinating and astounding source of inspiration: cognition in the human brain. Consequently, it is very pleasing that we are now in the throes of a swing back towards those original concerns and I hope this convention will encourage further interaction between psychology and computer science, not least through the exciting selection of topics and invited speakers.

The convention is structured around five independently organized symposia, each with their own Programme Chairs and Programme Committee. The proceedings of all symposia are published and several of the best papers have been selected for edited publication in the AISB Journal. The unifying theme of the convention is reflected in a series of plenary sessions which feature invited keynote speakers. The six speakers, all international authorities in their field, are: Professor Peter Hobson, from the Tavistock Clinic and University College London; Professor Rajesh Rao, from the Center for Mind, Brain and Learning, University of Washington; Professor Giulio Sandini, from LIRA-Lab, University of Genova; Professor Sorin Solomon, from Theoretical Physics, Hebrew University of Jerusalem; Professor Yiannis Aloimonos, from the Computer Vision Laboratory, Institute for Advanced Computer Studies, University of Maryland; and Professor Gregor Schöner, from the Institut für Neuroinformatik, Ruhr-Universität Bochum.

We are extremely grateful for the support of the EPSRC, who provided funds for both the invited speakers' travel costs and for a large number of bursaries for postgraduate research students. The bursaries cover most of the cost of attending the convention and are particularly valuable as they enable the next generation of scientists to interact and to discuss issues with some of the key players in their research field. Support and assistance has been given by the Department of Computer Science at UWA, Meinir Thomas and the Conference Office at UWA, the IEE, and the Committee of the SSAISB, especially Geraint Wiggins. Special thanks are

due to all the symposia Chairs for all their time and hard work: Kerstin Dautenhahn, Chrystopher Nehaniv, Dimitar Kazakov, Daniel Kudenko, Eduardo Alonso, Ulrich Nehmzow, Gregor Schöner, Horst Holstein, Fred Labrosse, Pablo Gervas and Simon Colton. Special thanks also to our Technical Programme Assistant, Joanne Walker, and, for typesetting expertise, to Fred Labrosse. Finally, thanks to all the authors for their contributions, which form the essential core of the event. I am looking forward to the programme and I wish everyone a stimulating and productive meeting.

Mark Lee (Convention Host and Programme Chair)

Biologically Inspired Machine Vision, Theory and Application

Computer vision has developed into a mature science over the last 40 years, but current machine vision systems are vastly different from, and in most cases lack the efficiency of, biological vision systems.

Historically, machine vision attempted to base itself on the functionality of biological systems, but this proved to be too complex both algorithmically and technologically. Research progress was therefore found to be more responsive to the purposive machine vision paradigm. Today, however, the improved technology allows more daring and complex algorithms. Biological vision therefore remains a strong metaphor for the design of machines that replicate intelligent behaviour in visually sensed environments. Immensely rewarding applications in human-machine interaction await advances in the multi-disciplinary threads of machine vision, perception and cognition.

Under the general AISB'03 Convention theme of Cognition in Machines and Animals, the aim of this symposium is to provide a forum for presentation and discussion not only of state-of-the-art research, but also for opening directions of research and technology related to biologically-inspired machine vision and visualisation. With this in mind, we have allowed time for a special session entitled "Is biologically inspired computation the future for Computer Vision?", to elicit participants' views through lively debate.

Our thanks go to all who contributed to the symposium, through authorship and discussion. We also express our gratitude to the programme committee, for helping with the symposium planning, preparation and refereeing:

Prof Eduardo Bayro-Corrochano, Instituto Politecnico Nacional, Mexico

Dr Yonghuai Liu, University of Wales, Aberystwyth, UK

Dr Derek J Paddon, University of Bath, UK

Dr Dimitris Rigas, University of Bradford, UK

Prof Marcos Rodrigues, Sheffield Hallam University, UK.

Horst Holstein,

Fred Labrosse,

Symposium Chairs

University of Wales, Aberystwyth

March 2003

Contents

AISB Keynote Paper	1
Action Representations	
<i>Y. Aloimonos and C. Fermüller</i>	3
Contributed Papers	25
Genetic Algorithm for flower growth	
<i>J. Lu, C. Willis and D. Paddon</i>	27
An artificial retina with a self-organised retinal receptive field tessellation	
<i>L. S. Balasuriya and J. P. Siebert</i>	34
Iterative Orientation Tuning of Simple Cells in V1: A Comparative Study of Two Computational Models for Contrast Detection in Images	
<i>M. Kolesnik and A. Barlit</i>	43
Biological Motivated Multi-modal Processing of Visual Primitives	
<i>N. Krüger, M. Lappe and F. Wörgötter</i>	53
Robust lip modeling and extraction using active shapes	
<i>I. Shdaifat and R.-R. Grigat</i>	60
Automatic Identification of Active Regions (Plages) in the Full-Disk Solar Images Using Local Thresholding and Region Growing Techniques	
<i>A. Benkhalil, V. Zharkova, S. Zharkov and S. Ipson</i>	66
An Automated Recognition of Sunspots on the Ca K1 and SOHO/MDI White Light Solar Images	
<i>S. I. Zharkov, V. Zharkova, S. Ipson and A. Benkhalil</i>	74
Object Localisation and Tracking Through Subsymbolic Classification	
<i>H. V. Neto and U. Nehmzow</i>	85
Estimating ego-motion using a panoramic sensor. Comparison between a bio-inspired and a camera-calibrated method	
<i>S. Gourichon, J. A. Meyer, S. H. Ieng, L. Smadja and R. Benosman</i>	91

Perception of Human Periodic Motion in Moving Light Displays <i>B. Li and H. Holstein</i>	102
On Laboratory Gait Analysis via Computer Vision <i>J.-H. Yoo and M. S. Nixon</i>	109
High-cost cortical image processing on low-cost clusters <i>H. du Buf</i>	114

AISB Keynote Paper

Action Representations

Y. Aloimonos and C. Fermüller

Computer Vision Laboratory, Center for Automation Research, Institute for
Advanced Computer Studies, and the Department of Computer Science, University of
Maryland, College Park, MD 20742-3275, USA

yiannis@cfar.umd.edu,

WWW home page: <http://www.cfar.umd.edu/~yiannis/research.html>

Abstract. We argue that actions represent the basic seed of intelligence underlying perception of the environment, and the representations encoding actions should be the starting point upon which further studies of cognition are built. In this paper we make a first effort in characterizing these action representations. In particular, from the study of simple actions related to 3D rigid motion interpretation, we deduce a number of principles for the possible computations responsible for the interpretation of space-time geometry. Using these principles, we then discuss possible avenues on how to proceed in analyzing the representations of more complex human actions.

1 Introduction and Motivation

During the late eighties, with the emergence of active vision, it was realized that vision should not be studied in a vacuum but in conjunction with action. Adopting a purposive viewpoint makes visual computations easier by placing them in the context of larger processes that accomplish tasks. This new framework, known by a variety of names such as active, purposive, animate or behavioral vision, has contributed a wealth of new technical results and has fueled a variety of new application areas. Several groups constructed active vision systems and studied the basics of many visual competences, but for the most part, this study of the perception/action coupling concentrated on problems related to navigation. A flurry of activity produced a veritable cornucopia of algorithmic approaches to 3D motion estimation, 3D shape recovery, tracking, and motion and scene segmentation. As a matter of fact, one can say that this is the most mature area in the field of computer vision. This was, perhaps, not surprising since all these problems are related to the most basic action of all, one that transcends all systems with vision, that of self-motion; although the perception/action cycle surrounding the basic action of self-motion is not yet fully understood (we will discuss later some of the intricacies involved), it is clear that the problem is one of geometry and statistics awaiting the right modeling for its complete resolution. But an intelligent system with perception is not just a system that understands its movement, and the shape and movement of other objects in its field of view. It has a large number of capabilities related to recognition,

reasoning, planning, imagination and learning. Despite the fact that the active vision framework was embraced as the right tool in dealing with the recovery of 3D information from image sequences, that was not the case when it came to higher-level problems such as recognition, planning and reasoning. Lacking a foundation, not much progress has been made in those higher-level problems. For example, with the exception of simple objects in controlled environments, not much progress has been made in the field of object recognition. All new proposals during the past decade have followed some advance in the structure from motion problem.¹ There are, of course, many reasons for this, and some of them will become clear in the course of our exposition.

The goal of this paper is twofold, one methodological and the other technical. Our intent is to put forward the thesis that action represents the basic seed of intelligence, underlying perception of our environment, which is our major interest, recognition and other high-level processes such as communication. If this is the case, action representations are very basic components of an intelligent system. What are they? What could their nature and kind be? How can we build them? What does it mean to understand them? Most important, what techniques should we employ in order to obtain them? Answering these questions is our second goal and we approach it using only computational arguments. Before we begin, we need to discuss other efforts attempting to answer such kinds of questions.

2 Intelligence, Learning and the Stratification of Reality

Slowly but steadily, the paradigm that has dominated the study of cognition and the fields surrounding it for the past century is starting to be challenged. The dominant approach has been to view mental states as having meaning in terms of symbol systems like natural language. The emerging new approach is referred to as embodied intelligence or the sensorimotor theory of cognition which proposes that higher cognition makes use of the same structures as those involved in sensorimotor activity. This view is highly exemplified in recent efforts in AI to achieve intelligence through building robots and having them “develop” through interaction with their environment. Although there exist interesting points in the contemporary sensorimotor theories of cognition, and the projects of AI engineers have merits of their own, there exist two fundamental problems with them that we explain here.

The belief that smaller systems develop through interaction with their environment and become integrated to form new and more advanced systems owes its appeal to its intended mimicking of evolution. It makes, however, an implicit assumption, namely that evolution is evolution in degree and not evolution in kind. By evolution in kind we mean the introduction of new components having

¹ When it became clear that 3D information could be extracted from point and line correspondences, researchers concentrated on pose recovery in the polyhedral world. When view interpolation was achieved for some cases, recognition approaches considering objects as collections of 2D views appeared.

qualitatively different characteristics from the existing components. This leads to a strange kind of one-way relationship, both between the whole and its constituent parts within the system, and between higher systems and their primitive ancestors. One can define this relationship by saying that the whole is its parts, and continues to be so even if, as the result of the introduction of new subsystems (kind), it acquires a number of additional characteristics in the course of its evolution. The subsidiary systems themselves do not gain any higher characteristics and may even lose some in the process of simplification. The one-way relationship consists in the system as a whole possessing all the characteristics of its component parts, but none of these parts possesses the characteristics specific to the whole. Similarly, every system possesses most of the characteristics of its primitive ancestors, but even complete knowledge of a system's characteristics will not make it possible to predict those of its more highly developed descendants [17]. Thus, we can explain a system only if we accept the present structures in its body as our data.²

The second problem, related to the first, is best explained by utilizing the ideas of the philosopher Nicolai Hartmann [15]. Like all processes in life, those of acquiring and storing relevant information take place on many different levels and are interlinked at many points. The world in which we live, according to Hartmann, is built of different strata, each with its own existential categories that distinguish it from other strata. "There are in the hierarchy of existence certain phenomenal realities whose fundamental differences our minds fail to bridge Any true and accurate theory of categories must have as much regard for these gaps as for the existential relationships that bridge them." These relationships, however, only transcend in a unilateral manner the divisions between the four great strata of existence—the inorganic, organic, cognitive and conscious. The principles of existence and laws of nature that govern inorganic matter apply with equal validity to higher strata. But Hartmann insists that the differences between the higher and lower strata are far from restricted to the distinctions between inorganic, organic, cognitive and conscious. "The higher elements that make up the world are stratified in a similar way to the world itself," he wrote. This means that each step leading from a system of a lower order to one of a higher order is the same in nature and complexity as the coming into existence of life itself. This viewpoint about the stratification of existence can guard us against common mistakes in studying cognition. If we neglect the categories and laws that are exclusive properties of the higher strata or even deny their existence, we commit the fallacy of an upward transgression of the limits imposed upon us by the stratification of the real world. It is not possible to explain the laws and processes proper to higher levels in terms of categories derived from lower levels. Another common mistake results from transgressing the boundaries between levels of existence in the opposite direction. This is described by Hartmann

² Irrational Residue: The number of historical causes one would need to know to fully explain why an organism is as it is may not be infinite but it is sufficiently great to make it impossible to trace all causal chains to their end. As Polanyi pointed out [19], a higher animal cannot be reduced to its simpler ancestors.

as follows: The base of the entire world picture is then chosen on the level of conscious experience—the level on which man experiences his own subjective life—and from there the principle is extended downwards to the lower level of reality.

What does this all mean for us who wish to gain an understanding of perceptual mechanisms and, in particular, how action representations make up an intelligent system? There are several lessons we can obtain. It is a mistake, while seeking unitary explanatory principles behind the world, to try to explain lower and more primitive systems on the basis of principles applicable only to higher systems and vice versa. Similarly, the attempts of some psychologists and behavioral scientists to achieve intelligence through learning are at fault. This is true for both primitive systems that are incapable of it and systems of more advanced organisms which are not only incapable of being modified by learning but whose phylogenetic program makes them resistant to all modification.

The only chance at understanding intelligent systems seems to be to study them by accepting their present structures as our data, and work towards understanding the different components. But here is the tricky part. The components we study should be such that they do not require other components in order to be understood, or if they do, all interrelated components should be studied together. Otherwise it appears that there is no hope for understanding. And finally, our only tools should be the ones of the physical sciences, geometry, statistics and computation.

3 Understanding Actions, Objects

Our task is to increase our understanding of how to piece together the components of an intelligent system with vision. Our thesis is that action is fundamental in this regard. It is of course irresistible to attempt to arrive at a unitary world image and try to explain the diversity of the world (and intelligent systems) on the basis of ontological and phenomenal principles of one single kind. According to our previous analysis this would be a mistake after all. But our goals are much more modest. We want to uncover principles underlying the workings of the perceptual system and in particular shed light on the nature of action representations.

Our basic thesis is that the content of the mind is organized in the form of a model of the world. That model contains objects and their relationships, events and the interaction of the system with its environment. In building, maintaining, manipulating and working with that model we should distinguish three items: the world itself, the system (its body) and the system's interaction with the world. What could the best candidate be for building a foundation on the basis of which this complex model can be understood?

Recall that it is trivially true that understanding a thing is relating it to something already understood; if we continue along this chain, this means that there must be something understood in terms of itself. This entity must be

something with which a system is intimately familiar in a non-symbolic way. We conclude that the best candidate for this is the agent's own activity.

There is a number of basic actions that are the basis of acquiring models of the shape and motion of the environment. These actions are related to the system's understanding of its own motion and they are universal, as all systems must have them in some form. The next sections are devoted to the computational mechanisms underlying the coupling of such actions with perception. By taking a look at these actions, on the one hand we can give an example and elaborate on what it means to understand these actions. On the other hand, since the computations involved in these basic actions form the foundation for many other computational processes we gain insight into the nature of space-time representations. That is, it allows us to deduce a number of principles on what the shape and motion models of the environment might be and how they could be computed. The computational principle of feedback loops is emerging as a foundational theme.

Next we are interested in the very large repertoire of more complex actions through which an agent interacts with his environment. Since understanding these actions should serve as a foundation for all other understanding, we should not presuppose understanding of other things. We merely assume that the agent understands its own body. Equipped with this knowledge and the foundation of space-time perception delivered from basic actions, we can provide a characterization of action as: *One understands an action if one is able to imagine performing the action with images that are sufficient for serving as a guide in actual performance.* This characterization provides two things. First, it shows that it is the ability to imagine an action which constitutes understanding it. Imagining it amounts to representing the constitutive components of the action as such, in a form that is capable of guiding the performance of the action. Because the components of the action form the content of the agent's mental state, the agent can be said to understand it. Second, this sort of understanding produces the feeling of confidence that one understands an action, because the understanding consists in a mental state of which the agent can be aware. Moving along this line of thought, understanding anything amounts to knowing the possible actions one might perform in relation to that thing and being aware of that understanding is consciously imagining those actions. Thus, we can say that one has an understanding of a perceived object if one is able to imagine incorporating the object into the performance of an action that is already understood, with an image sequence sufficient for serving as a guide in actual performance.

Thus, the ability to intentionally perform an action, manifested in the ability to imagine performing it, is a way of understanding the action that presupposes no other understanding. Similarly, an object that is intentionally incorporated into the performance of an (understood) action is itself understood in relation to that action. In moving from the understanding of actions to the understanding of objects we have crossed an important boundary. Understanding actions does not require a concept of the objective external world; that is how it can be a

foundation for such a concept. But understanding objects does require such a concept.

It is not our intention to delve further into a philosophical inquiry about understanding objectivity and the mind/world distinction. Our purpose here is to achieve an understanding of action representations. Having a crisp characterization of what it means to understand an action we will be ready in Sect. 4 to investigate constraints on such representations. We will not delve into object recognition issues but the framework is in place to carry this investigation further into higher-level problems.

But remaining truthful to the principles of embodied intelligence, we cannot study in a technical sense actions unless we commit ourselves to a particular body. Of course the same is true for the basic actions and the perception of shape and motion, but in that case the dependence lies mostly in image acquisition (eye design) and computational capacity; in our days of extreme computational power these issues are not that important. But when it comes to higher-level actions and interaction with the world, it is important to choose a body. A failure to do so will turn our study completely philosophical. As such a body, we choose the human body.

4 Basic Actions

The most basic actions are the ones which can be based on the most simple model of 3D motion, and this is the model of instantaneous rigid motion. As living organisms move through the environment their eyes undergo a rigid motion. The visual system has to recognize and interpret this rigid motion from the visual input, that is, the sequence of images obtained by its eyes. This immediately gives rise to two basic actions: first, the capability of estimating one's self-motion or egomotion; second, the detection of objects in the static scene which move themselves, that is, the detection of independently moving objects. Similarly, the motion of various objects is rigid or can be closely approximated by one. Thus a third action arising from this model is the estimation of rigid object motion. This action, however, is much more difficult than the estimation of self-motion. One reason is that objects usually cover a small field of view and thus less data is available for the estimation. Another reason is that objects lie outside the system's body and thus there is less information from other sensors. In contrast, for self-motion estimation biological systems—in addition to vision—also use inertial sensors (such as those in our ears). As a result, most systems only possess the capability to partially estimate rigid object motion. Even rather simple organisms understand the approaching of the enemy, or can estimate how long it will take to intercept an object (the estimation of time to contact). A somewhat more sophisticated but still partial estimation would be a rough estimate of the object's translation or rotation.

The visual input is a sequence of images, a video. Aside from how the images are formed (that is, the properties of the eye or camera), the images depend on the spatiotemporal geometry, that is, the layout of the scene and the relative 3D

motion between observer and scene, and the optics, that is, the light sources and the reflectance properties of the surfaces. By considering only changes of image patterns, an image representation of the movement of the 3D scene points is obtained. This representation, a vector field, ideally is only due to the geometry that is the relative 3D motion between the eye and the scene and the shape of the scene. If the relative 3D motion between the observer and parts of the scene is rigid, then segmentation amounts to the localization of those parts of the flow field corresponding to the same rigid motion and motion estimation amounts to decoding the 3D motion parameters. The shape of the scene determines the remaining components of the vector field and it can be decoded after motion estimation.

The interpretation of motion information, probably because of its well-defined geometric nature, has received a lot of attention in the computer vision literature. A methodology emerged under the hood of which most studies were conducted, which defined rigid motion interpretation as the “structure from motion” problem and it was considered to be carried out with three consecutive computational steps. First the exact movement of image points is estimated, either in the form of the optical flow field if images densely sampled in time are considered, or in the form of correspondences of single image points if images further apart in time are used. The optical flow field or correspondences are approximations of the projections of the 3D motion field which represents the movement of scene points. The second computational step consists in estimating the 3D rigid motion from the optical flow field, and the third step amounts to estimating the 3D structure of the scene usually from multiple flow fields using the estimates of the 3D motion.

With this framework it has been considered that exact reconstruction of space-time geometry is possible—it is just a matter of using the right sophisticated tools—and it can be addressed in a purely bottom-up manner computationally. We’ll argue that this cannot be. Throughout this and the next section we’ll provide computational arguments supporting our view and elaborate on what the computations should be.

First, optical flow or correspondence cannot be estimated accurately using only the image data, that is, before performing estimations about the 3D geometry. The local image data defines only one component of the image velocity. This is the so-called aperture problem illustrated in Fig. 1: Local image data usually only provides one-dimensional spatial structures. From the movement of these linear features between image frames the movement of single image points cannot be determined, but only the flow component perpendicular to the linear feature; this is the so-called normal flow. In order to estimate the two-dimensional image flow, a model of the flow field is needed; then normal measurements in different directions within local neighborhoods can be combined. Here lies the problem: In order to model the flow field, additional information is necessary. It is easy to model the parts of the image which correspond to smooth scene patches by using smoothness assumptions of different kinds, but there are discontinuities in the flow field, and their locations must be known prior to modeling the flow

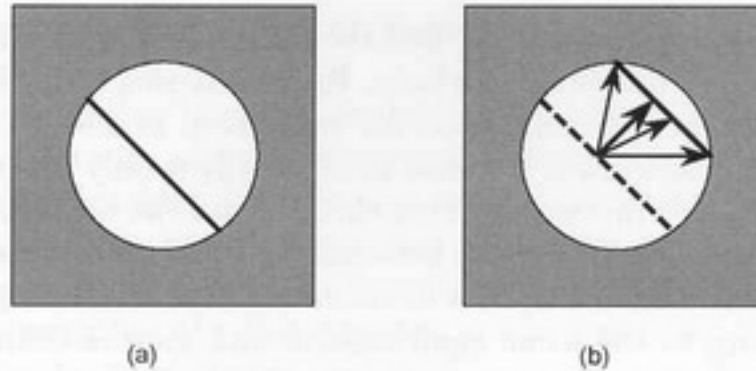


Fig. 1. Aperture problem: (a) Line feature observed through a small aperture at time t . (b) At time $t + \delta t$ the feature has moved to a new position. It is not possible to determine exactly where each point has moved to. From local measurements only the flow component perpendicular to the line feature can be computed

field. The discontinuities in the flow field are due either to independently moving objects in the scene or to scene surfaces at different depths. On the other hand, the discontinuities cannot be detected unless an estimate of the flow field is available. Thus the problems of estimating the flow field and locating the discontinuities are inherently coupled and the situation seems to present itself as a chicken and egg problem. Discontinuities provides an even more severe problem than usually considered. Most of the literature considers the idealized problem of estimating the motion for one rigidly moving observer in a static environment. Thus all the discontinuities they deal with are due to depth. Even if the locations of the discontinuities in the flow field are known, a system would need additional information about the 3D motion and scene to distinguish the motion discontinuities from the pure scene discontinuities. There also has been a lot of work in the literature on motion segmentation, but these studies consider the problem separate from 3D motion estimation: either they attempt a localization of the discontinuities on the basis of flow field information only, or they assume the 3D motion to be known. It is clear that for a real system the two problems of 3D motion estimation and independent motion detection have to be considered together.

Besides the problem due to discontinuities there are also statistical difficulties with the estimation of optical flow. As will be elaborated on later, the estimation of optical flow from noisy normal flow measurements even within neighborhoods of smooth flow is very difficult. To avoid bias and obtain very accurate flow, theoretically one would need detailed models of the flow field, and these in turn can only be obtained from 3D information; specifically, this means the discontinuities, the shape of the scene and the 3D motion.

Since accurate optical flow estimation bottom up seems to be infeasible, the question is, is it possible to perform 3D motion estimation using as input normal flow measurements? In the past we have studied this question, and we have developed two constraints that allow to extract 3D geometry directly from

normal flow. One constraint that relates image motion to 3D motion only, and a second one which also involves the scene.

5 Relating Image Measurements to 3D Geometry

Consider a system moving with rigid motion in a static environment. The motion is described by an instantaneous translational velocity $\mathbf{t}$ and a rotational velocity $\boldsymbol{\omega}$. Each scene point $\mathbf{R} = (X, Y, Z)$ measured with respect to a coordinate system $OXYZ$ fixed to the nodal point of the system's eye where Z is the optical axis has velocity $\dot{\mathbf{R}} = -\mathbf{t} - \boldsymbol{\omega} \times \mathbf{R}$.

We consider the image to be formed on a plane orthogonal to the Z axis at distance f (focal length) from the nodal point. Thus through perspective projection image points $\mathbf{r} = [x, y, f]$ are related to scene points $\mathbf{R}$ by $\mathbf{r} = \frac{\mathbf{R}}{\mathbf{R} \cdot \hat{\mathbf{z}}}$, where $\hat{\mathbf{z}}$ is a unit vector in the direction of the Z axis. The projection of the 3D motion field on the image gives the image motion field

$$\dot{\mathbf{r}} = \frac{1}{(\mathbf{R} \cdot \hat{\mathbf{z}})} (\hat{\mathbf{z}} \times (\mathbf{t} \times \mathbf{r})) + \frac{1}{f} \hat{\mathbf{z}} \times (\mathbf{r} \times (\boldsymbol{\omega} \times \mathbf{r})) = \frac{1}{Z} \mathbf{u}_{\text{tr}}(t) + \mathbf{u}_{\text{rot}}(\boldsymbol{\omega}) \quad (1)$$

where Z is used to denote the scene depth $(\mathbf{R} \cdot \hat{\mathbf{z}})$ and $\mathbf{u}_{\text{tr}}, \mathbf{u}_{\text{rot}}$ are the direction of the component of the flow due to translation and the component of the flow due to rotation, respectively. Due to the coupling of Z and $\mathbf{u}_{\text{tr}}$ in this relationship only the direction of translation (the Focus of Expansion, FOE, or focus of contraction, FOC, depending on whether the observer is approaching or moving away from the scene), scaled depth and the three rotational parameters can be obtained. The direction of the axis of rotation will be denoted as AOR. The normal flow $\mathbf{v}_n$ amounts to the projection of the flow $\dot{\mathbf{r}}$ on the local gradient direction. If $\mathbf{n}$ is a unit vector denoting the orientation of the gradient, the value of the normal flow, v_n amounts to

$$v_n = \dot{\mathbf{r}} \cdot \mathbf{n}$$

The first constraint, the "pattern constraint," uses as its only physical constraint the fact that the scene lies in front of the camera and thus all the depth values have to be positive. This allows the relation of the sign of normal flow measurements to the directions of the translation and the rotation, that is, the FOE and AOR. The basis lies in selecting groups of measurements of normal flow along pre-defined directions (orientation fields) which form patterns in the image plane, and these patterns encode the 3D motion [4], [5], [6], [7].

There are two classes of orientation fields. The first class are called copoint fields. Each copoint field parameterized by $\mathbf{s}$, is defined as the unit vectors perpendicular to a translation flow field with translation $\mathbf{s}$. These unit vectors are in the direction $\mathbf{v}_{\text{cp}}(\mathbf{s}, \mathbf{r}) = \hat{\mathbf{z}} \times \mathbf{u}_{\text{tr}}(\mathbf{s}) = \hat{\mathbf{z}} \times (\hat{\mathbf{z}} \times (\mathbf{s} \times \mathbf{r}))$ and they are perpendicular to the bundle of lines passing through $\mathbf{s}_0$ (the point where $\mathbf{s}$ intersects the image plane) (Fig. 2a).

Now consider a normal flow field due to rigid motion $(\mathbf{t}, \boldsymbol{\omega})$ and consider all the normal flow vectors of this field which are in direction $\mathbf{v}_{\text{cp}}(\mathbf{s}, \mathbf{r})$, that is,

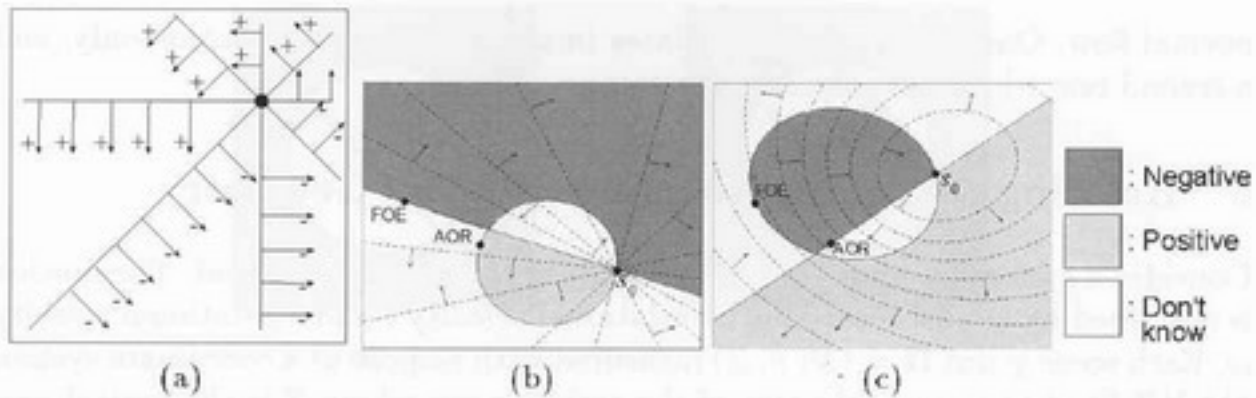


Fig. 2. (a) Copoint field filled with qualitative motion measurements. (b) copoint vectors and copoint pattern. (c) coaxis vectors and patterns

the vectors of value $\frac{\mathbf{v}_{cp}}{\|\mathbf{v}_{cp}\|} \cdot \hat{\mathbf{r}}$. The translational components of these vectors are separated by the conic $(\boldsymbol{\omega} \times \mathbf{r}) \cdot (\mathbf{s} \times \mathbf{r}) = 0$ into an area with negative values and an area with positive values. Similarly, the rotational components of the same vectors are separated by the line $(\mathbf{t} \times \mathbf{s}) \cdot \mathbf{r} = 0$ into areas of positive and negative values. The rigid motion field is the sum of the translational and the rotational flow field; thus the areas above need to be superimposed. Where both the translational and rotational components are positive the combined flow is positive and where both the translational and rotational components are negative the combined flow is negative. In the remaining areas the sign is undetermined. The areas of different sign form a pattern in the image plane as shown in Fig. 2b. For every direction $\mathbf{v}_{cp}$ parameterized by $\mathbf{s}$ there is a different pattern and the intersection of the different patterns provides the 3D motion.

The second class of patterns is defined by the orientation field of coaxis vectors, which are the unit vectors perpendicular to a certain rotational flow field, that is, the coaxis vectors parameterized by $\mathbf{s}$ are in direction $\mathbf{v}_{cA} = \hat{\mathbf{z}} \times (\hat{\mathbf{z}} \times (\mathbf{r} \times (\boldsymbol{\omega} \times \mathbf{r})))$. In this case the translational components are separated by a line and the rotational components are separated by a cone into positive and negative values and give patterns such as that shown in Fig. 2c. Using these pattern constraints the problem of self-motion estimation amounts to recognizing particular patterns in the image plane. The system would compute the normal flow and then for a number of orientation fields find the patterns which are consistent with the sign of the estimated normal flow vectors. The intersection of the possible patterns provides the solution, usually in the form of possible areas for the FOE and AOR, as most often there is not enough data to provide an exact localization of the 3D motion.

The question now is, is it possible at all to have a technique that accurately computes the 3D rigid motion parameters? The answer is no, if the data available is from a limited field of view. This statement is based on an error analysis conducted in [8], [9]. Estimating 3D motion amounts to minimizing some function. If the pattern constraints are used, this function is based on negative depth values and if the classical epipolar constraint requiring optical flow is used, this function is based on the distances of the flow vectors from the epipolar lines. The

correct 3D motion is defined as the minimum in the functions. A topographic analysis of the expected values showed that for all these functions the minimum lies within a valley, and locating the exact position of the minimum within this valley under noisy conditions generally is not possible.

If the 3D motion is known, the depth of the scene can be computed from the flow field. If $\hat{\mathbf{t}}$ and $\hat{\boldsymbol{\omega}}$ are the estimates of the direction of translation and the rotation the estimated depth $\hat{Z}$ is derived from (1) as

$$\hat{Z} = \frac{\mathbf{u}_{tr}(\hat{\mathbf{t}}) \cdot \mathbf{n}}{\hat{\mathbf{r}} \cdot \mathbf{n} - \mathbf{u}_{rot}(\hat{\boldsymbol{\omega}}) \cdot \mathbf{n}} \quad (2)$$

If the estimates $\hat{\mathbf{t}}$ and $\hat{\boldsymbol{\omega}}$ correspond to the correct values of the actual rigid motion $\mathbf{t}$ and $\boldsymbol{\omega}$, then $\hat{Z}$ will give a correct scene depth estimate, but what if there are errors in the estimates? Then the estimated depth will be a distorted version of the actual depth [2], [3], [8], [11].

Substituting for $\hat{\mathbf{r}}$ in (2) the actual motion parameters the estimated depth can be written as

$$\hat{Z} = Z \cdot \frac{\mathbf{u}_{tr}(\hat{\mathbf{t}}) \cdot \mathbf{n}}{\mathbf{u}_{tr}(\mathbf{t}) \cdot \mathbf{n} - Z \mathbf{u}_{rot}(\delta\boldsymbol{\omega}) \cdot \mathbf{n}},$$

with $\delta\boldsymbol{\omega}$ denoting the estimation error in rotation $\delta\boldsymbol{\omega} = \hat{\boldsymbol{\omega}} - \boldsymbol{\omega}$, or to make clear the distortion as $\hat{Z} = Z \cdot D$, where $D = \frac{\mathbf{u}_{tr}(\hat{\mathbf{t}}) \cdot \mathbf{n}}{\mathbf{u}_{tr}(\mathbf{t}) \cdot \mathbf{n} - Z \mathbf{u}_{rot}(\delta\boldsymbol{\omega}) \cdot \mathbf{n}}$ is the multiplicative distortion factor to express how wrong depth estimates result from inaccurate 3D motion values. Note that D also depends on $\mathbf{n}$. For different directions, $\mathbf{n}$, the distortion factor takes on different values, ranging from $-\infty$ to $+\infty$.

This concept of space distortion forms the basis of the second constraint relating normal flow to 3D geometry, the "depth variability constraint." The idea is, instead of formulating smoothness constraints on the 2D optical flow field as is usually done, to relate the smoothness of scene patches and the 3D motion directly to normal flow. To be more specific, if we estimate the depth from the flow field corresponding to a smooth scene patch using the correct 3D rigid motion, we will compute a smooth scene patch. If we use an erroneous 3D motion estimate, then the obtained depth function, assuming that normal flow vectors in different directions are available, will be rugged. Not only do incorrect estimates of motion parameters lead to incorrect depth estimates, but the distortion is such that the worse the motion estimate, the more likely depth estimates are obtained that locally vary more than the correct ones. Thus one has to define functions which express the variability of depth values and search for the 3D motion which minimizes these functions. Locally for the flow values $\hat{\mathbf{r}}_i$ in direction $\mathbf{n}_i$ within a region R we define depth variability measures as

$$\theta_0(\hat{\mathbf{t}}, \hat{\boldsymbol{\omega}}, R) = \sum W_i \left(\hat{\mathbf{r}}_i \cdot \mathbf{n}_i - \mathbf{u}_{rot}(\mathbf{u}) \cdot \mathbf{n}_i - \frac{1}{\hat{Z}} (\mathbf{u}_{tr}(\hat{\mathbf{t}}) \cdot \mathbf{n}) \right)^2$$

where $1/\hat{Z}$ is the depth estimate minimizing θ_0 , which is obtained by modeling the depth within the region using a parametric model, and W_i are weights that can be chosen differently. Solving for the minimum $1/\hat{Z}$ and substituting back

we obtain θ_1 . The sum of all θ_1 in all regions, θ_2 , is the global function which we are interested in minimizing.

To algorithmically utilize these constraints one has to proceed in iterations as follows. Split the image into patches and perform a search in the space of translational directions. For every candidate translation estimate the best rotation using the depth variability measure, then perform a depth segmentation using the image data and the candidate motion parameter, and finally evaluate the depth variability measure taking into account the segmentation. The solution is found as the 3D motion minimizing the global depth variability measure.

In summary of the above discussion a number of conclusions can be drawn regarding the possible computations of space-time geometry.

- It is not possible to reliably estimate the 3D rigid motion very accurately if the field of view is limited. This is true even for the simplified case of one camera moving in a static environment. If there is less data available—as it is when one considers motion of objects covering a small field of view—the estimation is much complicated and all we can expect are bounds on the motion parameters.
- It is also not possible for a real-time system to derive accurate depth estimates. Considering the cue of motion, this follows from the fact that 3D motion cannot be estimated accurately. As a result of this one obtains a distorted depth map, but it also has been shown experimentally for other cues that human space perception is not exactly Euclidean [16].
- Since reconstruction is not possible the information describing the models must be encoded in the form of patterns in the flow field.
- The computations carrying out the space-time geometry interpretation must be implemented as feedback processes. This is true even for the simple, basic actions as will be elaborated on in the next section, and for more complicated actions the interplay between bottom-up and top-down processes becomes extremely important.

6 Feedback

The interpretation of motion fields is a difficult computational problem. From the raw images only, it is not possible to compute a very accurate 2D motion field, and from erroneous motion field measurements it is not possible to perform good 3D motion estimation, and, in the sequel, scene estimation. To estimate good flow we need information about the space-time geometry, and to estimate the geometry well we need at least the flow field to be segmented accurately.

This seemingly chicken and egg situation calls for a simultaneous estimation of image motion and 3D motion and scene geometry, and this could only be implemented through an iterative or feedback process. The whole interpretation process should take the following form: First, the system estimates approximate image velocity by combining normal flow measurements. The representation of these estimates could be in the form of qualitative descriptions of local flow field

patches or bounds on the flow values, but it should allow for finding the more easily detected discontinuities in the flow field. Using the flow computed in this way or maybe even normal flow measurements within segmented areas an estimate of 3D motion is derived and a partial 3D shape model of the scene computed. Subsequently, the computed information about the space-time geometry is fed back to use image measurements from larger regions to perform better flow estimation, more accurate discontinuity localization, and in the sequel the estimates of 3D motion and structure are improved. Naturally the whole interpretation process has to be dynamic. While the system computes structure and motion from one flow field, the geometry changes and new images are taken. Thus the system also has to be able to make predictions and relate the 3D information computed earlier in time to images acquired later.

There are a number of ways these feedback processes could be implemented in a system. From a computational perspective we can ask for an optimal way, and this will depend on the computational power of the system and the accuracy of the estimates needed. There are many efforts in the biological sciences, in fields such as psychophysics, neurophysiology and anatomy, to figure out the structure and processes which exist in human and other primate brains—this structure is generally referred to as the motion pathway; but many of these studies do not consider that the sole purpose of motion processing is to interpret the 4D space-time geometry. We'll next elaborate on two visual illusions that in our opinion demonstrate that the human visual system must perform feedback processes to interpret image motion.

6.1 The Ouchi Illusion

The striking Ouchi illusion shown in Fig. 3 consists of two black and white rectangular checkerboard patterns oriented in orthogonal directions—a background orientation surrounding an inner ring. Small retinal motions, or slight movements of the paper, evince a segmentation of the inset pattern, and motion of the inset relative to the surround. Our explanation lies in the estimation of differently biased flow vector fields in the two patterns which in turn give rise to two different 3D motion estimates. When the system feeds back these motion estimates it performs a segmentation of the image using the flow fields and the static image texture, and because of the difference in 3D motion one pattern appears to move relative to the other.

The bias is easily understood by considering the most simple model of image motion estimation. Local image measurements define at a point one component of the image flow, for example, if the image measurements are the spatial derivatives E_x, E_y and the temporal derivative E_t of the image intensity function, E , at point $\mathbf{r}_i$ the flow $\dot{\mathbf{r}} = (u, v, 0)$ is constrained by the equation

$$E_{x_i}u + E_{y_i}vE_{t_i} = E_{t_i} \quad (3)$$

We consider additive zero mean independent noise in the derivatives of E . Let n be the number of indices i to which (3) applies, which for convenience we write

in matrix form as

$$\mathbf{E}_s \mathbf{u} = \mathbf{E}_t \quad (4)$$

where $\mathbf{E}_s$ is the n by 2 matrix incorporating the spatial derivatives E_{x_i}, E_{y_i} , $\mathbf{E}_t$ is an n by 1 matrix incorporating the E_{t_i} , and $\mathbf{u}$ is the vector (u, v) .

If (4) is solved by standard least squares estimation the solution is

$$\mathbf{u} = (\mathbf{E}_s^T \mathbf{E}_s)^{-1} \mathbf{E}_t^T \mathbf{u}$$

and this solution, if there are errors in the spatial derivatives, is biased. The expected value, $E(\mathbf{u})$, of $\mathbf{u}$ amounts to

$$E(\mathbf{u}) = \mathbf{u}' - n\sigma_s^2 (\mathbf{E}_s'^T \mathbf{E}_s')^{-1} \mathbf{u}',$$

where primes are used here to denote actual values and σ_s^2 is the variance of the noise in the spatial derivatives. This expected value of $\mathbf{u}$ is an underestimate in length and in a direction closer to the majority of gradient directions in the patch.

It has been shown in [13], [14] that it is not only gradient based techniques and least squares estimation that produce bias, but other techniques (frequency-domain, correlation techniques, and different estimation procedures) as well. Correcting for the bias would require knowledge of the noise parameters and these are difficult to obtain. In the case of the Ouchi illusion the 3D motion, which is caused by small eye movements or rapid motion of the paper, changes very quickly so the system does not have enough data available to acquire the statistics of the noise. In more natural situations, however, the system can reduce the bias problem if it performs feedback processes. This way, after some estimates of motion and structure have been obtained, image measurements from larger image regions can be considered to acquire more accurate statistical noise sampling.

6.2 *Enigma* and Variants

The static figure, *Enigma*, painted by Leviant and shown in Fig. 4, consists of radial lines emanating from the center of the image and interrupted by a set of concentric, uniformly colored rings. Upon extended viewing of *Enigma*, most humans perceive illusory movement inside the rings which keeps changing direction. Based on a study of positron emission tomography, researchers in neurophysiology [20], [21] have argued that higher level processes of motion interpretation in the human brain are responsible for the perception of this illusion. To be more specific, these researchers compared the cerebral blood flow during the viewing of *Enigma* and a similar reference image which does not give rise to an illusion, and they found the values comparable in the early motion processing area V1, but found significantly higher values for *Enigma* in V5 (or MT)—a later area in visual processing.

The spatial texture in *Enigma* is closely related to the patterns described in Sect. 5 that form constraints for 3D motion estimation. This lets us hypothesize that the illusion essentially is due to higher level activity. It is due to the

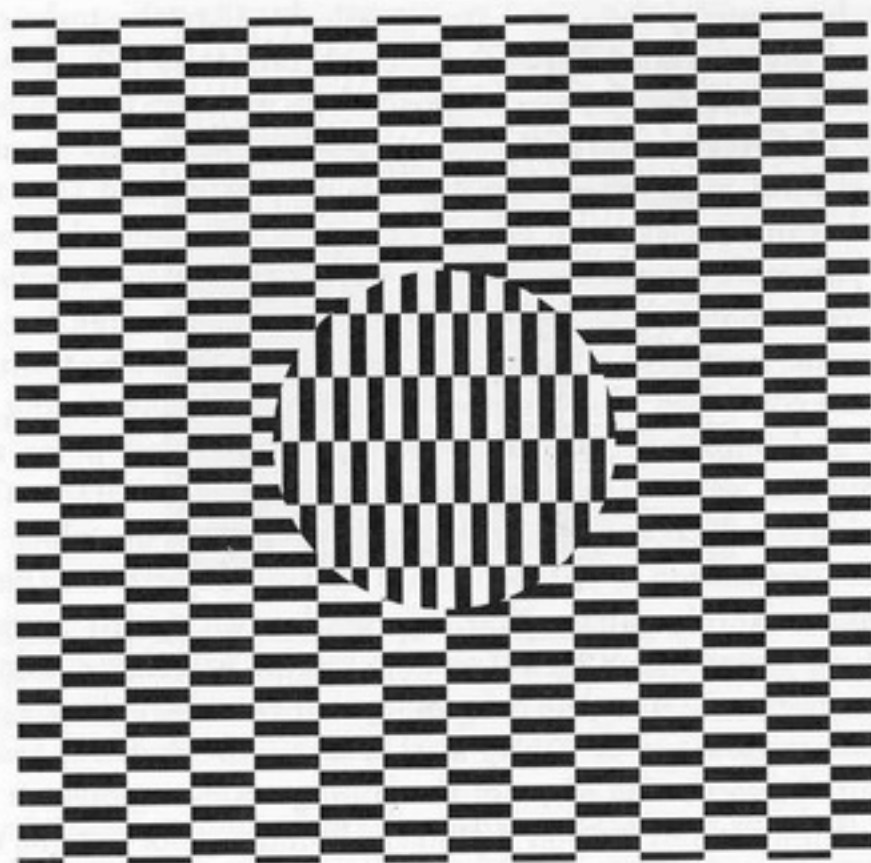


Fig. 3. A pattern similar to one by Ouchi [18]

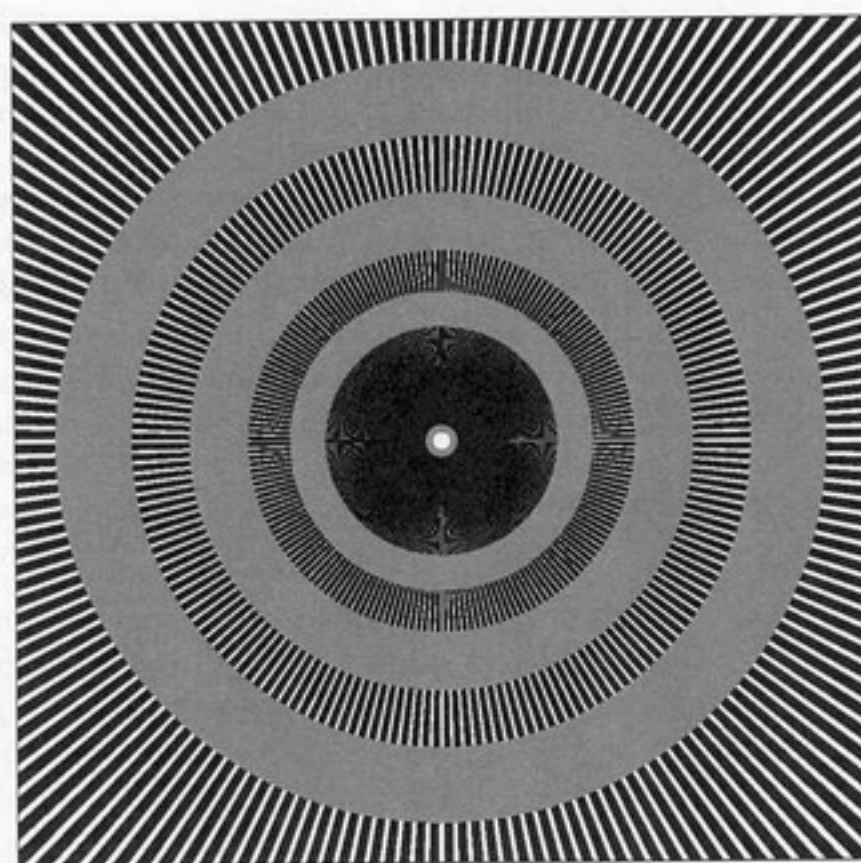


Fig. 4. *Enigma*, giving rise to the Leviant illusion: Fixation at the center results in perception of a rotary motion inside the rings

particular architecture of the visual system which solves the tasks of 3D motion estimation and segmentation through feedback processes [12]. The explanation in more detail is as follows. Small eye movements give rise to retinal motion signals. Because of *Enigma*'s particular structure motion signals occur only in the areas of radial lines; within the rings there is no texture and thus also no early motion signals can be picked up there. The motion signals, that is the normal flow vectors perpendicular to the radial lines, all belong to exactly one copoint field, the one corresponding to s being the axis passing through the center of the image. During the next processing stages, normal flow vectors from increasingly larger areas are combined to estimate the rigid motion and during these stages some spatial and temporal integration of smoothing takes place which causes motion signals also within the rings. After an estimate of 3D motion is found, this information is fed back to the earlier processing elements which then have to perform exact localization of independently moving objects. At this stage, temporal integration cancels out the flow information within the areas covered by the rays. It, however, does not cancel out the motion within the homogeneous regions since the responses from the processing elements there do not contradict an existing motion in these areas. Since, at this stage, the system's task is to accurately segment the scene, the edges of the homogeneous regions are perceived as motion discontinuities and motion within the homogeneous regions is reinforced. The defining feature of *Enigma* is that all the spatial gradients in its patterns excite only one class of vector fields involved in 3D motion estimation—a copoint field. To test our hypothesis we created figures based on the same principle as *Enigma*—that is with black and white rays giving rise to normal flow vectors corresponding to exactly one coaxial or copoint vector field and homogeneous areas perpendicular to these rays. For an example, see Fig. 5. We found that it is exactly these patterns giving rise to illusory motion. Other figures based on the same principle (rays interrupted by perpendicular homogeneous areas) which do not correspond to the vector fields involved in 3D motion estimation have been found not to give rise to an illusion, thus supporting the hypothesis.

7 Complex Actions

One may be able to derive a simple procedure for recognizing some action based on the statistics of local image movement. There exists a variety of such techniques in the current literature mostly applied to human movement or facial expressions, but according to our account of what it means to understand an action, representations that enable this sort of recognition are simply inadequate. What we need are representations that can visualize the action under consideration. Let us summarize a number of key points that are important in figuring out the nature of action representations:

1. Action representations should be view independent. We are able to recognize and visualize actions regardless of viewpoint.

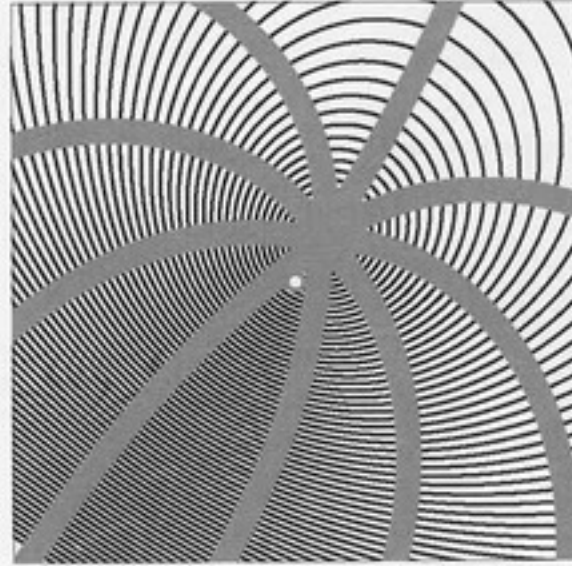


Fig. 5. Example of an illusory movement figure based on a coaxis field. Here the focal length is chosen to be about equal to the size of the image. Thus, the illusion is best experienced at a short distance

2. Action representations capture dynamic information which is manifested in a long image sequence. Put simply, it is not possible to understand an action on the basis of a small sequence of frames (viewpoints).
3. It is the combination of shape and movement that makes up an action representation.

A number of issues become clear from the impositions of the constraints listed above. *First*, to understand human action, a model of the human body should be available. *Second*, the movement of the different parts of the body relative to each other and to the environment should be recovered in a way that allows matching. *Third*, since action amounts to the movement of 3D space, action representations should contain information about both components. Let us further examine those issues in order. Acquiring a sophisticated model of the human body, although of extreme practical importance, is not essential to our discussion. We would like to caution, however, against learning approaches. It is probably an impossible task to learn such a model from visual data in a reasonable time. It is quite plausible that humans are born with such a model available to them. After all, it is a model of themselves. Later, we will describe technological solutions to this question. The second issue deserves more attention, because it introduces constraints. The previous section discussed the difficulties in acquiring 3D motion and shape, when normal motion fields were available in the whole visual field. From this we conclude that it is impossible to acquire exact 3D motion information from a very small part of the visual field, that is, it is not feasible to accurately estimate the 3D motion of the different rigid parts of the body from image sequences of it. Whatever the action representation is, it appears very hard to achieve recognition in a bottom-up fashion. The concept of feedback explained in the previous section acquires here a major importance. Recognizing action should be mostly a top-down process. Finally, the third issue reflects the need to

incorporate both shape and motion in the representation. This means that any motion representation should be spatially organized. There are, however, various options here. Do we explicitly represent movement in terms of motion fields, or do we do it implicitly in terms of forces, dynamics, and inertial constraints? Both options have their pros and cons.

8 How Could Action Representations Be?

To gain insights on action representations we consider them in a hierarchy with two more kinds of representations and the properties of the possible mappings among them. *First* there is the image data, that is, videos of humans in action. Considering the cue of motion, then our image data amounts to a sequence of normal flow fields computed from the videos. The second kind of representations are intermediate descriptions encoding information about 3D space and 3D motion, estimated from the input (video). These representations consist of a whole range of descriptions of different sophistication encoding partially the space-time geometry, and they are scene dependent. Finally, we have the action representations themselves, which are scene independent.

Before we start considering the relationships in this hierarchy, let us introduce a novel and very useful tool in the study of action. It is a tool because it can help us in this study, but at the same time it amounts to a theoretical intermediate representation that combines the advantages of view independence with the capture of the action as a whole. It is a theoretical tool in the same way that the image motion field is a theoretical construct. An intermediate representation for the specific action in view is then a sequence of evolving 3D motion fields and it is the most sophisticated intermediate description that could be obtained. Acquiring this representation is no simple matter, but it can be achieved by employing a very large number of viewpoints. We have established in our laboratory a multi-camera network (sixty-four cameras, Kodak ES-310, providing images at a rate of eighty-five frames per second; the video is collected directly on disk—the cameras are connected by a high-speed network possessing sixteen dual processor Pentium 450s). By observing human action from all viewpoints we are able to create a sequence of 3D motion fields [10]. See Fig. 6 that displays a schematic description of our facility. The solution requires a calibration of the camera network [1] and proceeds by carving the rays in space using image intensity and image motion values.

Let us now consider the mapping from the second level of the hierarchy (intermediate descriptions and, in particular, 3D motion fields) to the scene independent action representations (top level). This mapping should be such that it extracts from a specific action quantities of a generic character common to all actions of the same type. These quantities most probably take the form of spatiotemporal patterns in four dimensions.

One way of obtaining such patterns is to perform statistics on a large enough sample. Considering, for example, a particular action (e.g., walking), we can obtain data in the multi-camera laboratory described before for a large number

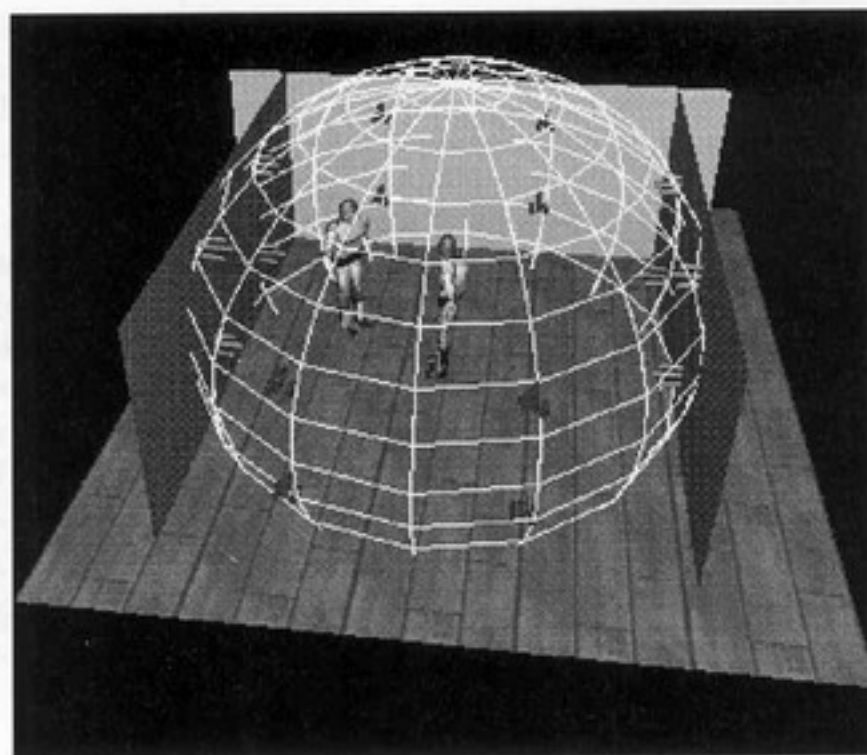


Fig. 6. A negative spherical eye

of individuals. In each case we can obtain a 3D motion field and thus have at our disposal a large number of 3D motion fields. A number of statistical techniques, such as principal component analysis, can reduce the dimensionality of this space and describe it with a small number of parameters, which parameters will in turn constitute a representation for the generic action under consideration.

Another way of obtaining these patterns would be to employ a geometric approach as well. Since a human action consists of a large number of different rigid motions, one possibility is to obtain these rigid motions from the 3D motion field, encode their relationships, and study invariances related to symmetry, and geometric quantities in space-time (angles, velocities, accelerations, periodicity, etc.).

Considering now the mappings from the top level to the image data and to the intermediate representations we can gain further insight. For the sake of recognition, action descriptions will have to be projected on the image and matched against the intermediate descriptions and the image data. But, as explained in previous sections, these intermediate descriptions are errorful and constitute distorted versions of the scene. In addition, there is variability among actions of the same type. In trying to perform this matching in space we will have to revise both the action representations and the nature of the intermediate descriptions so that the matching can be facilitated. In actual fact we may have to develop specific spatiotemporal patterns in the image data for this goal and these patterns will constitute some form of action representations for quick recognition. To conclude, the interplay among the mappings between the intro-

duced hierarchy levels seems to be a feasible and structured way for building powerful action descriptions. This constitutes our current research efforts.

9 Conclusions

Being truthful to Hartmann's and Lorenz's approach regarding evolution and the structure of intelligent systems, we wish to study intelligent systems as they actually are, using the present structures in their bodies as our data. Considering representations for a number of important complex actions, we have two options about their origin: either the system learned them or it was born with them. We find the argument about the innate character of (at least some) action representations very appealing, because humans have retained the same basic structure and have been performing, for the most part, the same repertoire of actions for a very long time. It makes good sense to build these representations using all available knowledge and technology. This paper offered a number of ways for accomplishing this goal.

References

1. P. Baker and Y. Aloimonos. Complete calibration of a multi-camera network. In *Proc. IEEE Workshop on Omnidirectional Vision*, pages 134–141, Los Alamitos, CA, 2000. IEEE Computer Society.
2. T. Brodský, C. Fermüller, and Y. Aloimonos. Structure from motion: Beyond the epipolar constraint. *International Journal of Computer Vision*, 2000. In press. Also available as CfAR Technical Report CAR-TR-911.
3. L. Cheong, C. Fermüller, and Y. Aloimonos. Effects of errors in the viewing geometry on shape estimation. *Computer Vision and Image Understanding*, 71:356–372, 1998.
4. C. Fermüller and Y. Aloimonos. Direct perception of three-dimensional motion from patterns of visual motion. *Science*, 270:1973–1976, 1995.
5. C. Fermüller and Y. Aloimonos. Qualitative egomotion. *International Journal of Computer Vision*, 15:7–29, 1995.
6. C. Fermüller and Y. Aloimonos. Vision and action. *Image and Vision Computing*, 13:725–744, 1995.
7. C. Fermüller and Y. Aloimonos. On the geometry of visual correspondence. *International Journal of Computer Vision*, 21:223–247, 1997.
8. C. Fermüller and Y. Aloimonos. Ambiguity in structure from motion: Sphere versus plane. *International Journal of Computer Vision*, 28:137–154, 1998.
9. C. Fermüller and Y. Aloimonos. Observability of 3D motion. *International Journal of Computer Vision*, 2000. In press.
10. C. Fermüller, Y. Aloimonos, P. Baker, R. Pless, J. Neumann, and B. Stuart. Multi-camera networks: Eyes from eyes. In *Proc. IEEE Workshop on Omnidirectional Vision*, pages 11–18, Los Alamitos, CA, 2000. IEEE Computer Society.
11. C. Fermüller, L. Cheong, and Y. Aloimonos. Visual space distortion. *Biological Cybernetics*, 77:323–337, 1997.
12. C. Fermüller, R. Pless, and Y. Aloimonos. Families of stationary patterns producing illusory movement: Insights into the visual system. *Proc. Royal Society, London B*, 264:795–806, 1997.

13. C. Fermüller, R. Pless, and Y. Aloimonos. The Ouchi illusion as an artifact of biased flow estimation. *Vision Research*, 40:77–96, 2000.
14. C. Fermüller, D. Shulman, and R. Pless. The statistics of optical flow. Technical Report CAR-TR-928, Center for Automation Research, University of Maryland, 1999.
15. N. Hartmann. *Der Aufbau der Realen Welt*. de Gruyter, Berlin, 1964.
16. J. J. Koenderink and A. J. van Doorn. Relief: Pictorial and otherwise. *Image and Vision Computing*, 13:321–334, 1995.
17. K. Lorenz. *Behind the Mirror: A Search for a Natural History of Human Knowledge*. Harcourt Brace Jovanovich, New York, 1977.
18. H. Ouchi. *Japanese and Geometrical Art*. Dover, 1977.
19. M. Polanyi. *Personal Knowledge: Towards a Post-Critical Philosophy*. The University of Chicago Press, Chicago, 1958.
20. S. M. Zeki. The cortical *enigma*: A reply to Professor Gregory. *Proc. Royal Society, London B*, 257:243–245, 1994.
21. S. M. Zeki. Phenomenal motion seen through artificial intra-ocular lenses. *Proc. Royal Society, London B*, 260:165–166, 1995.

Contributed Papers

Abstracts

Contributed Papers

The first paper in this section is by Chris Smith and Chris Smith. It is titled 'Genetic Algorithms for flower growth'. The paper describes a genetic algorithm for simulating the growth of a flower. The algorithm is based on a model of the flower's growth process. The model is a simple representation of the flower's growth process. The algorithm is used to simulate the growth of a flower. The results of the simulation are shown in the paper.

The second paper in this section is by Chris Smith and Chris Smith. It is titled 'Genetic Algorithms for flower growth'. The paper describes a genetic algorithm for simulating the growth of a flower. The algorithm is based on a model of the flower's growth process. The model is a simple representation of the flower's growth process. The algorithm is used to simulate the growth of a flower. The results of the simulation are shown in the paper.

The third paper in this section is by Chris Smith and Chris Smith. It is titled 'Genetic Algorithms for flower growth'. The paper describes a genetic algorithm for simulating the growth of a flower. The algorithm is based on a model of the flower's growth process. The model is a simple representation of the flower's growth process. The algorithm is used to simulate the growth of a flower. The results of the simulation are shown in the paper.

Contributed Papers

The first paper in this section is by Chris Smith and Chris Smith. It is titled 'Genetic Algorithms for flower growth'. The paper describes a genetic algorithm for simulating the growth of a flower. The algorithm is based on a model of the flower's growth process. The model is a simple representation of the flower's growth process. The algorithm is used to simulate the growth of a flower. The results of the simulation are shown in the paper.

The second paper in this section is by Chris Smith and Chris Smith. It is titled 'Genetic Algorithms for flower growth'. The paper describes a genetic algorithm for simulating the growth of a flower. The algorithm is based on a model of the flower's growth process. The model is a simple representation of the flower's growth process. The algorithm is used to simulate the growth of a flower. The results of the simulation are shown in the paper.

The third paper in this section is by Chris Smith and Chris Smith. It is titled 'Genetic Algorithms for flower growth'. The paper describes a genetic algorithm for simulating the growth of a flower. The algorithm is based on a model of the flower's growth process. The model is a simple representation of the flower's growth process. The algorithm is used to simulate the growth of a flower. The results of the simulation are shown in the paper.

The fourth paper in this section is by Chris Smith and Chris Smith. It is titled 'Genetic Algorithms for flower growth'. The paper describes a genetic algorithm for simulating the growth of a flower. The algorithm is based on a model of the flower's growth process. The model is a simple representation of the flower's growth process. The algorithm is used to simulate the growth of a flower. The results of the simulation are shown in the paper.

Genetic Algorithms for flower growth

Joy Lu

University of Bath
Bath BA2 7AY, UK.

Claire Willis

University of Bath
Bath BA2 7AY, UK.
cpw@cs.bath.ac.uk

Derek Paddon

University of Bath
Bath BA2 7AY, UK.
djp@cs.bath.ac.uk

Abstract

We describe a method for generating flower growth animation in which a petal surface and shape can be evolved in real time. The novelty of the method proposed here is the use of a general framework that combines a mass-spring model with control points embedded in a bicubic patch formulation. In addition, genetic algorithm techniques have been used to optimise the collision avoidance among the flower organs and the environment. The model generates non-deterministic evolutionary results which give more realistic and varied growth than can be obtained using pre-defined surfaces or interpolating between given initial and final shapes.

1 Introduction

The use of computer graphics methods to simulate and animate biological entities has been continuously developed since computers became available to scientists in the second half of the last century (Durik98; Fowle92; Linte99). However, the animation of flowers has been neglected, other than very simplistic methods that use libraries of pre-defined shapes and components for the user to build into completed entities.

This research describes a method for generating flower growth animation in which a petal surface and shape can be evolved in real time. Most plant modelling systems animate the plant development process by assuming a time interval and a corresponding growth direction. Unfortunately, such a method is restrictive in that one cannot easily change the time step or deform the shape in a natural way (Prusi93).

We present a free-form model for surface deformation and growth. The novelty of our method is the use of a general framework that combines a mass-spring model with control points embedded in a bicubic patch surface representation. The key component is to enable the control points to move, such that we achieve satisfactory results when the surface growth is constrained by an evolutionary theory. This evolution theory must take account of natural and artificial perturbations in the growth cycle. This implies that the transient movement of the control points must be flexible and adaptable, thus the model generates non-deterministic results which give more realistic and varied petals than can be obtained by using pre-defined surfaces or interpolating between given initial and final shapes.

2 Mathematical Framework

Since petals or leaves change in shape and form during their life-cycle, it is important to construct a suitable dynamic surface model to represent this life-cycle. A popular dynamic surface model is the classic mesh method. This has been widely used to model complex smooth surfaces, such as those encountered in cloth (Provo95) and human character animation. However, these methods suffer from at least two difficulties. First, the computation is expensive when the number of mesh nodes is very large. Second, it is difficult to maintain smoothness, especially when mesh nodes need to be repositioned during animation.

Bicubic patches have the potential to overcome both of these problems and to effectively model surface growth. Bicubic patches have limited numbers of control points, and smoothness of the model is automatically guaranteed.

2.1 Model structure

Though our application is concerned with petal surface growth instead of simple surface deformation, the mass-spring model structure appears to be highly suitable for applying the growth function to our bicubic patch surface. If we consider the match between the mass-spring model and our requirements for growing a surface, we find the following:

1. The control points must be able to move with the related growth function. The mass-spring model has similar control for its mass-points. So we can easily use control points as mass-points.
2. The most important feature of a plant organ is that it grows, however, petal growth is not equivalent to stretching. It includes size change, shape change and mass change, and maybe even distortions if there is external intervention, such as an obstruction. With

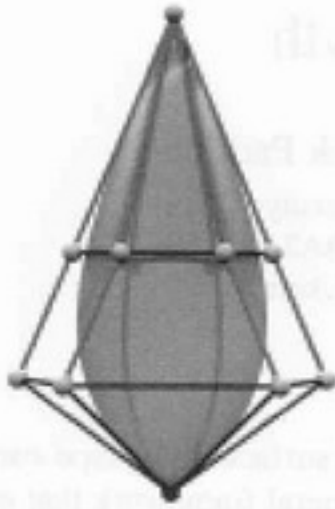


Figure 1: Three-dimensional petal surface with mass control points and spring links (Note, apart from the discontinuity points, all control points are above the petal surface.)

all the links provided by springs, we can reposition the control points on the surface to enable such growth to be animated.

3. The petal surface itself has mass, therefore gravity is one of the forces helping it to open. Gravity is also related to the shape change on the petal surface because it applies a direct force at the control points. We can view the natural growth of a flower as a force that pushes all the control points apart (such as along the length growth direction or along the width growth direction to grow a longer and wider petal surface).

Of course, external intervention is simply another external force on the petal, so the growth function can be wholly converted to forces applied to the surface.

4. Even though a force may only apply to one control point of the surface, the growth will affect all the linked neighbouring points.

From the above discussion, we have sufficient justification to introduce a new method which combines the bicubic patch with the mass-spring model. Figure 1 shows a petal with mass control points and spring links. In this approach the model of the petal is not considered as a continuous surface, but instead will be represented as a discrete structure of elements where each mass-point and each spring can be handled individually. We must now consider in more detail how to apply the mass-spring model to a petal surface.

2.2 Dynamics and Forces

The system under study is the bicubic patch of masses, each mass being positioned at time t at the point $\mathbf{p}_i$, where $\mathbf{p}_i$ and $\mathbf{v}_i$. The evolution of the system is governed by Newton's law of dynamics:

(1)

where m_i is the mass of each point and $\mathbf{a}_i$ is its acceleration caused by the force $\mathbf{F}_i$. $\mathbf{F}_i$ can be divided between the internal and external forces. The internal force is the resultant of the tensions of the springs linking $\mathbf{p}_i$ to its neighbours, which are called contact and constraint forces:

(2)

where $\mathbf{S}_{ij}$ is the set regrouping all couples $\mathbf{p}_j$ such that $\mathbf{p}_i$ is linked by a spring to $\mathbf{p}_j$. $\mathbf{r}_{ij}$ is the vector from $\mathbf{p}_i$ to $\mathbf{p}_j$, and k_{ij} is the stiffness of the spring linking $\mathbf{p}_i$ and $\mathbf{p}_j$.

However, there is another important internal force in our model, this is the force representing the growth function of the petal surface. From the growth function, we can derive the growth rate function which, typically, is slow initially, accelerating to reach a maximum, then slowing again and eventually ceasing when the petal has reached its final size and shape. We take this growth into account via the internal forces which push the control points away from each other in the direction of growth. The forces will act in three growth directions, which are related to length, width and curvature growth. The internal growth force is given by:

(3)

where $\mathbf{a}_g$ is the acceleration of the growth rate in the time interval Δt . The important feature of these forces is that they are integrated with the growth function. At the final stage described by a natural growth function, there is no growth force and the petal will only deform under external forces.

A plant can be subjected to external forces in various ways, such as simple deflection, obstructing forces, periodic forces, etc. One obvious force will be gravity, let $\mathbf{g}$ be the acceleration of gravity, then the force on $\mathbf{p}_i$ is given by:

(4)

Other external forces may come from natural perturbations, such as collisions with other flower petals, and artificial perturbations, such as human intervention. These forces will apply to specific control points which will change the positions of some control points to generate the new growing direction.

In addition, we need to take into account the mass of each control point. In our model, these masses are not constant, but vary with the growth. This is different from

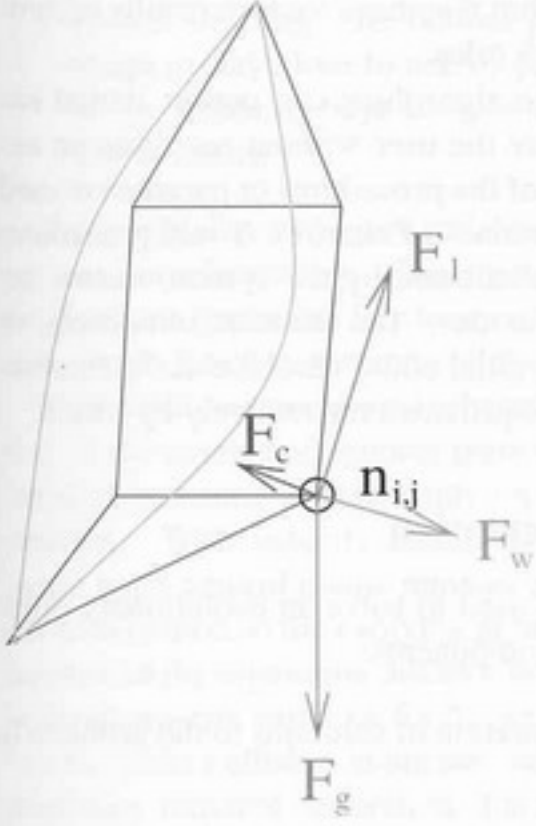


Figure 2: Petal forces directions at a mass point

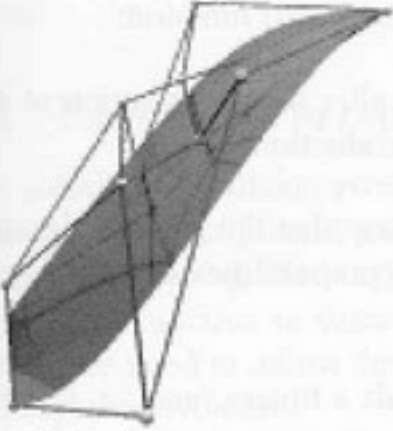


Figure 3: Growing petal surface with mass control points

other mass-spring models. Most mass-spring models are for flexible objects, such as cloth, where the mass normally remains constant during deformation. However, our petal surface is a growth surface, therefore it is obvious that its mass will increase according to the applied growth function until it reaches a maximum value.

Figure 2 gives an example of one mass-point with all related forces in relevant directions (except the constraint forces which are along the linked springs). F_l represents the gravity, F_w represents the length growth force, F_c represents the width growth force and F_g represents the force from the curvature change.

2.3 Results from our model

Our method can generate a smoothly curved surface with only a few forces applied to the control points. However, if those control points are nodes on a traditional mesh, they cannot form a smooth surface without the key curvature points. Many extra mesh nodes must be added to

the current mesh, each containing details of forces and growth information, which is difficult for real-time animation. Because of these difficulties we reject the traditional mesh method as inappropriate for animating petals.

However, the combination of bicubic patches and the mass-spring model delivers a very efficient method to represent and directly control the growth of a petal surface. This method is flexible and creative, while at the same time requiring considerably less computation than a traditional mesh.

3 Collisions

Collision and deformation for free form surface modelling remains a challenging area. Collision handling is an important task for any growing surface model, and must include both collision detection and collision response.

In nature, individual flowers touch each other, which modifies their positions and shapes. Consequently, the organs must be carefully modelled and sized to avoid intersections. This is feasible while modelling static structures, but proper simulation of collisions would become crucial in the realistic animation of plant development.

3.1 Collision detection

The simplest approach to detecting a collision in a physical simulation is to test every geometric element, such as point, edge, face, against each other geometric element for a possible collision (DeRos98; Mech96). To achieve practical running times for large simulations, the number of possible collisions must be culled as rapidly as possible using some type of spatial data structure. If the surface connectivity does not change, a two-dimensional surface-based data structure can be used for the elements. Another method is to distribute the elements into a three-dimensional volume-based data structure. However, this is very expensive in the computation and subdivision of the bounding boxes of all the elements, especially when all the elements are surfaces. This is because surface intersection has a very high computation cost and it is not easy to balance the three conflicting goals of accuracy, robustness, and efficiency.

Our method will focus on how to construct a suitable surface-based data structure for the petal surface. As each surface patch has sixteen control points, we can use each group of four control points as a data structure. An efficient way of constructing the hierarchy of boxes is to compute the bounding surfaces using the convex hull property (Lu01).

If there is no intersection between the two horizontal convex polyhedra (from the two middle sets of control points) and the vertical convex polyhedron, there is no collision between the two adjacent surfaces. If there is an intersection, it does not imply that a collision has occurred, the surfaces may or may not be in collision. We must calculate the exact surface points to determine the

result (one possible approach would be to use forward differencing techniques to calculate the points).

3.2 Collision response

After the collision is detected, we must also consider how to deform and arrange the surface to avoid further intersections.

An underlying model (MacCr96) can be deformed by establishing positions of the model's points within the converging sequence of lattices and then tracking the new positions of these points within the deformed sequence of lattices.

An efficient deformation process must be developed for the free-form surface. Such a process must meet a number of requirements. Firstly, the movement space of the control points is established from the result of the above analysis. This gives us a specific space for the control points, which makes it possible to find the optimal position for the control points. Secondly, the method should be applied for efficient deformation for the whole group of control points. Thus, we need to develop a technique or function for controlling all the control points. Thirdly, surface smoothness features and growth principles must be maintained throughout the deformation process.

From the collision analysis and growth direction of the control points, we suggest a more efficient deformation method to respond to collisions based on genetic algorithms. In this way, we will be able to predict when the control points will form a collided surface and avoid the collision, which makes it possible to have a more natural surface growth. We will discuss how to implement this in the next section.

4 Genetic Algorithms for Surface Control

The surface shape is converted into a mathematical framework with the surface control provided by internal and external forces. However, these forces applied to the surface may not generate a natural surface, particularly if the surface is in collision. It is inappropriate to require the user to adjust the growth force to avoid all collisions, since this could mean that the user would have to calculate and place the control points very carefully. Fortunately, as an alternative, optimisation techniques offer possibilities for the automatic generation of complex systems. The genetic algorithm is a form of artificial evolution, and is a commonly used method for optimising complex systems.

Genetic algorithms are search algorithms based on the mechanics of natural selection and natural genetics (Goldb89; Rana95). They combine survival of the fittest among string structures with a structured yet randomised information exchange, to form a search algorithm with some of the innovative flair of human search. It means

that the same algorithm may have various results according to different search rules.

The use of genetic algorithms can permit virtual entities to be created by the user without requiring an explicit understanding of the procedures or parameters used to generate them (Brunn97; Zakar01). These procedures and parameters, implemented by the system creator, remain hidden from the user. The measure of success, or fitness, of each individual entity can be calculated automatically, or it can be provided interactively by a user.

4.1 Model Description

A genetic algorithm used to solve an evolutionary problem must have five components:

1. a string representation of solutions to the problem (a genome);
2. a way to create an initial population of solutions;
3. an evaluation function that plays the role of the environment, rating solutions in term of their "fitness"; this is usually called the fitness function;
4. genetic operators that alter the composition of control factors during reproduction;
5. values for the parameters that the genetic algorithm uses (population size, probabilities of applying genetic operators, etc.).

A genetic algorithm needs a fitness function to be defined for a given problem. It is an evaluation function which evaluates if the search result is fit for the application. In this section we develop fitness function criteria that are appropriate for our surface control.

The choice of a fitness function needs careful consideration. The convergence of the algorithm depends crucially on the relative search size of the whole solution space. In addition, we must create a cost function that has some meaning, that is, we aim to find an intermediate qualitative measure which counts the number of good controls. This will then be transformed into a function which will be the fitness passed to the genetic algorithm.

To build a simulated growth system, we start with a model that supports geometric structure. From the analysis of the geometric structure, we can define the criteria for the fitness function of the petal surface model. We add behaviours that correspond to the opposing forces of collision avoidance and the urge to grow. The behaviours that lead to simulated growth for all the petals are:

1. Collision avoidance: avoid collisions with nearby petals as the petals grow and change shape;
2. Velocity matching: attempt to match velocity with nearby petals when they are open together;

3. Flower centring: the bottom part of the petals attempt to stay close to nearby petals and the top part of the petals attempt to open along the line to the flower centre.

Static collision avoidance and dynamic velocity matching are complementary. Together they ensure that the petals of one flower are free to open within the constraints and growth function without colliding with one another.

Static collision avoidance is based on the relative position of the petals and ignores their velocity. Conversely, velocity matching is based only on velocity and ignores position. With velocity matching, the separations between petal control points remains approximately invariant with respect to the evolving geometric structure. This means that the separation distance between petals remains in the geometric structure for the next time step. In other words, static collision avoidance serves to establish the minimum required separation distance between petals, whereas velocity matching tends to maintain it.

Flower centring is another important rule to keep all the petals open within the constraints of the flower boundary and also provides the opening directions for the flower petals.

4.2 Parameters to represent the problem

The solution to collision avoidance by genetic algorithms requires that we adjust the parameters in the growth function while keeping the petals growing. As we convert the growth function to forces controlling a mass-spring model, we need to adjust the forces for the petal growth and collision avoidance.

As a result, the parameters required to represent the collision problem are the same as those controlling the surface change, which relate to the changing forces applied to the surface control points. In our implementation, these parameters are length, width, surface curvature and flower opening speed.

4.2.1 Growth values for the parameters

We know that genetic algorithms work with a coding of the parameter set, not the parameters themselves. In order to find the probabilities of applying genetic operators to the parameters, we need to provide the values for the parameters that genetic algorithms use. In this way we will be able to create the possible growth directions for all the control points in our petal surface model.

Boolean values provide a simple encoding method for the surface model. In our system, we set the value of 0 or 1 for the force parameters, which are in different directions according to the related growth parameters. A value 1 indicates that the growth can continue in its direction and value 0 means the growth will not happen in that direction. For example, if in a specific time step, a control mass-point has growth in length, width, surface curvature and flower opening direction, the coding value for all the

force parameters applied on that point is 1111. However, if the petal will incur a collision if it continues to grow wider, we might have the force parameters code as 1011, which means the width growth will not happen. It is not always clear which factors will eventually cause a collision. Thus, searching for a suitable code is the task of the genetic algorithm.

4.3 The fitness function

The fitness function determines if the forces applied meet our criteria of natural growth without incurring collision. In our application, we consider three requirements for realistic growth: collision avoidance, velocity matching and flower centring.

The collision penalty depends on collisions between petals at corresponding points on the petal surface model, thus the equation for the collision avoidance factor is given by:

(5)

where P_i represents all the points on one petal model, from 1 to N . C_i represents the penalty on point i . It is 1 if there is collision, and 0 if there is not.

The distance moved in a given direction during a unit of time determines the velocity. Thus the velocity matching factor is given by the distance metric:

(6)

where d_i is the distance that point i travel during a time interval, and d_j is the distance that point j travel during the same time interval. We must emphasise that point i is not from the same petal as point j , instead it is from the adjacent petal.

The flower centring factor is given by:

(7)

where $\vec{v}_i$ is the vector at point i , with the direction from the previous position pointing to the new position after the time interval. $\vec{h}$ is the vector along the horizontal line going through the flower centre. It is obvious that will be 0 if these two vectors are perpendicular.

Thus, the fitness function is given by:

(8)

where α , β and γ are constants. The fitness function attempts to place a penalty on any movement of the surface that does not meet the criteria for collision avoidance. Thus, the value of F is kept relatively high, since collisions are to be avoided at all costs. The value of F on the other hand is kept very low, since normally the force can keep the petal moving along the flower centre direction. It is obvious the algorithm should try to search the solution space for the lowest F value for petal growth.

4.4 Genetic operations

The basic genetic operations for the genome include reproduction, crossover, and mutation. The reproduction operator determines how the genome is initialised. It is formed by the order of a list of parameters from the forces applied to the specific point on the petal surface model. The mutation operator defines the procedure for mutating each genome. A typical mutator for a binary string genome simply flips the bits in the string (see figure 4) with a defined probability. In our approach, we randomly perturb the control points force for opposite growth direction. That is, if there is a collision, an opposite force will be generated to push the point in the opposite direction. The crossover operator defines the procedure for generating a child from two parent genomes. Here, we use a standard single-point crossover to generate offspring (see figure 5). Given two individuals (possible solutions to a problem) in a population, the genetic algorithm generates a child solution by randomly taking some of the genes from one individual, and taking the remaining genes from the other.

before	1 0 1 1 0 0 0 1 1 1 1
after	1 0 1 1 0 1 0 1 1 1 1

Figure 4: Example of mutation in binary encoding

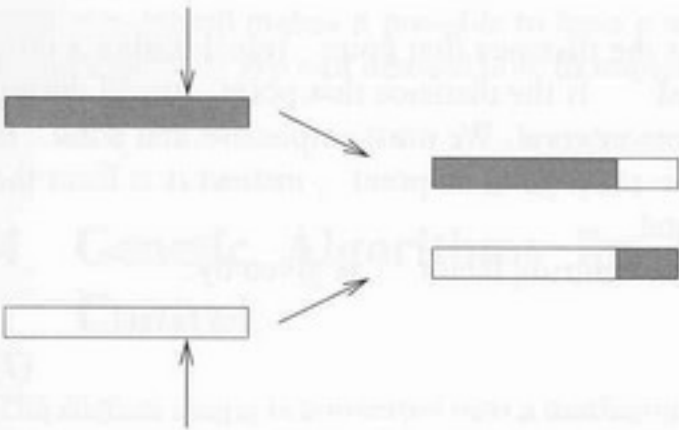


Figure 5: Single point crossover

4.5 Results

In the traditional method, without a search optimisation algorithm, the user must choose the growth force for each control point, such that collisions are avoided. Choosing these forces to represent natural growth is almost impossible, with the inevitable result that the control points are moved either too little or too much at each time step. Choosing the correct force would require the user to have foresight about the relative movements of the associated control points.

We can see from Figure 6 that as the flower grows the major part of the three petals cross each other from state

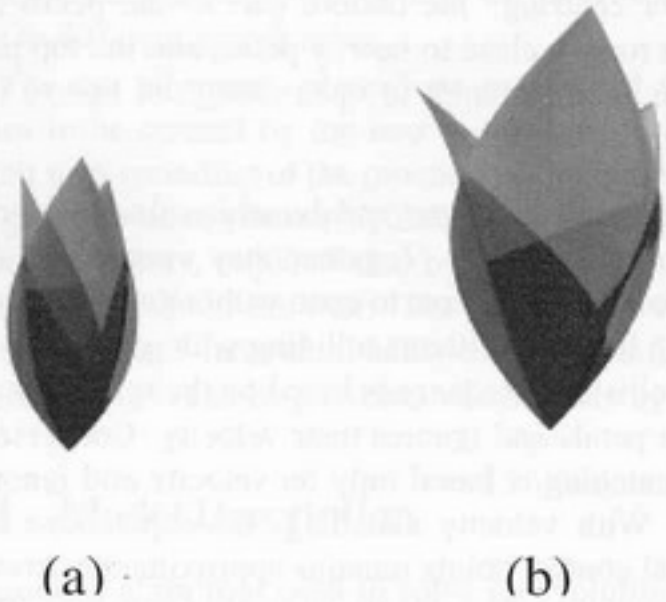


Figure 6: Flower growth without genetic algorithm

to if we allow the flower to grow with the parameters as set by the user.

However, with the genetic algorithm, the forces are changed in both value and direction under the control of the fitness function and the genetic evolution. Figure 7 shows the effect of using a genetic algorithm. Here at each stage of growth the petals have maintained a tight formation but without any intersection of petals. Without the use of the genetic algorithm this particular result would have been very difficult to derive, even if the user was highly skilled and modified the growth parameters at each time step. Thus, with the use of the genetic algorithm the forces have been automatically modified at each time step to ensure that intersecting petals cannot occur.

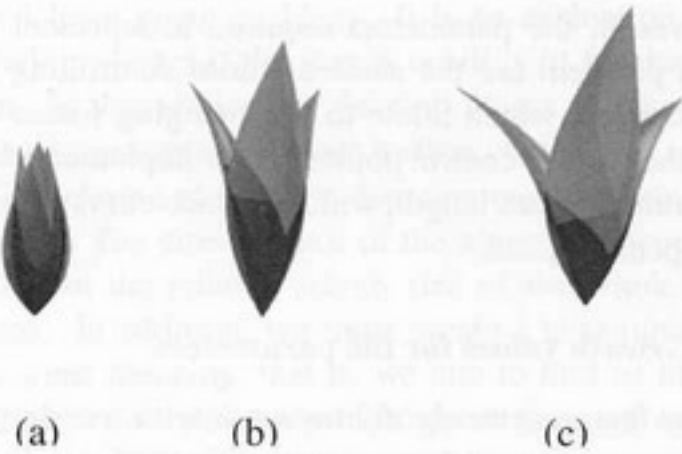


Figure 7: Flower growth with genetic algorithm

5 Conclusion

We have presented a novel system for creating virtual flowers that grow and behave in a simulated three-dimensional physical world. The modifications of the growth controlling forces on the surface are generated automatically using genetic algorithms. Suitable fitness evaluation functions are developed to guide the simulated evolutions towards specific interventions such as collisions and other artificial interruptions.

The application of genetic algorithms has allowed a flower to develop in a realistic manner. The absolute fitness functions can be defined by the user, thus allowing automatic selection of fit individuals and enabling growth forces to evolve without user intervention. The search is also guided by human creativity, through the choice of fitness functions to produce more interesting and acceptable results. So the user still has control over the flower shape. The summation of these advantages has enabled a flower to develop in a realistic manner.

The ability to use genetic algorithms was a direct consequence of developing the mass-spring model. Without access to the dynamic forces involved in collision detection and collision avoidance, we would not have been able to build a relevant fitness function. Therefore, there is additional benefit in adopting the combined surface and mass-spring model.

In addition, we have been able to simulate physical flower growth which handles collisions in a more realistic way than can be achieved by a skilled human animator, who would intervene at various time steps to halt collisions of flower parts.

Many previous models of plant growth avoid the modelling of surface detail, often resorting to imposing details such as leaves and petals from a library of simplistic elements. In this work we have demonstrated that such surface detail can be effectively represented by dynamic elements that grow and deform naturally as the plant evolves through its lifecycle.

References

- [Brunn97] Brunnstrom,K, Stoddart,A: Genetic Algorithms for Free-Form Surface Matching, *13th International Conference on Pattern Recognition*, 1997.
- [DeRos98] DeRose,T, Kass,M, Truong,T: Subdivision Surfaces in Character Animation, *SIGGRAPH*, pp.85-94, 1998.
- [Durik98] Durikovic,R, Kaneda,K, Yamashita,H: Animation of Biological Organ Growth Based on L-systems, *Computer Graphics Forum*, Vol. 17, No. 3, 1998.
- [Fowle92] Fowler,D, Prusinkiewicz,P, Battjes,J: A Collision-based Model of Spiral Phyllotaxis, *Computer Graphics*, Vol. 26, No. 2, pp.361-368, July 1992.
- [Goldb89] Goldberg,D: *Genetic Algorithm in Search, Optimization, and Machine Learning*, Addison-Wesley, 1989.
- [Linte99] Lintermann,B, Deussen,O: Interactive Modeling of Plants, *IEEE Computer Graphics and Applications*, pp.56-65, Jan/Feb, 1999.
- [Lu01] Lu,Z, Willis,C, Paddon,D: Surface Animation for Flower Growth, *Mathematical Methods for Curves and Surfaces: Oslo 2000*, Vanderbilt University Press, 2001.
- [MacCr96] MacCracken,R, Joy,K: Free-form Deformations with Lattices of Arbitrary Topology, *SIGGRAPH*, pp.181-188, 1996.
- [Mech96] Mech,R, Prusinkiewicz,P: Visual Models of Plants Interacting with Their Environment, *Proceedings of SIGGRAPH*, pp.397-410, 1996.
- [Provo95] Provot,X: Deformation Constraints in a Mass-Spring Model to Describe Rigid Cloth Behavior, *Proc. Graphics Interface*, 1995.
- [Prusi93] Prusinkiewicz,P, Hammel,M, Mjolsness,E: Animation of plant development, *SIGGRAPH*, pp.351-360, 1993.
- [Rana95] Rana,A, Zalzal,A: An Evolutionary Algorithm for Collision Free Motion Planning of Multi-arm Robots, *Proceeding of Genetic Algorithms in Engineering Systems: Innovations and Applications*, pp.123-130, 1995.
- [Zakar01] Zakaria,M: Interactive Evolutionary Approach to Character Animation, *WSCG*, 2001.

An artificial retina with a self-organised retinal receptive field tessellation

L. S. Balasuriya and J. P. Siebert

Department of Computer Science

University of Glasgow

sumitha@dcs.gla.ac.uk

psiebert@dcs.gla.ac.uk

Abstract

In this paper we present a means to generate a receptive field tessellation for an artificial retina using a self-organisation methodology. The approach addresses the problem of creating a receptive field tessellation that seamlessly changes from the uniform fovea to the space-variant periphery of a retina. We sampled and reconstructed images by placing overlapping space-variant receptive fields on this retinal tessellation and scaling the field size according to the local node density. The results of sampling and reconstructing images using the above self-organising retina tessellation are presented.

1 Introduction

An artificial retina that can sample an image or scene in a space-variant manner is an essential front end to a biologically inspired machine vision system. This sampling defines the information content, representation and processing of the entire vision system. Researchers have reported problems computing plausible retinæ that sample an image without over-sampling the central foveal region or creating discontinuities in the retinal tessellation as the spatial distance between sampling points increases with eccentricity (refer section 3).

Conventional approaches for creating retinal tessellations have been based on analytic transforms. However, the authors question the tractability of the problem, from an analytic perspective, that meets the constraints of a continuous regular (in the fovea) to log-polar (in the periphery) sampling regimen.

This paper describes the design, implementation and initial evaluation of a space-variant, continuous retina with receptive field locations that were determined by self-organisation. This is essentially an artificial (software) sensor that can be used to sample an image. Simple difference of Gaussian receptive fields (Marr, 1982) with varying receptive field sizes were placed on the generated retinal tessellation to extract visual information channels from an input image that correspond to the distinct layers of the X visual pathway. This multi-resolution, multilayer representation of the input image has been structured to form a space-

variant image pyramid (Burt and Adelson, 1983). We have demonstrated the inverse process of superimposing the retina-space image reconstructed from each layer of our pyramid to regenerate a foveated version of the original input image.

2 Motivation and Approach

The physiology of the primary visual cortex comprises uniform units of neurological machinery that processes the whole visual field of view (Hubel and Wiesel, 1979) arriving from the retina. Therefore, a biologically motivated artificial retina should be capable of extracting visual information such that it can be processed by uniform computational machinery in an artificial cortex.

Wilson (1983) showed that retinal structures could extract information such that the uniform cortical machinery would output an 'ensemble of messages' that does not change with differences in scale and orientation of an object for a given point of fixation. The implementation of such a system can then result in the coding of an object irrespective of its scale and orientation.

In order to achieve image coding uniformity, it is necessary to implement a retina tessellation model that similarly preserves sampling continuity. However, real retinas appear to exhibit an almost uniform hexagonal receptor packing structure in the fovea that transforms seamlessly into an exponential

sampling structure (log-polar retino-cortical transformation) towards the peripheral field. While the exponential sampling strategy of the retina periphery appears to have many desirable properties, capable of ameliorating subsequent visual information processing (Schwartz, 1977 and Wilson, 1983), it must transform into a uniform fovea without generating discontinuities to allow a single set of uniform 'coding units' to be constructed in the cortex. These coding units would then be able to represent the local (multi-resolution) image features (together with their topographic relationships) to generate Wilson's "ensemble of messages". Subsequent higher analysis might take the form of dynamic link matching (von der Malsburg, 1981), classical graph matching or more radical schemes such as spike coding (Thorpe and Gautrais, 1997).



Figure 1: The standard Lena input image that was used for all image processing operations in this paper.

While continuity in the sampling strategy is the key to achieving a uniform coding strategy, the nature of the particular form of space-variant sampling adopted is also an issue. Exponential sampling provides the means to achieving quasi-scale invariance and a conformal retina-to-cortical mapping ensures that local codings retain their approximate invariance to scale changes.

The retina model itself can be implemented within a filter-sample scheme, where bandpass filters are located at locations defined by the exponential sampling strategy and these filters are scaled spatially to ensure that the output cortical image does not contain aliasing components. In other words, a frequency shift (frequency normalisation) is taking place in the sampled retinal image with respect to sampling eccentricity that results in uniform cortical

(radial) spatial frequencies for exponentially increasing retinal (radial) spatial frequencies.

A foveated, space-variant extraction of visual information from an image prevents the combinatorial explosion of information and computational processing suffered by conventional approaches where the whole image is given equal processing emphasis. A retina will reduce the dimensionality and bandwidth of the image data that is being sampled and processed. When the retina is fixated at a salient object in the image the important foveal region of the image is sampled with a very high sampling density while the less important peripheral region is given less emphasis.

Researchers have not been able to define a retina with the above properties that can extract visual information from an image without over-sampling the image or creating discontinuities in the retina tessellation (section 3). In fact, there are few examples of full vision systems that classify or conduct recognition tasks with information extracted from a space-variant retina (Smeraldi et. al., 1998).

Solving the problem of generating a viable retina model will allow biologically motivated vision to progress and therefore if we can construct a continuous retinal tessellation that circumvents the mentioned limitations, this can drive attentional structures which are part of an overall vision system.

The objective of this paper is to present an approach to create a retina for such an artificial visual pathway.

3 Previous retina models

Most space-variant retinæ used in conventional machine vision systems are based around a log-polar sensor. The mapping from image space to what is called a 'cortical space' is computed according to the $\log(z)$ model proposed by Eric Schwartz (1977). If $z = x + jy$ is the location (in complex coordinates) of a receptive field centre in the image plane, then the location of its output in the artificial cortex is given simply by $\log(z)$ where,

$$\begin{aligned}\log(z) &= \log(|z|) + j \arg(z) \\ &= \log(\text{eccentricity}) + j (\text{angle})\end{aligned}$$

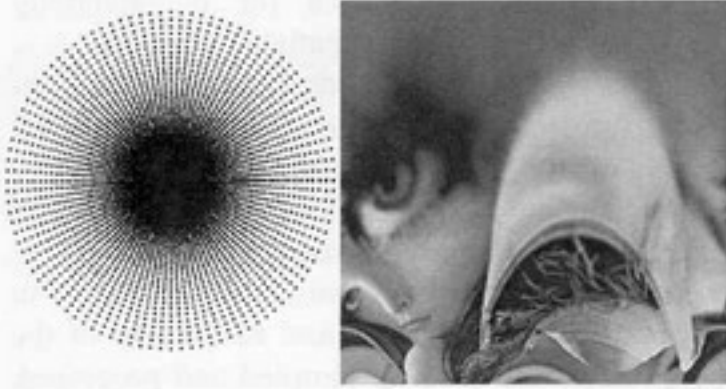


Figure 2: A retina with receptive field locations based on the $\log(z)$ model (left). A cortical images created by sampling the standard Lena image with Gaussian receptive fields placed on the retina (right).

Higher level processing can now be conducted on the cortical image (figure 2) as this is a space-variant, foveated representation of the image. Because an image has an underlying minimum spatial frequency any attempts to sample an image at higher spatial frequencies would result in a highly correlated output. This super-Nyquist sampling can be observed in the foveal region of the $\log(z)$ retina (figure 2). In fact a large percentage of the cortical image is highly correlated information and the sampling process has not optimally reduced the dimension of the extracted visual information.

The $\log(z+\alpha)$ model by Schwartz(1980) tried to avoid the singularity at the fovea and reduce the over-sampling in the foveal region, however the highly correlated foveal region in the cortical image can still be seen for realistic values of α (figure 3). Furthermore the $\log(z+\alpha)$ retinal tessellation has an anisotropic fovea which is elongated vertically. An interesting feature of the $\log(z+\alpha)$ model is that the cortical image is split along the vertical meridian into two visual hemispheres as in biology.



Figure 3: A schematic representation of receptive field locations based on the $\log(z+\alpha)$ model (left). A cortical images created by sampling the standard Lena image with Gaussian receptive fields using the $\log(z+\alpha)$ model (right).

Researchers have also tried to separate the foveal and peripheral regions of the retina (Bolduc and Levine, 1998) and thereby uniformly sample the foveal region while sampling the peripheral region with a space-variant tessellation. However one can see that this creates a distinct discontinuity in the sampling between the fovea and periphery (figure 4).

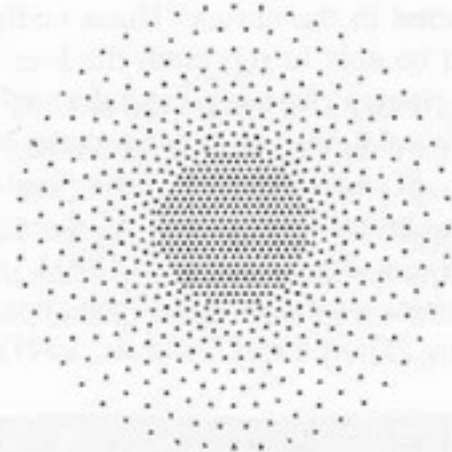


Figure 4: A plot of the receptive field centres from a retina with a uniform foveal region.

We haven't found any analytic or geometric mapping that can describe the change in topography of the receptive field centres between the fovea and periphery. However by forgoing geometric regularity in the retinal structure, a self-organisation methodology (Clippingdale and Wilson, 1996) has enabled us to build retinal tessellations that instead exhibit continuity in sampling density. The final structure of such a retina locally resembles a pseudo-regular hexagonal lattice with slight deviations in the hexagonal topology in some locations.

4 Generating a self-organised tessellation

The retinal tessellations in Section 5 were created using Clippingdale and Wilson's (1996) 'self-similar neural network' model. The tessellations derived using this model have uniform foveal regions which seamlessly merge into the space-variant periphery of the retina (figures 6 and 7). The 2-dimensional coordinates of the receptive field centres of the tessellations are represented in the network as X_i network weights which are stimulated by an input stimulus which is derived by applying a random transformation to the network weights themselves.

Therefore, for a network of N units, each characterised by a 2 dimensional weight vector, the input stimulus $y_i(n)$ at iteration n is calculated by the following,

$$y_i(n) = T(n) x_i(n-1)$$

where $x_i(n-1)$ is the i th network unit at iteration $n-1$ and $1 \leq i \leq N$. In our work we used the following T composite transformation:

- (i) A random rotation between $[0, 2\pi)$
- (ii) A dilation (increase in eccentricity) comprising of the exponent of a dilation factor which is random between $[0, \log(8))$
- (iii) Translations in the vertical, horizontal and radial (away from centre) directions random between $[0, f)$, where f is associated with the required foveal percentage of the resultant retina.

Any input stimuli $y_i(n)$ which lie outside the bounds of the retina were culled before the network weights $x_i(n-1)$ were stimulated to calculate $x_i(n)$.

The network was initialised with a random weight configuration and iterated with the above transformations and the following learning rule:

$$x_j(n) = x_j(n-1) + \alpha(n) \sum_{i \in \Lambda_j(n)} (y_i(n) - x_j(n-1))$$

where,

$$\Lambda_j(n) = \left\{ i : \begin{aligned} &\|y_i(n) - x_j(n-1)\| \\ &< \|y_i(n) - x_k(n-1)\|, k \neq j \end{aligned} \right\}$$

$\Lambda_j(n)$ contains the indices to the input stimuli $y_i(n)$ to which $x_j(n-1)$ is the closest network vector. $\alpha(n)$ is a learning parameter which controls the stimulation of the network weights. We linearly reduced (annealed) α from 0.1 to 0.001 throughout the self-organisation to increase convergence.

Intuitively, one can visualise each network unit being updated individually by the input

stimuli that are closer to that unit than any other unit in the network. This training methodology causes the network to evolve a regular tessellation which is governed by the composite transformations T .

The retinæ that evolved using self-similar neural networks tend to have a regular hexagonal packing tessellation or mosaic.

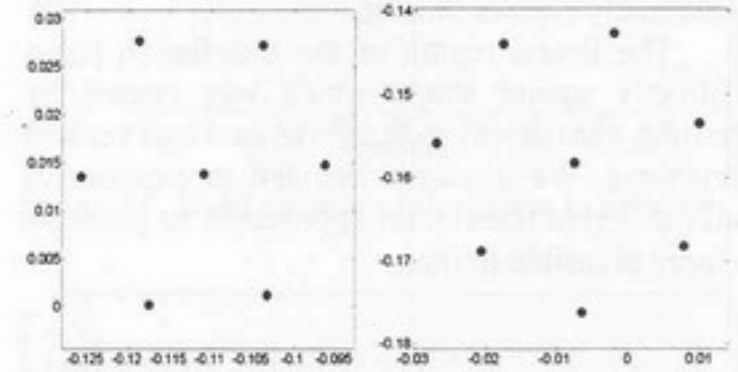


Figure 5: Magnified areas of a self-organised retina that show different packing mosaics.

This is not surprising as a hexagonal tessellation is the approximate pattern in which receptive fields are found in biological retinæ (Polyak, 1941). In fact this is a pattern commonly found throughout nature (Morgan, 1999). Receptive fields placed on this tessellation would then be equidistant from their immediate neighbours and Dudgeon and Mersereau (1984) showed that such a hexagonal tessellation is the optimal sampling scheme for a 2D space.

Close inspection of the tessellation (figure 5) shows that at some locations in the retina the tessellation undergoes an intermediate transition phase. In these locations the packing mosaic deviates slightly from a hexagonal cell construction (figure 5, right). These deviations occur relatively frequently in the region between the fovea and periphery of the retina where the retinal tessellation is changing from a uniform to a space-variant packing while maintaining regularity of the node mosaic.

Further details about self-similar neural networks and a discussion of their convergence properties are given by Clippingdale and Wilson (1996).

5 Results of the self-organisation process

Figure 6 contains a retinal mosaic that was created using self-similar neural networks with translations in the horizontal and vertical

directions up to 20% ($f = 0.2$) of the radius of the retina (refer section 4). As in Clippingdale and Wilson (1996) no radial translations were used to create input stimuli. The tessellation comprises of 4096 network nodes and was self-organised for 250000 iterations. This tessellation was self-organised for an extremely long period in comparison to our other experiments and the high iteration count let the network converge to a remarkably regular mosaic.

The foveal region of the tessellation has a distinctly square shape which was caused by limiting translations to the horizontal and vertical directions. We therefore decided to experiment with different translation approaches to generate a more plausible retina.

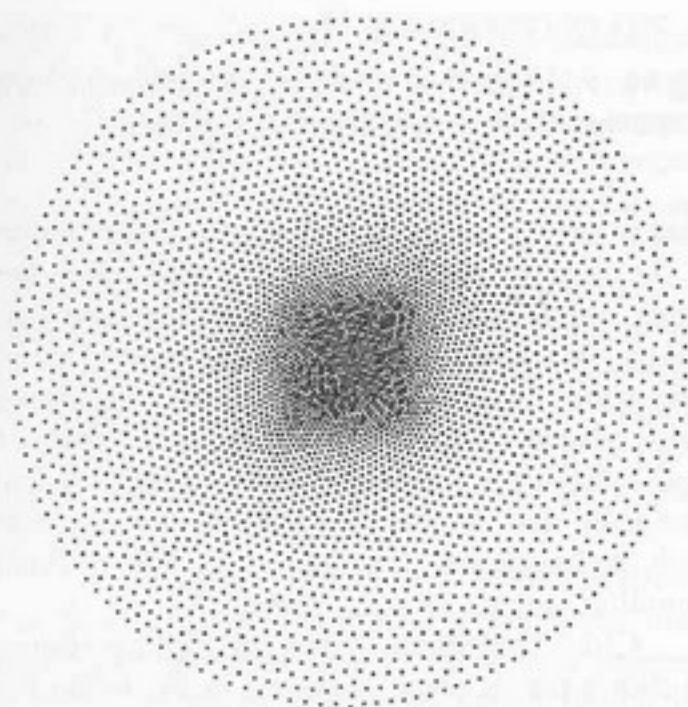


Figure 6: A plot of a self-organised retina with translations made in horizontal and vertical directions.

When we translated a 4096 node network radially instead of in the horizontal and vertical directions we obtained the pattern in figure 7. The radial translation was up to 20% ($f = 0.2$) of the radius of the retina (section 4) and the network was self-organised for 20000 iterations.

This approach resulted in a mosaic with a higher packing density in the fovea. However the centre of the fovea is approximately uniform and does not reach singularity (figure 8).

We were able to create a tessellation with a large uniform foveal region which was isometric by self-organising a network with a composite of horizontal, vertical and radial translations.

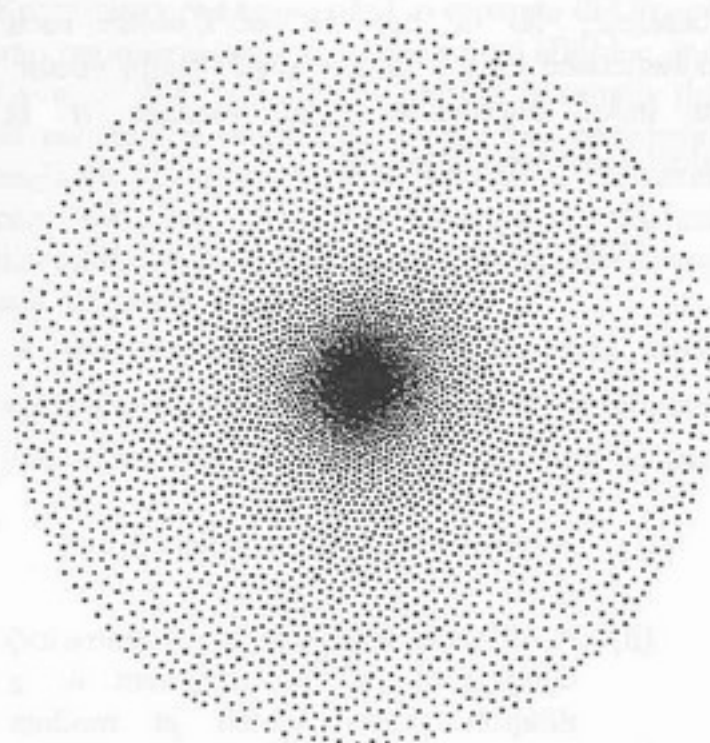


Figure 7: A plot of a self-organised retina with translations made radially.

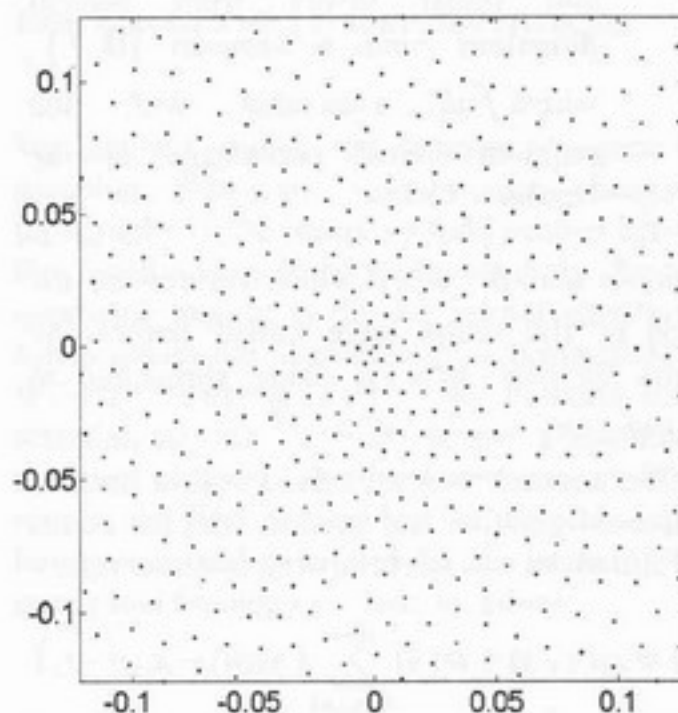


Figure 8: A magnified view of the fovea from the retina illustrated in figure 7 (the radius of the retina is 1 unit).

Figure 9 contains the mosaic that was obtained after self-organising with a radial translation that was up to 20% ($f = 0.2$) and horizontal and vertical translations which were up to 6.6% ($f = 0.066$) of the radius of the retina. The network consists of 4096 nodes and was annealed for 20000 iterations.

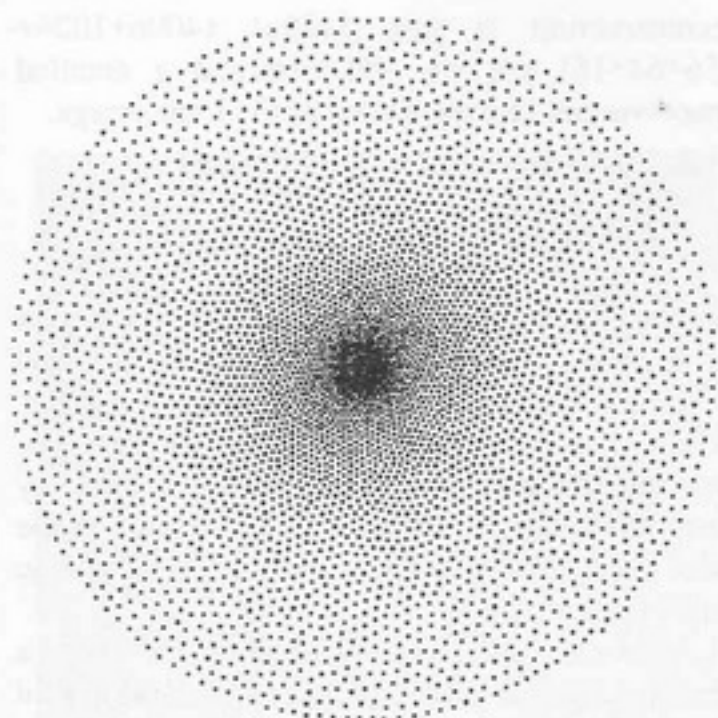


Figure 9: A plot of a self-organised retina with translations made horizontally, vertically and radially.

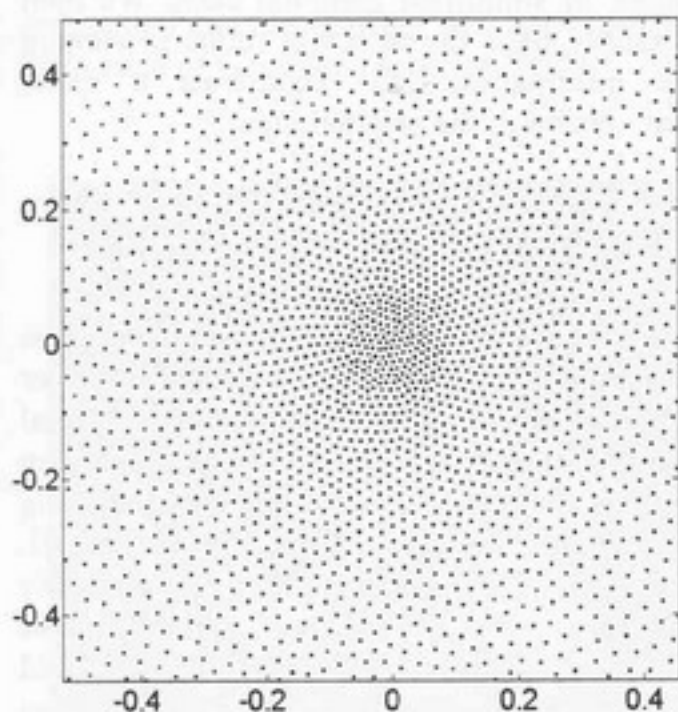


Figure 10: A magnified view of the fovea from the retina illustrated in figure 9 (the radius of the complete retina is 1 unit).

Figure 11 contains a plot of the node density of the retina in Figure 9 with eccentricity. The nodes were binned according to eccentricity into 50 bins and the resulting histogram frequency was calculated. The plot reflects the topography of the retina with a node density reaching a plateau in the fovea.

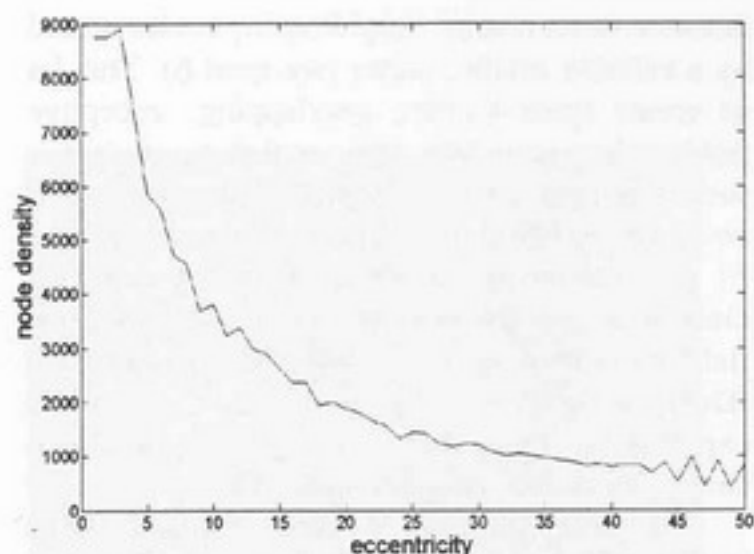


Figure 11: Node density of the retina in figure 9.

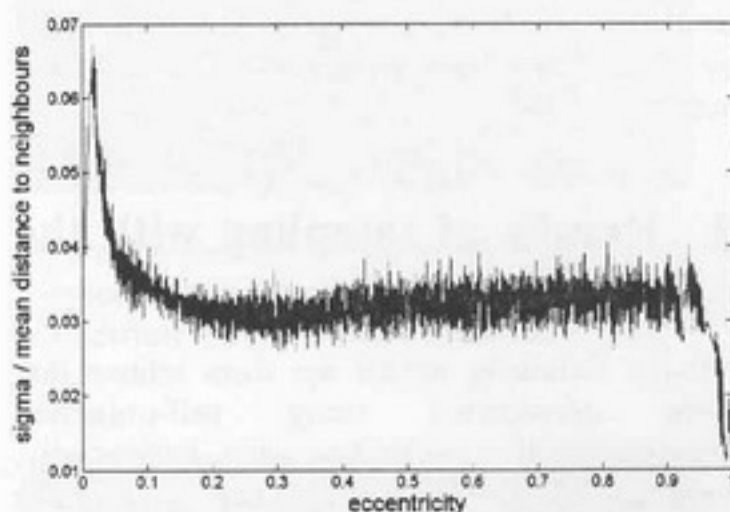


Figure 12: Standard deviation of the distance to a node's immediate neighbours in the retina in figure 9.

The standard deviation (figure 12) of the retina increases in the foveal region and the authors' hypothesise that this is due to the tessellation trying to maintain regularity while being subject to increased stimulatory forces in the fovea.

Further experiments revealed that the node density and variance values of retinæ exhibit less variability as one increases the number of iterations during self-organisation.

6 Placing receptive fields on the self-organised tessellation

We constructed retinæ out of the self-organised tessellations by placing overlapping receptive fields (or filters) at the calculated node locations. The receptive fields were placed on the calculated node locations with sub-pixel accuracy by altering the coefficients of the filters to reflect the sub-pixel jitter. The size of a receptive field was calculated based on the mean

distance to its nearest neighbouring nodes scaled by a suitable scaling factor (we used 6). This let us create space-variant, overlapping receptive fields. The retina was then scaled to an image such that the mean distance between closest nodes in the fovea was equal to one pixel. Figure 17 is a reconstruction of a cross stimulus that clearly illustrates the space-variant receptive field sizes in the retina. Difference of Gaussian (DoG) receptive fields were used in these experiments to resemble the processing of a simplified retinal ganglion cell layer.

An approximation to a space-variant octave Laplacian pyramid was constructed by sampling an image with retinæ constructed from several self-organised tessellations (figure 16). The number of nodes in the tessellations was reduced by a factor of 4 between resolution levels of the octave pyramid.

7 Results of sampling with the retina

All the following results are from retinæ that were constructed using self-organised tessellations that were translated horizontally, vertically and radially (figure 9). The reconstruction images are reversals of the sampling operations of the retina on an image. The low dimensional extracted responses from a retinal sampling (for example a 4096x1 dimensional vector in the case of a 4096 node retina) were distributed over the respective receptive fields of the retina to form the reconstruction.

Figure 15 contains the reconstruction of the sampling of the greyscale Lena image by a 4096 node retina with difference of Gaussian receptive fields. Since the receptive fields near the point of fixation are small, these have sampled and reconstructed higher spatial frequencies than in the periphery where there is reduced acuity in the reconstruction. The retina has a restricted field of view and therefore has not sampled the entire Lena image. This was because we scaled the retina to the image as described in section 6.

Figures 13 and 14 contain the reconstructions from retinæ with 64 and 256 nodes respectively. We similarly constructed retinæ with 16 and 1024 nodes and used all these retinæ to construct a superposition of the reconstructions of the Lena image which were sampled by retinæ with 4096, 1024, 256, 64 and 16 difference of Gaussian receptive fields (figure 16). The vector that was used for the

reconstruction is just 5456x1 (4096+1024+256+64+16) yet we still obtained a detailed space-variant representation of the Lena image.

8 Conclusion

We have created a plausible artificial retina with a regular receptive field tessellation and continuity between foveal and peripheral regions. While the tessellation was locally pseudo-random it maintained a sampling density continuum at a macroscopic level, though at the price of sacrificing (analytic) geometric uniformity.

Despite the pseudo-random nature of the tessellation, it is still possible to conduct useful computation on this visual representation. To demonstrate this, space-variant difference of Gaussian receptive fields were placed on the generated retinal tessellation to simulate 5 layers, i.e. 5 separate spatial frequency bandpass channels, of simplified ganglion cells. We then successfully used the resulting retina to sample image information and reconstruct a space variant version of the original input.

9 Future Work

The visual information extracted using this retinal front end will need to undergo further processing in subsequent stages of a visual pathway. However performing image processing operations on visual information extracted using an irregular sampling regimen is not trivial. Wallace et. al. have proposed connectivity graphs for reasoning with space variant images (Wallace et. al., 1995). Since the retina we used was not based on an analytical transform such as the $\log(z)$ type mappings in Wallace et. al. (1995), we recently adopted Delaunay triangulation (Barber et. al. 1996) to define a 'topological cortical space' which we were able to further transform by applying image filtering operations.

To perform a convolution with any arbitrary filter we calculated the coefficients of the filters at topological cortical space node locations. We have already been able to filter oriented image features from the responses extracted from the retina by placing Gabor receptive fields on the topological cortical space and hope to demonstrate the uniform coding of visual information across the entire visual field when using our representation.

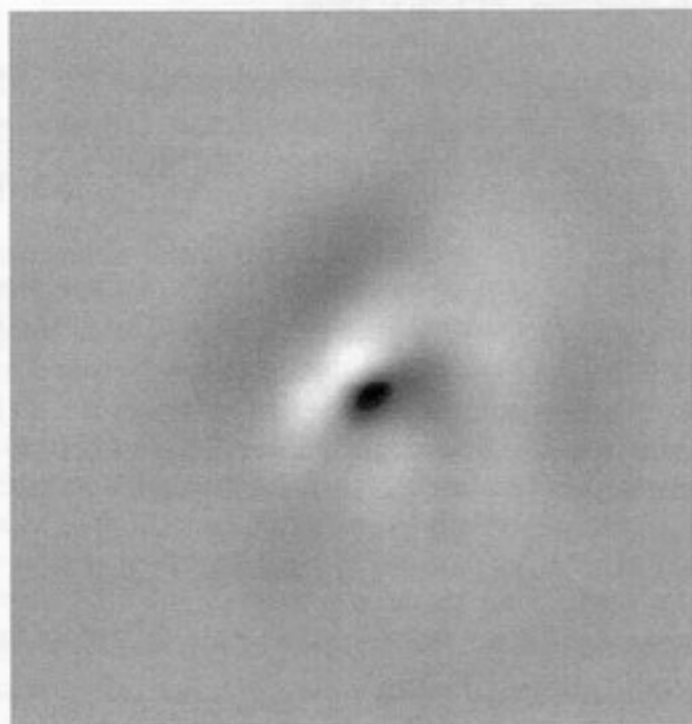


Figure 13: Reconstruction of the sampling of the greyscale Lena image using a 64 node retina with difference of Gaussian receptive fields.

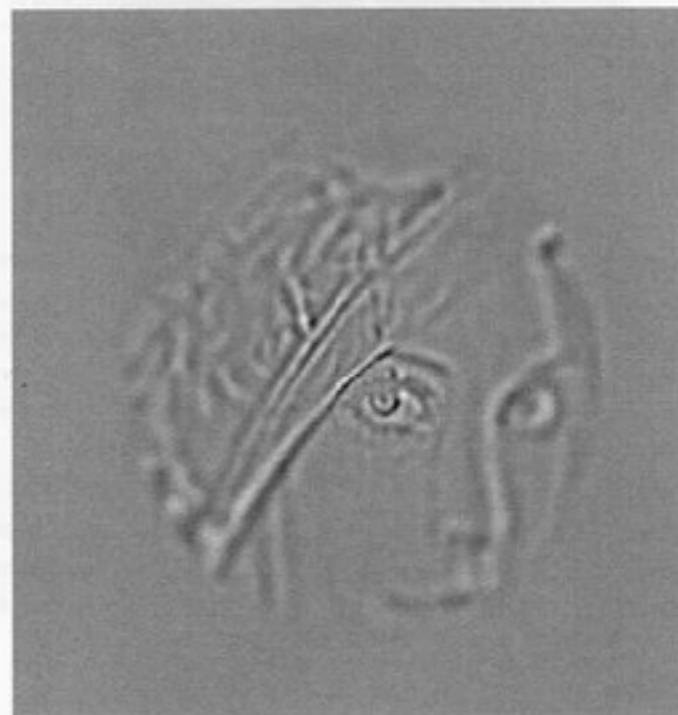


Figure 15: Reconstruction of the sampling of the greyscale Lena image using a 4096 node retina with difference of Gaussian receptive fields.

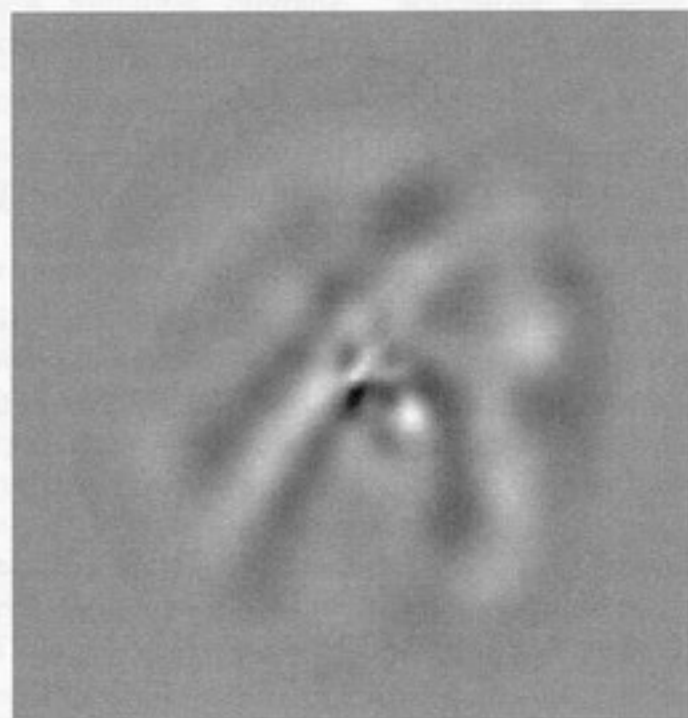


Figure 14: Reconstruction of the sampling of the greyscale Lena image using a 256 node retina with difference of Gaussian receptive fields.



Figure 16: Reconstruction of the sampling of the greyscale Lena image by the superposition of the image pyramid individual resolution layers

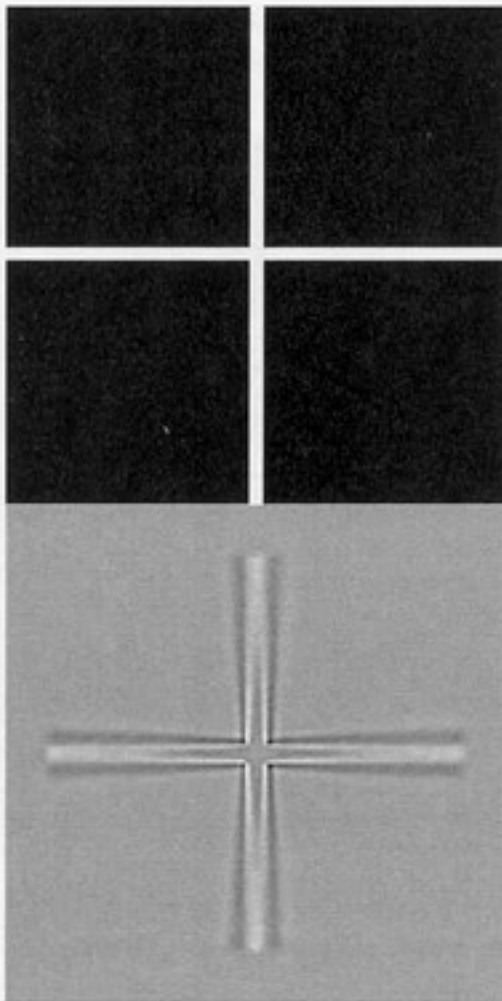


Figure 17: Input cross stimulus (top) and the reconstruction of the cross stimulus using a 4096 node retina with difference of Gaussian receptive fields (bottom).

Acknowledgements

The authors wish to acknowledge the support of the Department of Computer Science, University of Glasgow and the UK Imaging Faraday Partnership.

References

Barber, C. B., Dobkin, D. P. and Huhdanpaa H. T., The Quickhull Algorithm for Convex Hulls, *ACM Transactions on Mathematical Software*, Vol. 22, No. 4, Dec. 1996

Bolduc, M. and Levine, M. A review of biologically motivated space-variant data reduction models for robot vision. *Computer Vision and Image Understanding*, 69, 1998

Burt, P. J. and Adelson, E. H. The Laplacian pyramid as a compact image code, *IEEE Transactions on Communications*, pp. 532--540, April 1983

Clippingdale, S. and Wilson, R. Self-similar Neural Networks Based on a Kohonen Learning Rule. *Neural Networks*. Volume 9, Issue 5, pp. 747-763. 1996

Dudgeon, D. E. and Mersereau, R. M. Multidimensional Digital Signal Processing. *Prentice-Hall, Inc.*, Englewood-Cliffs, NJ, 1st edition. 1984

Hubel D. H. and Wiesel T. N. Brain mechanisms of vision. *Scientific American*. 241:150-162, 1979.

Kohonen, T. Self-organization and associative memory. *Berlin: Springer-Verlag*. 1984

Marr, D. Vision. *W. H. Freeman and Co*. 1982

Morgan, F. The hexagonal honeycomb conjecture. *Transactions of the American Mathematical Society* 351(May):1753. 1999

Polyak, S. L. The Retina. *University of Chicago Press*, Chicago. 1941

Schwartz, E. L. Spatial mapping in primate sensory projection: Analytic structure and relevance to perception, *Biological Cybernetics*, 25:181-194. 1977

Schwartz, E. L. Computational Anatomy and functional architecture of the striate cortex, *Vision Research*, vol 20, pp. 645-669. 1980

Smeraldi, F., Makarov, A. and Bigun, J. Saccadic search with Gabor features applied to eye detection. Technical Report EPFL-DI-98/256, *Swiss Federal Institute of technology*, CH-1015 Lausanne, Jan 1998.

Thorpe, S. and Gautrais, J. Rapid Visual Processing using Spike Asynchrony. NIPS, Jordan M. editor, MIT press, pp 901-907. 1997

von der Malsburg, C. The correlation theory of brain function. Internal Report 81-2, *MPI Biophysical Chemistry*, Göttingen, Germany. 1981

Wallace, R., Ong, P., Bederson, B. and Schwartz, E. Space variant image processing. *International Journal of Computer Vision*, 13(1):71--90, 1995.

Wilson, S. W. On the retino-cortical mapping. *International Journal of Man-Machine Studies*, 18(4):361-389. 1983

Iterative Orientation Tuning of Simple Cells in V1: A Comparative Study of Two Computational Models for Contrast Detection in Images.

Marina Kolesnik¹, Alexander Barlit

Fraunhofer Institute for Media Communication, Schloss Birlinghoven, D-53754 Sankt-Augustin, Germany

Abstract

The orientation selectivity of simple cells in visual cortex gives a striking example of the biological system perfectly adapted to the perception of oriented stimuli. Several models have employed major principles of orientation selectivity for the processing of contrast variations in images. We have recently suggested a model for iterative orientation tuning, in which the astonishingly regular layout of simple cells is explicitly involved in the processing of oriented stimuli. In this work we extended the iterative model by incorporating a mechanism of cross-oriented inhibition. We then investigated the two models using synthetic, noisy and natural images. We found that the two models account for a large fraction of the contrast invariance of orientation selectivity – another striking aspect of the behaviour of simple cells. Our results indicate that the iterative processing of visual stimuli combined with local amplification of proximate simple cells is responsible for ~75% of the contrast invariance. Contrary to some earlier studies, the cross-oriented inhibition did not have any significant contribution to the contrast invariance but accelerated the convergence of the iterative processing on a stable solution. When probed with different images, the new model with cross-oriented inhibition generated a clear pattern of object contours.

1 Introduction

Edge detection is a cornerstone processing stage in the analysis of visual information by humans. This has triggered the development of numerous algorithms for detection of local luminance changes in images. The most popular edge detectors include Marr-Hildreth zero-crossings [1], Canny [2], Haralick [3], Deriche [4] approaches, and a full list of suggested algorithms would go on. From the sheer number of suggested edge detectors, one may conclude that the detection of contours of objects in images is not an easy task for a computer. The fact that a person can effortlessly find the contours of objects has inspired the investigation, and modelling of a contrast detection circuitry in mammalian visual system. This work attempts at developing a processing algorithm for contrast detection in images built upon principles of the physiological orientation selectivity in mammals.

Forty years ago, Hubel and Wiesel ([5], 1962) discovered that *simple cells* in cat primary visual cortex (V1) are tuned for the orientation of light/dark borders. The inputs to V1 come from the lateral geniculate nucleus (LGN), whose cells are not significantly orientation selective [6]. LGN cells themselves get their input from the retinal ON and OFF *ganglion cells* with centre-surround receptive fields (RFs), first discovered by Kuffler ([7], 1953). The orientation selectivity of simple cells in V1, as proposed by Hubel and Wiesel [5], derives from an oriented arrangement of input from the lateral geniculate nucleus (LGN): ON-centre LGN inputs have receptive fields (RFs) centres aligned over simple cell's ON subregions, and similarly for OFF-centre inputs. Because of this input arrangement, simple

cells perform a linear spatial summation of light intensity in their fields and have an elongated shape of their RFs. The orientation preference of simple cells is quite narrow, and turning a bar-stimulus through more than about 20° from the preferred orientation, greatly reduces the cell's firing rate.

A traditional *feed-forward model* of the orientation selectivity performs linear spatial summation of input signals from the LGN followed by a non-linear rectification stage, in which a threshold filters out small inputs evoked by improperly oriented stimuli [8], [9]. Although many aspects of simple cell responses are consistent with this linear model, there also are important violations of linearity. For example, scaling the contrast of a stimulus would identically scale the responses of a linear cell. At high contrasts, however, the responses of simple cells show clear saturation. Such behaviour of the simple cells is referred to as *contrast invariance* of orientation selectivity [10].

Several neuronal models have attempted to address the nonlinearities of simple cell responses by extending the linear model to include a gain control stage [11], [12], [13]. It is suggested the response of a simple cell is governed by *shunting inhibition* – the divisive normalization of the cell activity due to interaction with other cells. The shunting inhibition controls the gain of the transformation of the cell's input current to output membrane potential [14]. A followed rectification stage converts the latter into a firing rate of the cell. The models with shunting inhibition predict response saturation because the divisive normalization increases with stimulus contrast. Another class of models, which also exhibit the contrast invariance rely on the amplification of LGN input by recurrent excitation

¹ To whom correspondence should be addressed: marina.kolesnik@imk.fraunhofer.de

occurring within the cortical column [29], [30], [31], [32]. This amplification is gated selectively by intracortical inhibition and thereby sharpens weak and poorly oriented LGN input. To arrive at these results the models make an assumption that LGN synapses comprise 5%-10% of the total excitatory synapses present in layer 4. However, recordings of visually evoked membrane potential changes in simple cells [33] indicate that the LGN input is responsible for generating approximately 35% to 46% responses of simple cells.

Despite remaining controversy over the details of synaptic mechanisms underlying orientation selectivity, advances in understanding of major principles of its functionality, laid the foundation for the development of computational models for contrast detection in images [15], [16], [17]. The typical architecture of such a model is built upon a simple cell circuit, which is composed of segregated ON- and OFF-data streams interacting via mechanism of opponent inhibition as suggested by Ferster [18].

None of these models, however, explicitly employed the impressive regularity in a spatial layout of simple cells called by Hubel and Wiesel [5], the *functional organisation* of visual cortex. Visual cortex has a distinctive striped appearance in cross-section, caused by the arrangement of cells in layers of different densities and for this reason it is also known as the *striate* cortex. Simple cells that respond more strongly to stimuli in one eye than in the other, and are said to show *ocular dominance*, are aligned into *ocular dominance stripes*. Moreover, when the orientation preference of cells in the ocular dominance stripes was related to their position, an astonishingly systematic organisation has emerged: the orientation preference changed linearly with position across V1 [19], [20]. After some distance where cells had shown a systematic clockwise stepping of their orientation preferences, the sequence would reverse to anticlockwise. Hubel and Wiesel therefore suggested that orientation-selective cells are organised in columns or "slabs", in which all cells have the same preferred orientation, and that adjacent slabs represent adjacent orientation. Because, furthermore, the orientation slabs tended to be at right angles to ocular dominance stripes, their regular structure was nicknamed as *Icecube* layout.

Since axons and dendrites take up a significant fraction (about 60%) of the cortical volume [21], limitations on the brain size require keeping neuronal processes as short as possible. Evolution was likely to select for developmental rules that produce orientation preference maps which are sufficiently optimised in terms of length of neuronal connections. Numerical simulations relating orientation preference maps to the length of intracortical wiring have shown that the optimised layout is the *Icecube* if the strength of local connections is Gaussian [22]. Therefore we assume that local interactions of spatially close simple cells within the *Icecube* ought to be important for their functionality. A first attempt at utilizing the regularity of orientation preference maps for contrast detection in images, has

been made in [23]. It is suggested that the processing of video input undergoes several iterative cycles. The responses of simple cells at different iterations change due to local interactions of proximate cells. The model takes advantage of the regular layout of orientation preference in a very explicit way: each simple cell is sending activation into a regular net of local connections and amplifies the activity of spatially close cells. The model achieves a significant level of contrast invariance of orientation selectivity due to the iterative amplification of cells of similar orientations at retinotopically close positions.

The work here discusses an extension of the iterative model [23] by incorporating a mechanism of orthogonal suppression of spatially close simple cells of V1. The new model accounts for another aspect of the behaviour of simple cells, namely that simple cells are subject to *cross-oriented inhibition*; the responses to an optimally oriented stimulus can be diminished by superimposing an orthogonal stimulus that is ineffective in driving the cell when presented alone [24], [25]. We suggest that a highly systematic wiring of simple cells in neighbouring ocular dominance bands is involved in the transition of inhibiting signals to nearby cells of orthogonal orientation.

We test the performance of both models using a selected set of two-dimensional stimuli as well as noisy and natural images. Comparison of processing results reveals a very similar behavioural pattern. Both models account for a large fraction of the contrast invariance of orientation selectivity. Our results indicate that incorporation of the cross-oriented inhibition does not significantly improve its performance in terms of contrast invariance, rather stabilises the model and accelerates its convergence on equilibrium. The new model is robust to noise and, when probed with natural images, it generates a clear pattern of contours.

2 Model 1: the iterative orientation tuning of simple cells

The first model 1 has been introduced in [23] and is built upon the idea that visual perception is a continuous process of interpretation of incoming visual data. We adopt the view that the brain has no internal representation of the outside world because it is continuously available "out there" for active perception. While the eye is fixating a particular object, the low-level processing of constant visual input may be undergoing several iterative cycles. Every moment when light hits retina it would cause a different neural activity in the underlying visual circuitry, the activity which depends on a level of current neuronal excitation. Consequently, neural responses to the same visual input at different processing cycles would vary.

This iterative approach to the low-level visual processing gains a further meaning when the role of feedback connections is considered. It is logical to suggest that feedback projections, activated at subsequent iterations, would alter cell responses to the

visual input, which itself remains constant in time. It is even more likely that the regular layout of simple cells in V1 reinforces responses of simple cells at subsequent iterations. In the model, we assume that the activation of a simple cell amplifies the activity level of proximate cells, so as to tune these neighbouring cells to a local orientation pattern. After several cycles of iterative tuning, the whole system reaches equilibrium and responses of simple cells to visual input stabilise.

A neural circuit of the model for iterative orientation tuning (Fig. 1) consists of two ON- and OFF-pathways interacting via a mechanism of opponent inhibition. Visual input is processed sequentially, first by retina-LGN followed by a simple cell circuit in V1. A key feature of the model is the iterative processing of visual input, which imitates an instance when the eye is fixating a particular object and the processing of still video input might undergo several iterative cycles. Local intracortical interaction of simple cells changes their responses to the visual input at subsequent iterations. The local interaction of simple cells is governed by their spatial layout. We adopt an *Icccube* model to describe the spatial layout of simple cells. In the process of local interaction the activation of each simple cell causes excitation of close cells in the *Icccube* layout.

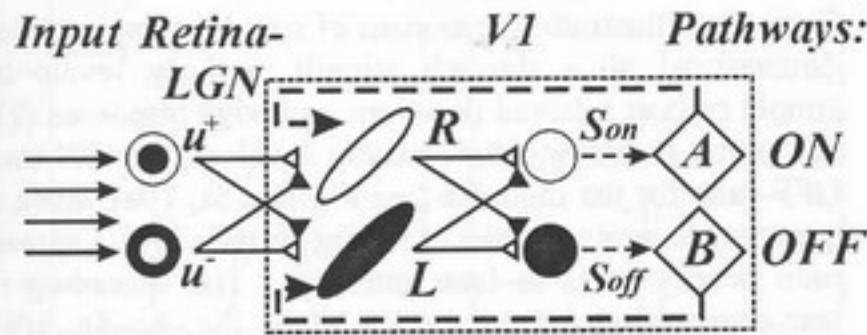


Figure 1. The architecture of the model neural network. The network consists of two major stages - the retina-LGN stage, followed by a simple cell circuit. Triangles at the end of lines denote the excitatory input; filled-in-black triangles denote the inhibitory input. Two dashed lines show the iterative interaction of spatially close simple cells.

The first processing stage deals with responses of retinal ganglion cells with centre-surround receptive fields (RFs) [7]. The retinal ganglion cells are modelled at each spatial position by the difference of input stimulus and its convolution with a 2-dimensional Gaussian kernel (A1), [27]. Retinal ON and OFF ganglion cells synapse mainly onto respective ON and OFF cells of the LGN. In the model, retinal inputs do not change while passing through the LGN.

Simple cells of V1 are driven by oriented input from the LGN. Physiological studies on simple cell responses recorded in cat striate cortex suggest that elongated sensitivity profile of a simple cell subfield is best modelled by a difference of two elongated Gaussians (A2). Each simple ON cell receives excitatory input from the LGN ON cells beneath it and is inhibited by LGN OFF cells at the same retinotopic position (A3).

In addition, simple cells undergo local interaction, which amplifies the activity of cells belonging to a same channel. All simple cells are considered to be stacked

into a 3-D array (*Icccube* layout, Fig. 2), in which two coordinates define the spatial (retinotopic) position of the cell and the remaining third coordinate is related to the cell's preferred orientation. Each simple ON cell undergoes additive amplification received from those simple ON cells that are spatially close in the 3-D array; the same ON cell is inhibited by proximate simple OFF cells (A3). The reverse arrangement holds true for simple OFF cells.

Final activation of a simple cell results from the cross-channel inhibition obtained as the steady-state solution of inhibitory shunting interaction (A4), [28].

Because the strength of amplification is an inverse function of squared distance (A6), the activation of proximate cells effectively decays within the distance of 3 units. Due to the weighting factor $\omega = 16$ in (A6), the effect of amplification affects only one neighbouring cell in all 8 directions within the spatial layer, and about 6 neighbouring cells in the orientation column (3 orientations both up and down the column). This local amplification enhances responses of both retinotopically proximate cells and cells tuned to similar orientations.

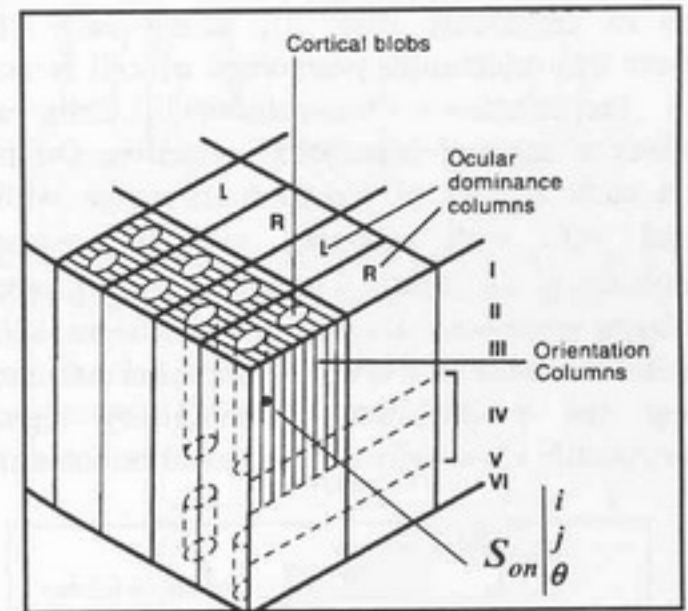


Figure 2. A schematic diagram of the spatial layout of V1 - *Icccube* layout. The primary visual cortex is divided into ocular dominance columns; running perpendicular to these are orientation columns. Orientation preference within columns changes systematically so that each column represents directions from 0° to 180° . The *Icccube* layout of simple cells is modelled by the 3-D array consisting of spatial layers and orientation columns. Each element of the array, (i, j, θ) has two spatial coordinates, i and j , for position within the layer and one orientation coordinate, θ , for position in the orientation column. It is assumed that each 3-D position contains a pair of ON and OFF cells, S_{on} and S_{off} . This array layout is repeated twice for both contrast polarities $p = 1, 2$.

The local amplification changes responses of simple cells to the same visual input over time. This is mediated by the iterative processing of visual input: amplification functions (A6) for the ON and OFF cells are fed into (A3) and the processing cycle is repeated. As the model proceeds through iterations, responses of simple cells would increase. It is however important that the model reaches equilibrium and the amplification of proximate cells stabilises. The corresponding balancing mechanism

is provided by the cross-channel interaction (A4), which does not let responses of simple cells to grow indefinitely. However, the cross-channel interaction alone cannot fully prevent a small growth of cell responses cells in the vicinity of sharp luminance changes.

3 Model 2: the iterative tuning with cross-orientation inhibition

Incorporation of the mechanism of cross-oriented inhibition into the model 2, is based on considerable experimental evidence suggesting that stimuli at non-optimal orientations suppress the background activity of simple cells [13], [24], [25]. Clearly, the cross-oriented inhibition has the potential to suppress weak responses in the vicinity of contrastive edges, thus increasing model's robustness to noise. Similar to the local amplification, the cross-oriented inhibition is governed by the spatial layout of simple cells in V1.

Optical imaging studies on patterns of activation across a region of monkey cortex [26] revealed a regular structure of "iso-orientation" contours radiating from points of singularity (Fig. 3). Along each of these contours the orientation preference of cell is constant, hence the name - "iso-contours". Cells at the singularities are not orientation selective. Orientations within each bundle of iso-contours range within the interval 90° with adjacent contours representing orientations 11.25° apart. A complete circle around each singularity represents a rotation from 0° to 180° . We suggest that these iso-oriented contours are the links serving the propagation of inhibitory signals to retinotopically close cells of orthogonal orientation.

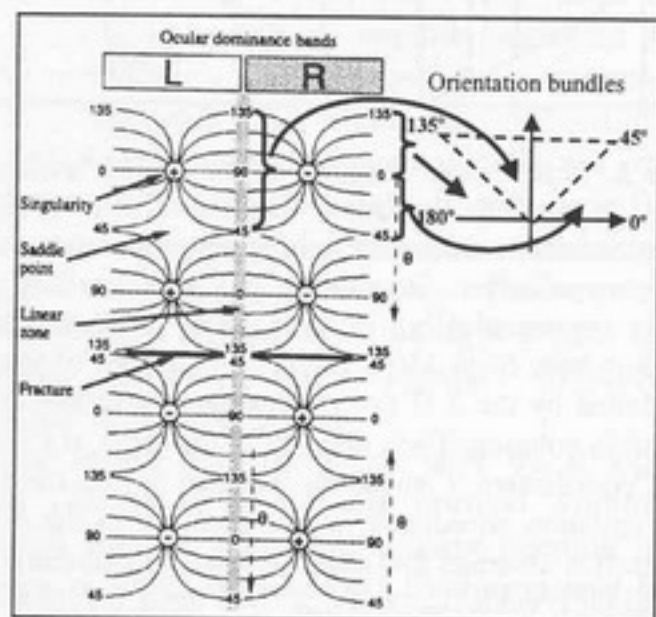


Figure 3. Schematic presentation of relationship between ocular dominance bands and the organisation of orientation selectivity in the visual cortex (Obermayer and Blasdel [26]). Cells along "iso-oriented contour" have the same optimal orientation. Adjacent iso-contour bundles linked at points of singularity range within two complementary sectors of 90° each (shown in black arrows). Due to this arrangement for each cell in a given bundle there exist a counterpart cell of orthogonal orientation belonging to the adjacent bundle.

The activity of each simple cell is inhibited by four cells from the neighbouring orientation columns that are

tuned to orthogonal orientation (A5). This mechanism of cross-orientation suppression affects the activity of simple cells in two ways. On one hand, the activity of a simple cell is cancelled out by the activity of retinotopically close cells of orthogonal orientation if these are strongly activated. On the other hand, the response of a strongly activated cell would only be slightly suppressed by retinotopically close cells of orthogonal orientation if these are weakly activated. This cross-oriented inhibition eliminates weak responses of simple cells while sharpening their strong responses.

4 Perception of luminance changes by the two models

We investigated the behaviour of the two models through a set of computer simulations. Each model is probed with several test stimuli, illustrating typical instances of contrast variations in images. All test stimuli are two-dimensional functions. Each model proceeds through 5 iterative cycles before edge responses of simple cells are generated. Edge response S is computed by rectifying the sum of activities of ON and OFF cells minus their difference:

$$S = [S_{on} + S_{off} - |S_{on} - S_{off}|]^+ \quad (1)$$

Each plot, illustrating responses of simple cells, is a one-dimensional slice through stimuli, activity levels of simple cells at selected iterations and edge responses (1). All plots, displaying the activity level of the ON and OFF cells for the model 1 (see Fig. 4a, 5a, 10a), share a common feature: regions of strong activity levels spread onto nearby areas at later iterations. The spreading is less pronounced for the model 2 (see Fig. 4c, 5c, 10b) because the cross-oriented suppression (A5) wipes out the low-level activity at tale regions of the Gaussian-shaped response of ON and OFF cells.

Ramp transition. We conducted two series of simulations investigating the strength of responses, depending on the transition range and width. On average, responses of the ON and OFF cells generated by the model 2 grow less with iterations than responses generated by the model 1 (Fig. 4 a, c; and Fig. 5 a, c). This behavioural difference occurs due to the orthogonal suppression. However, both models exhibit almost identical edge response (1) to the ramp transition regardless of its range (Fig. 4, b, d and Fig. 5 b, d). Our investigation shows that the perception of ramp profile by the two models depends largely on the profile's width and much less on the range of ramp transition. The strength of edge responses decreases approximately linearly with the increasing width of the ramp transition (Fig. 6). When the ramp width exceeds 28 pixels, responses of simple cells decay completely even though the transition range is high. This behaviour is similar for both models although the trend is quicker for the model 2: it becomes insensitive to the transition ramp wider than 22 pixels.

The dependency of responses on the ramp range is strongly non-linear for both models. Fig. 7 shows, that

the increase of the ramp range by a factor of 9 causes about 25% growth in the strength of respective edge response. This non-linear dependency is aggregated during the iterative processing due to advantageous enhancement of initially weaker responses. This result supports a great part of the contrast invariance of orientation selectivity observed experimentally.

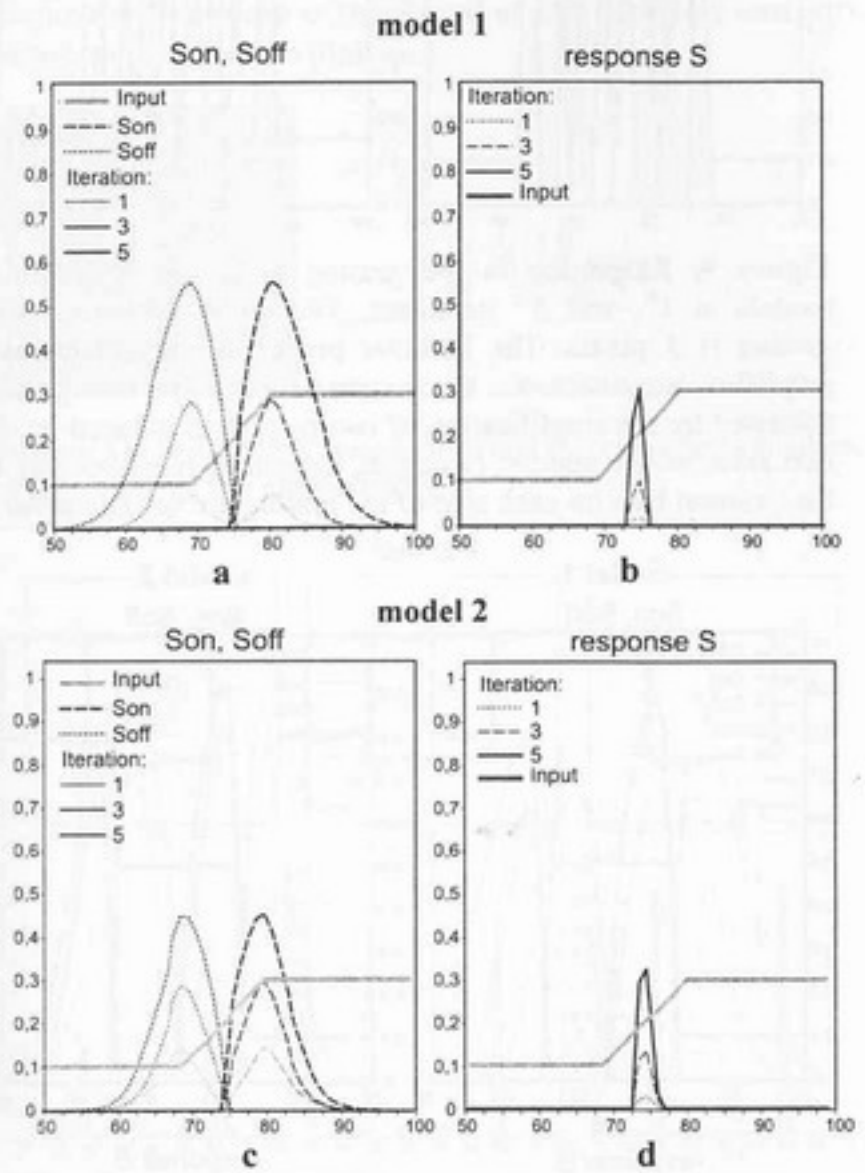


Figure 4. Responses to the ramp transition generated by the two models. Ramp range is equal to 0.2. The responses of ON and OFF cells, S_{on} and S_{off} , generated by the model 1 (a), grow stronger and spread wider with iterations as compared to the corresponding responses generated by the model 2 (c). Due to a steep fall off in the ON and OFF response for model 2 (c), it generates as strong edge response at 5th iteration (d) as that one of model 1 (b).

Further analysis of curves in Fig. 7 shows that a subtle difference in the behaviour of two models appears at later iterations. Edge responses generated by the model 2 stabilise at 6th iteration. The convergence rate is slower for the model 1. The explanation for this comes from the analysis of mechanisms stabilising the two models. There exist two such mechanisms. The first one, the cross-channel inhibition (A4), is common for both models. The second one, the cross-oriented suppression (A5) which is only present in the model 2, imposes an additional constraint on the propagation of excitation onto cells which do not receive salient oriented excitation from the visual input. Also, the cross-oriented suppression accelerates the convergence of the iterative processing of model 2 when compared to the model 1.

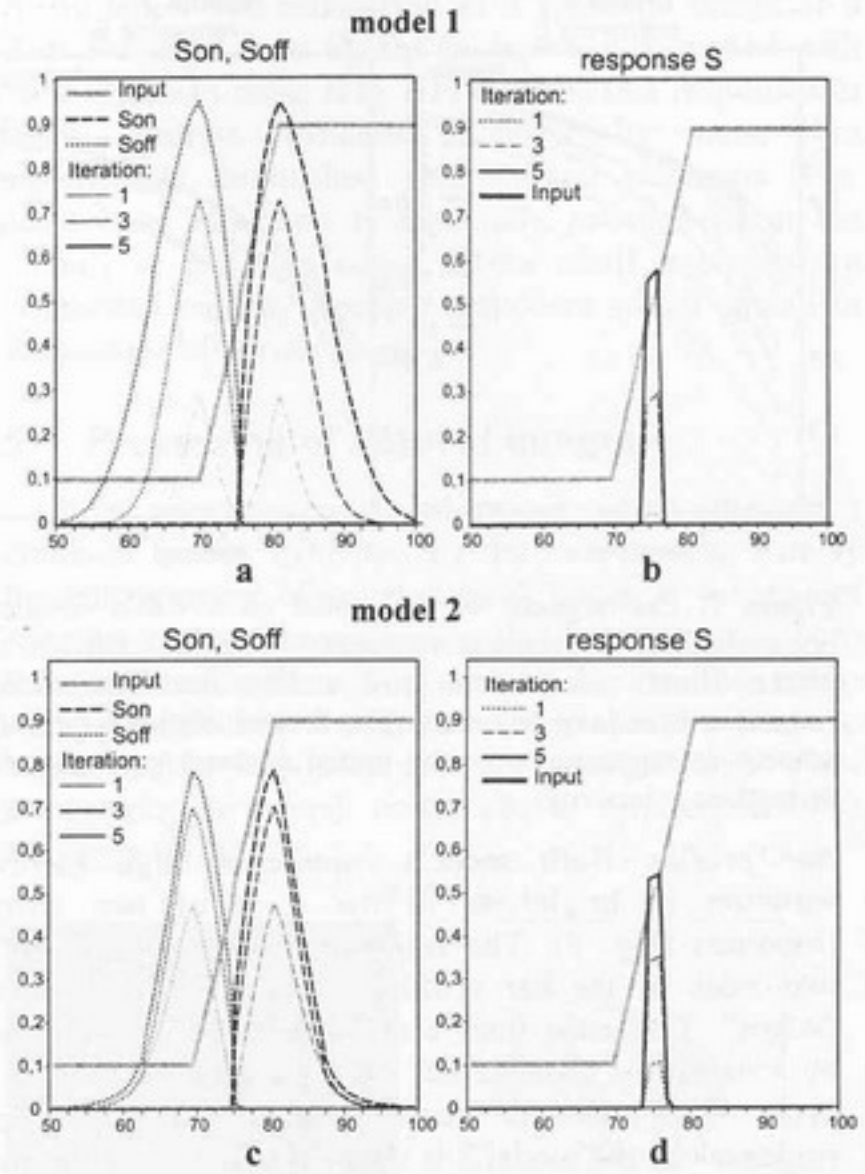


Figure 5. Responses to the ramp profile of range 0.8. Edge responses display similar tendencies in the shape and rate of growth as responses to the ramp transition of 0.2 (Fig. 4). Also the strength of edge response at 5th iteration of model 2 (d) is pretty the same as that one of model 1 (b).

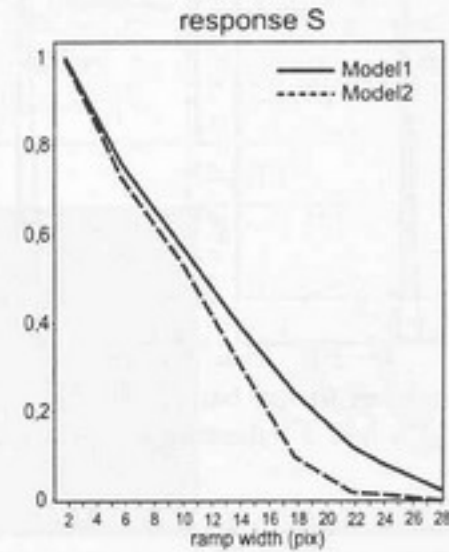


Figure 6. Edge response to ramp profiles of different widths. The strength of edge responses generated by the two models is plotted against the width of ramp transition with range 0.8. For both models edge response drops almost linearly with the increase of the ramp width. The fall of is quicker for the model 2, which does not perceive the luminance ramp above 22 pixels width.

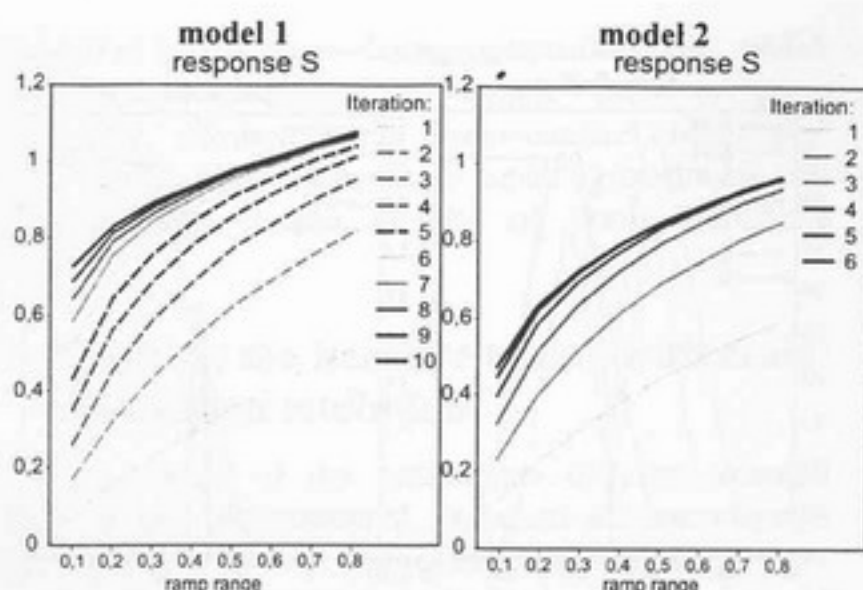


Figure 7. Convergence of the model on a stable solution occurs quicker for stronger responses. Whereas responses to weaker stimuli continue to grow at later iterations, strong responses do not rise any more. This feature, common for both models, is responsible for the partial contrast invariance of orientation selectivity.

Bar profile. Both models capture a large narrow variation in brightness in the form of two sharp responses (Fig. 8). The responses are associated with two sides of the bar profile, which are perceived as “edges”. The same double response pattern is induced by a narrowest possible bar profile with the width of 1 pixel. Note that the iterative amplification of edge responses by the model 2 is stronger than that one of the model 1.

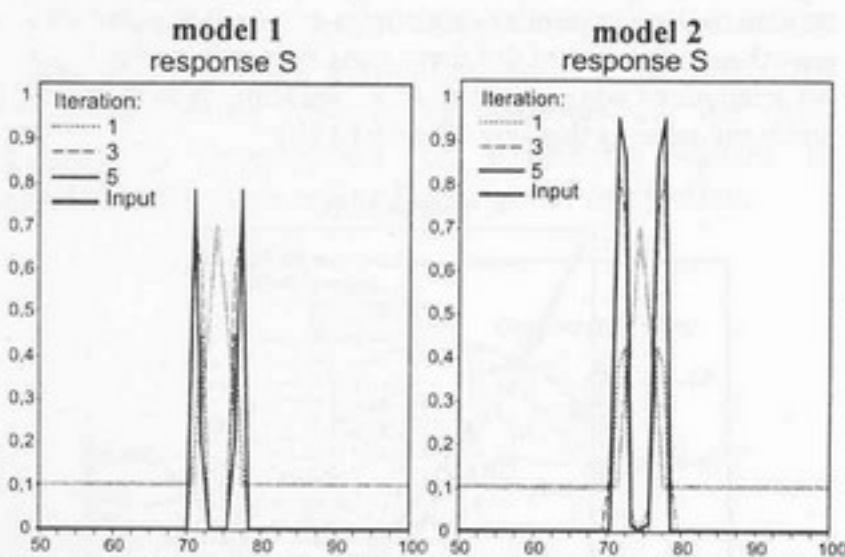


Figure 8. Responses to the bar profile generated by the two models at 1st, 3rd, and 5th iterations. Bar width is equal to 3 pixels.

Grating. Simulation of responses induced by a grating composed of four equal contrast bars produces eight sharp responses at “edge” positions, each one associated with particular bar side. However, boundary responses evoked by the two external sides of the grating undergo stronger amplification at iterations than responses to any of the internal sides of grating bars. It seems that more isolated responses tend to override nearby responses of smaller or comparable magnitude. This behavioural aspect is particularly useful for the processing of noisy images. Weak responses to spontaneous luminance variations caused by noise get eliminated after several iterations. The elimination process is especially efficient in the vicinity of strong luminance changes.

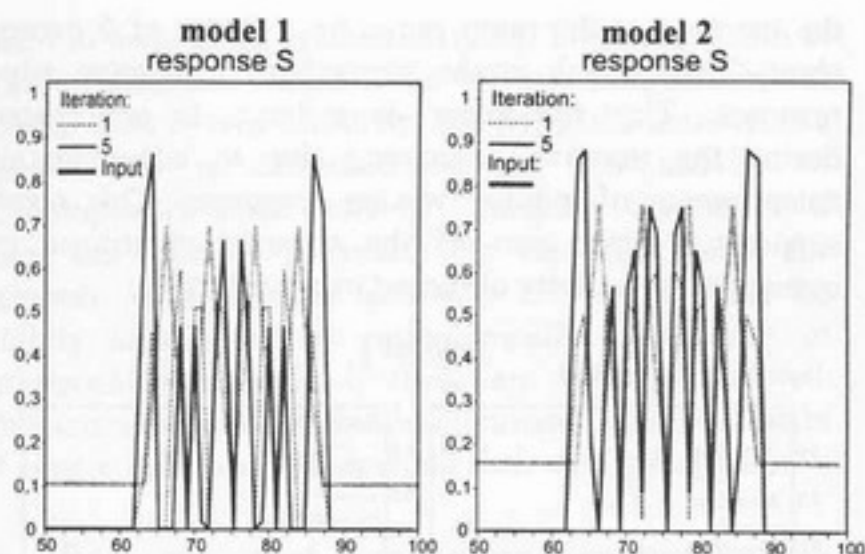


Figure 9. Responses to the grating generated by the two models at 1st, and 5th iterations. The width of bars in the grating is 3 pixels. The iterative processing advantageously amplifies responses to the external sides of the grating followed by the amplification of two responses induced by the two sides in the middle. Note that responses, neighbouring to the external bars on each side of the grating are not amplified.

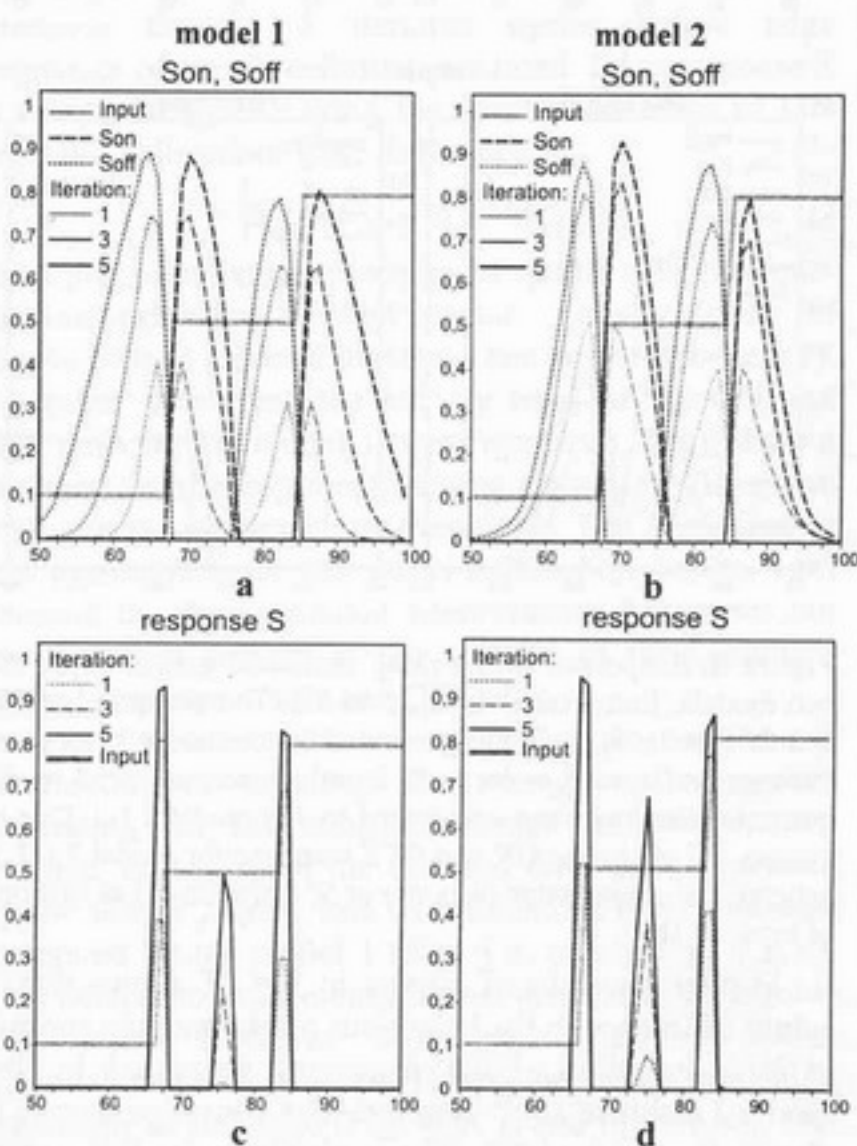


Figure 10. Responses to the staircase profile generated by the two models at 1st, and 5th iterations. The models perceive the illusory line in the middle of the step. Due to a sharper shape of the ON and OFF responses generated by model 2 (b), the perception of illusory line by the model 2 is stronger.

Staircase profile. The processing of a luminance staircase reveals that both models perceive an illusory line right in the middle of step’s plateau. The origin of this curious phenomena becomes clear when we analyse the shape of responses of the ON and OFF cells (Fig. 10, a, b). The OFF cell response to the abrupt luminance change between two subsequent steps has overlay with

the ON cell response evoked by the adjacent luminance change in the staircase. Computation of edge responses (1) results into a sharp edge response in the middle of the step's plateau (Fig. 10, c, d). The illusory line vanishes when the width of step's plateau exceeds 26 pixels. The perception of illusory line might be related to the well known Chevreul illusion, in which a regular luminance staircase is perceived as not perfectly uniform along the luminance plateaus.

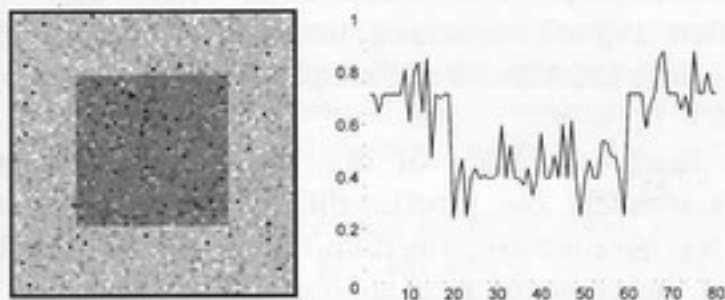


Figure 11. Noisy input image (left) and the cross section taken at the centre of the image (right).

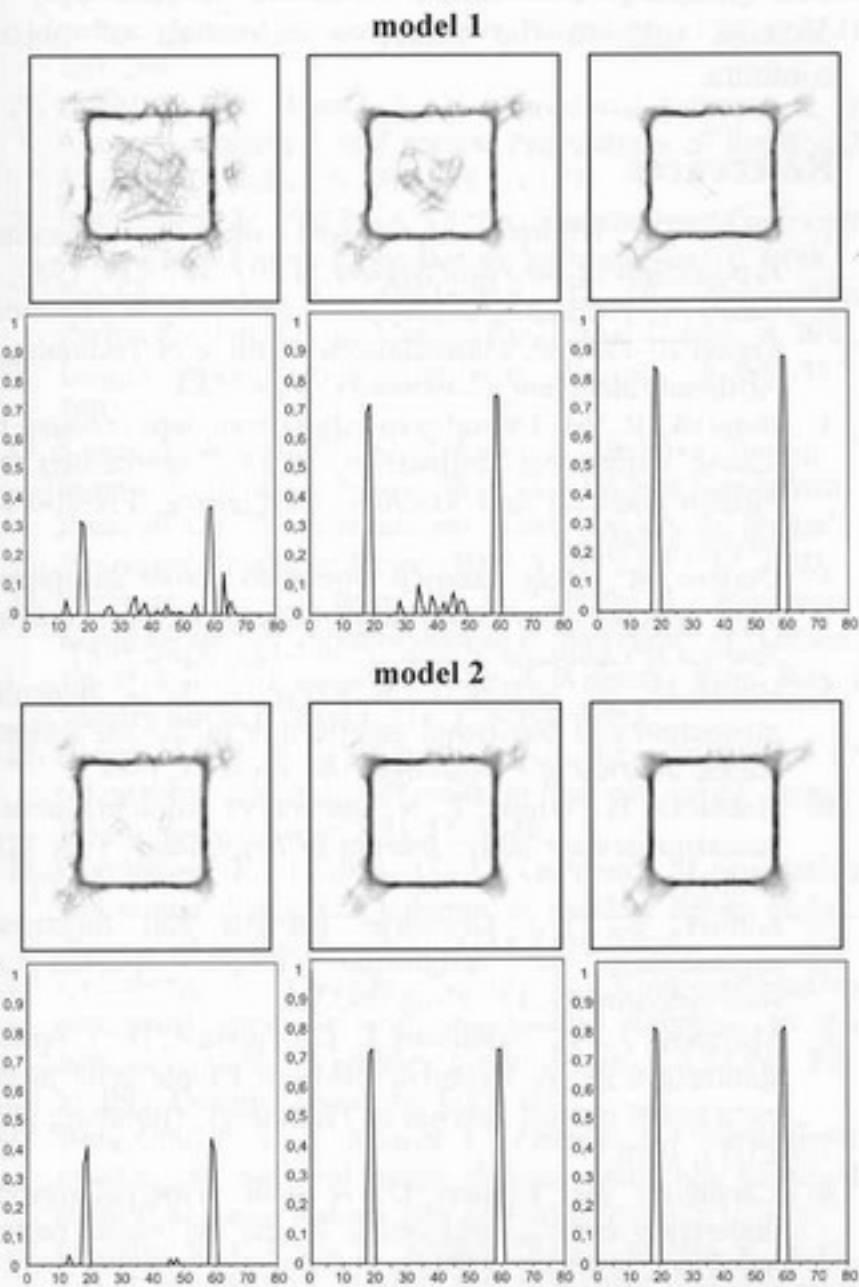


Figure 12. Edge responses to the noisy image (Fig.11) after 1st, 3rd, and 5th iteration generated by the model 1 (two upper rows) and model 2 (two bottom rows). Images of edge responses are inverted. Corresponding cross sections are taken at the centre of the images. Responses to noise weaken noticeably as the models proceed through iterative cycles. The cross-oriented inhibition introduces an additional mechanism of noise suppression for the model 2 due to suppression of the activity of cells at non-optimal orientations.

Noisy input. The processing of a synthetic image of a dark rectangle on a lighter background corrupted with 50% Gaussian noise (Fig. 11) exhibits that responses to weak contrast variations caused by noise are significantly diminished after several iterations (Fig. 12). Noise reduction is especially pronounced in the vicinity of rectangle edges, where small responses are “cancelled out” by stronger responses which spread in the process of iterative tuning.

5 Processing of natural images

Edge responses to natural images clearly illustrate a common feature exhibited by the two models, namely the enhancement of isolated weak edges at subsequent iterative cycles. The reason for this behaviour is a non-linear normalization of responses due to the cross-channel inhibition: the divisive normalisation tends to boost weaker responses over the stronger ones while normalising an overall magnitude of cell responses to the range [0,1].

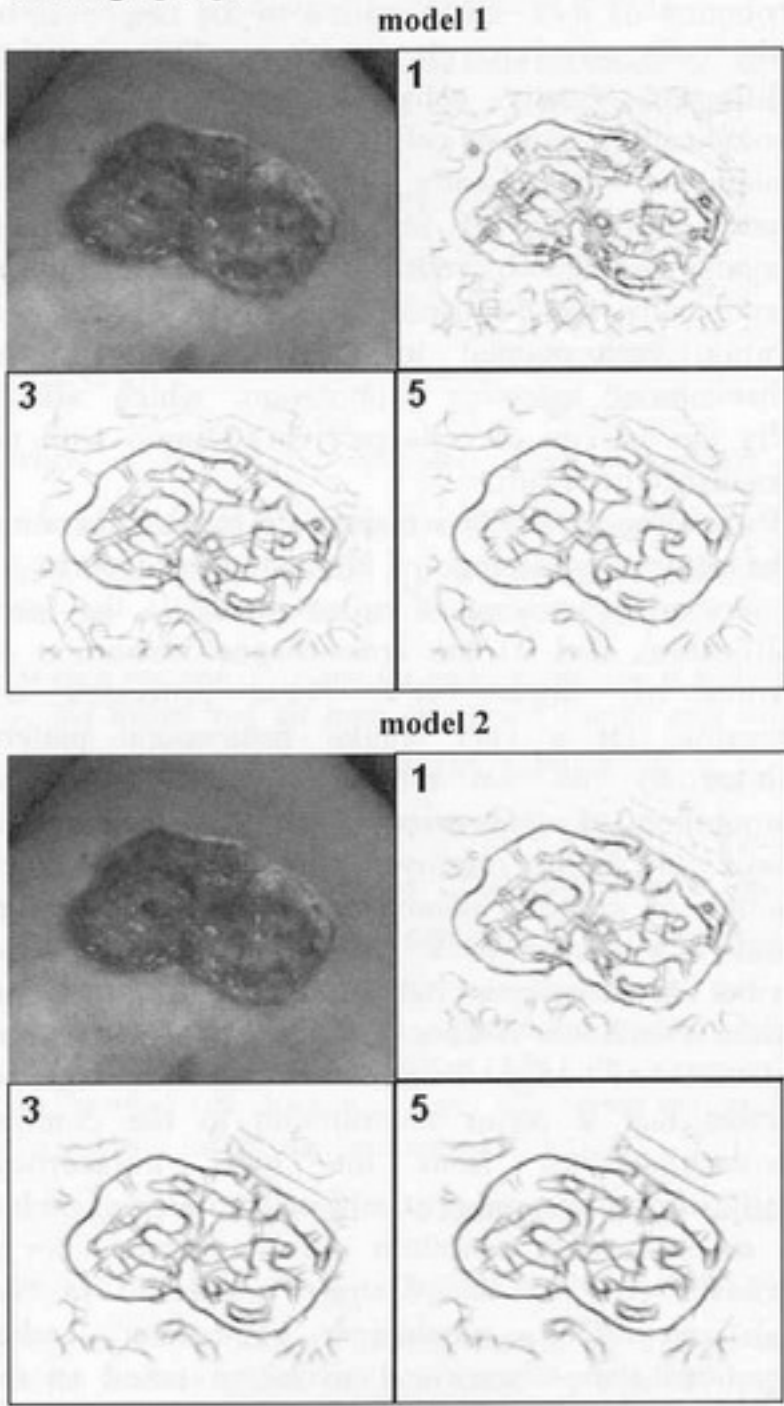


Figure 13. Input image of a wound and edge responses (inverted) generated by the two models at 1st, 3rd, and 5th iteration (clockwise from top left). Weak edges disappear and strong edges are amplified as the models proceed through iterative cycles. The iterative tuning diminishes spurious responses both for skin and the wound region. Edge enhancement is more pronounced for the model 2.

6 Summary and conclusions

The motivation of this work is the development of biologically justified model for contrast detection in images, the model, which has the potential to outperform purely computational approaches to edge detection. The orientation selectivity of simple cells in V1 gives us an exciting example of a biological system, which is capable of responding to oriented visual stimuli with great efficiency. We have proposed a model for the iterative orientation tuning with cross-oriented suppression. The model is an extension of the iterative orientation tuning model introduced in [23]. We have compared the performance of the two models by probing them with synthetic stimuli, synthetic noisy images and natural images.

The two major and common for both models features are the iterative processing of visual input and the local intracortical amplification of proximate cells belonging to a same channel. The local amplification explicitly exploits morphology of simple cells in V1. The contribution of local amplification to the responses of simple cells grows with iterations. This iterative amplification greatly enhances responses of both retinotopically proximate cells and cells tuned to similar orientations. Consequently, the local amplification activates a process of selective orientation tuning enhancing responses of cells of "proper" orientations at retinotopically close positions. The cross-orientation inhibition, incorporated in the model 2, is the mechanism of selective suppression, which affects locally the activity of cells receiving stimuli with no distinguished orientation.

Three processes play a major role in the generation of the contrast invariance of orientation selectivity: 1) the iterative processing of video input, 2) the local amplification, and 3) the cross-channel inhibition of activities of simple cells. These processes are responsible for a very similar behavioural pattern exhibited by the two models. We note that the incorporation of cross-oriented suppression into the model 2 only slightly improves the model's performance in terms of contrast invariance, which, on average, remains at a level of 75%. The orthogonal suppression does not seem to play a crucial role in the generation of contrast invariance: it does not significantly affect the magnitude of edge responses. Our investigation indicates that a major contribution to the contrast invariance comes from the local intracortical amplification of responses at subsequent iterative cycles. The cross-oriented inhibition did not account for a significant part of the contrast invariance in our simulations. This conclusion contradicts earlier suggestions that intracortical inhibition tuned to the orthogonal orientation plays a major role in the generation of cortical orientation selectivity [13], [24], [25]. However, our simulations do suggest that the cross-oriented suppression sharpens responses of simple cells. It appears that this sharpening accelerates the convergence of edge responses on a stable solution.

We conclude that the cross-oriented inhibition introduces an important stabilising factor into the process of orientation tuning, but cannot preserve the contrast invariance of orientation selectivity. Additional mechanisms may therefore be involved in the generation of the contrast invariance observed experimentally in monkeys and cats.

Although neither model includes any additional mechanism for the suppression of noise, both of them have demonstrated high robustness to noisy input. It seems that a good resistance to noise is an inherent feature of the functionality of simple cells taken over by the models for free.

One final conclusion of this study is as follows: however efficient the functionality of simple cells is, neither the iterative amplification of activities of simple cells nor the cross-oriented suppression, can provide a selective extraction of object edges. It seems that additional mechanisms such as object recognition linked with memory association feedback should play a decisive role in the selective extraction of object contours.

References

1. Marr, D., Hildreth, E. Theory of edge detection. *Proceedings of the Royal Society, B* 207: 187-217, 1980.
2. Canny, J., F. Finding edges and lines in images. *Technical Report AI-TR-720*, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, June 1983.
3. Haralick, R., M. Digital step edges from zero crossing of second directional derivatives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(1): 58-68, 1984.
4. Deriche, R. Using Canny's criteria to derive an optimal edge detector recursively implemented. *The International Journal of Computer Vision*, 2: 167-187, April 1987.
5. Hubel, D., H., Wiesel, T., N. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *Journal of Psychology*, 160:106-154, 1962.
6. Hubel, D., H., Wiesel, T., N.: Integrative action in the cat's lateral geniculate body. *Journal of Psychology*, 155: 385-398, 1961.
7. Kuffler, S., W.: Discharge patterns and functional organization of mammalian retina. *Journal of Neurophysiology*, 16: 37-68, 1953.
8. Movshon, J., A., Thompson, I., D., Tolhurst, D., J. Spatial summation in the receptive fields of simple cells in the cat's striate cortex. *Journal of Physiology*, (London), 283: 53-77, 1978.
9. Carandini, M., Ferster, D. A tonic hyperpolarization underlying contrast adaptation in the cat visual cortex. *Science*, 276:949-952, 1997.
10. Sclar, G., Freeman, R.: Orientation selectivity in the cat's striate cortex is invariant with stimulus contrast. *Experimental Brain Research*, 46: 457-461, 1982.
11. Albrecht, D., G., Geisler, W., S. Motion sensitivity and the contrast response function of simple cells in the visual cortex. *Visual Neuroscience*, 7:531-546.
12. Heeger, D., J. Nonlinear model of neural responses in cat visual cortex. In M. Landy, J. A. Movshon (eds.): *Computational models of visual processing*. Cambridge, MIT: 119-133, 1991.
13. DeAngelis G., C., Robson, J., G., Ohzawa, I., Freeman, R.,

- D. The organization of suppression in receptive fields of neurons in the cat's visual cortex. *J. Neurophysiology*. 68:144-163, 1992.
14. Carandini, M., Heeger, D., J. Summation and division by neurons in visual cortex. *Science*, 264:1333-1336, 1994.
 15. Pessoa, L., Mingolla, E., Neumann, H.: A contrast- and luminance-driven multiscale network model of brightness perception. *Vision Research*, 35:2201-2223, 1995.
 16. Neumann, H., Pessoa, L., Hansen, Th.: Interaction of ON and OFF pathways for visual contrast measurement. *Biological Cybernetics*, 81:515-532, 1999.
 17. Hansen, Th., Neumann, H. A model of V1 visual contrast processing utilizing long-range connections and recurrent interactions. In Proc. of the International Conference on Artificial Neural Networks, Edinburgh, UK, September 7-10: 61-66, 1999.
 18. Ferster, D.: The synaptic inputs to simple cells in the cat visual cortex. In: D. Lam and G. Gilbert (eds.): Neural mechanisms of visual perception, Ch. 3, Portfolio Publ. Co, The Woodlands, Texas: 63-85, 1989.
 19. Hubel, D., H., Wiesel, T., N.: Sequence regularity and geometry of orientation columns in the monkey striate cortex. *Journal of Comparative Neurology*, 158, (1974) 267-294.
 20. Hubel, D., H., Wiesel, T., N.: Functional architecture of macaque monkey visual cortex. Proceedings of the Royal Society of London, B, 198, (1977) 1-59.
 21. Braitenberg, V., Schüz, A. Cortex: Statistics and Geometry of Neuronal Connectivity. Berlin: Springer-Verlag, 1998.
 22. Koulakov, A., A., Chklovskii, D., B. Orientation Preference Patterns in Mammalian Visual Cortex: A Wire Length Minimization Approach. *Neuron*, 29:519-527, 2001.
 23. Kolesnik, M., Barlit, A., Zubkov, E. Iterative Tuning of Simple Cells for Contrast Invariant Edge Enhancement. Proc. of the 2nd International Workshop on Biologically Motivated Computer Vision (BMCV'2002): 27-37, 2002.
 24. Morrone, M., C., Burr, D., C., Maffei, L. Functional implications of cross-orientation inhibition of cortical visual cell. 1. Neurophysiological evidence. Proc. Royal Society London [Biol.], 216:335-354, 1982.
 25. Bonds, A., B. Role of inhibition in the specification of orientation selectivity of cells in the cat striate cortex. *Visual Neuroscience*, 2:41-55, 1989.
 26. Obermayer, K., Blasdel, G., G. Geometry of orientation and ocular dominance columns in monkey striate cortex. *Journal of Neuroscience*. 13:4114-4129, 1993.
 27. Grossberg, S., Raizada, R., D., S.: Contrast-sensitive perceptual grouping and object-based attention in the laminar circuits of primary visual cortex. CAS/CNS TR-99-008, Boston University: 1-35, 1999.
 28. Borg-Graham, L., J., Monier, C., Fregnac, Y.: Visual input invokes transient and strong shunting inhibition in visual cortical neurons. *Nature*, 393, (1998) 369-373.
 29. Douglas, R., J., Koch, C., Mahowald, M., Martin, K., A., C., Suarez, H., H. Recurrent excitation in neocortical circuits. *Science*, 269: 981-985, 1995.
 30. Ben-Yishai, R., Bar-Or, R., L., Sompolsky, H. Theory of orientation tuning in visual cortex. *Proc. National Academy of Science, USA*92: 3844-3848, 1995.
 31. Somers, D., C., Nelson, S., B., Sur, M. An emergent model of orientation selectivity in cat visual cortical simple cells. *Journal of Neuroscience*. 15: 5448-5465, 1995.
 32. Maex, R., Orban, G., A. Model circuit of spiking neurons generating directional selectivity in simple cells. *Journal of Neurophysiology*. 75: 1515-1545, 1996.
 33. Chung, S., Ferster, D. Strength and orientation Tuning of the Thalamic input to Simple Cells Revealed by Electrically Evoked Cortical Suppression. *Neuron*.20: 1177-1189, 1998.

Appendix

The processing of visual input undergoes several iterative cycles, each one containing either four (model 1) or five (model 2) subsequent stages. Stages #1, #2, #3, and #5 are common for both models. Stage #4 stands for the model 2. Below we list all processing stages noting explicitly the differences between the two models when these are present.

1. **Retina-LGN.** Responses of retinal ganglion ON and OFF cells, u_{ij}^+ and u_{ij}^- , are given by:

$$\begin{aligned} X_{ij} &= I_{ij} - G_{\sigma} * I_{ij}; \\ u_{ij}^+ &= [X_{ij}]^+; \quad u_{ij}^- = [-X_{ij}]^+ \end{aligned} \quad (A1)$$

where $*$ is the spatial convolution operator and G_{σ} is a centre Gaussian with standard deviation $\sigma=3$ for the model 1 and $\sigma=5$ for the model 2. The Gaussian is sampled within a filter mask of 35x35 and 45x45 pixels for the model 1 and model 2, respectively. Visual input I is normalised to $[0,1]$.

2. **Simple cell subfields.** Simple cells are modelled for twelve discrete orientations $\theta = 0^\circ, 15^\circ, \dots, 165^\circ$, and two opposite contrast polarities $p=1, 2$:

$$\begin{aligned} D_{\theta\sigma_M\sigma_m\tau} &= G_{\theta\sigma_M\sigma_m\tau} - G_{\theta\sigma_M\sigma_m-\tau}, \\ G_{\theta\sigma_M\sigma_m\tau} &= \frac{1}{2\pi} \exp\left[-\frac{1}{2}((\mathbf{x}-\boldsymbol{\tau})^T \mathbf{R}^T \mathbf{C} \mathbf{R}(\mathbf{x}-\boldsymbol{\tau}))\right] \\ \mathbf{C} &= \begin{pmatrix} 1/\sigma_m^2 & 0 \\ 0 & 1/\sigma_M^2 \end{pmatrix}, \quad \mathbf{R} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \end{aligned} \quad (A2)$$

where $\mathbf{x}^T = (i, j)$ denotes the position (i, j) , $\boldsymbol{\tau}^T = (\cos\theta, \sin\theta)$ is relative offset for two Gaussian lobes from their central position (i, j) , and the space constants $\sigma_m=1$ and $\sigma_M=4$ define the degree of filter's elongation.

At each position, (i, j) , and for each orientation, θ , and polarity, p , the model has an even symmetric simple cell with two parallel elongated parts: an ON subfield, $R_{i,j,\theta,p}$, which receives excitation from LGN ON cells beneath it, u_{ij}^+ , and is inhibited by input from the LGN OFF cells at the same position, u_{ij}^- ; and an OFF subfield, $L_{i,j,\theta,p}$, for which the reverse relation to the LGN channels holds true. This physiology is embodied in the equation for the ON subfield by subtracting the half-wave rectified LGN OFF channel, u^- , from the rectified ON channel, u^+ , and convolving the result with the positive lobe of the oriented filter, $[D_{\theta\sigma_M\sigma_m\tau}]^+$, [27]. The OFF subfield, $L_{i,j,\theta,p}$, is constructed similarly. In addition, each ON subfield, $R_{i,j,\theta,p}$, receives excitatory input, $A_{i,j,\theta,p}$, from all simple ON cells that are spatially close to position (i, j) in the *Iccube* layout, and is inhibited by input $B_{i,j,\theta,p}$ from all close OFF cells. The reverse arrangement holds true for the computation of the activation level of each OFF subfield, $L_{i,j,\theta,p}$. The mutual amplification-inhibition of neighbouring cells is a time varying function which is updated iteratively.

The above considerations give rise to the following expressions for $R_{i,j,\theta,p}^n$ and $L_{i,j,\theta,p}^n$, at iteration n :

$$\begin{aligned} R_{i,j,\theta,p}^n &= \left[(u_{ij}^+ + A_{i,j,\theta,p}^n - u_{ij}^- - B_{i,j,\theta,p}^n) * [D_{\theta\sigma_M\sigma_n\tau}^p] \right]^+ \quad (\text{A3}) \\ L_{i,j,\theta,p}^n &= \left[(u_{ij}^- + B_{i,j,\theta,p}^n - u_{ij}^+ - A_{i,j,\theta,p}^n) * [-D_{\theta\sigma_M\sigma_n\tau}^p] \right]^+ \end{aligned}$$

The strength of local interaction for both models, $A_{\theta,p}$, $B_{\theta,p}$, that we call amplification functions, varies over time. The values of amplification functions at initial iteration $n=0$, are set to $A_{\theta,p} = B_{\theta,p} = 0$, for all orientations and polarities. Note, that due to the offset of the positive and negative lobes of $D_{\theta\sigma_M\sigma_n\tau}$, subfield responses are shifted from their central positions. To compensate, both subfields, $R_{i,j,\theta,p}$ and $L_{i,j,\theta,p}$, are shifted in the opposite directions, τ and $-\tau$, respectively.

3. *Cross-channel inhibition.* The activation of simple ON cell, S_{on}^n , at iteration n , is obtained as the steady-state solution of inhibitory shunting interaction:

$$S_{on}^n = \left[(R^n - L^n) / (1 + R^n + L^n) \right]^+ \quad (\text{A4})$$

Here variables occur for all positions, orientations and polarities; indexes i , j , θ , and p are omitted to simplify notations. Activation of simple OFF cell is obtained by interchanging R^n and L^n .

4. Cross-oriented suppression.

Model 2: Simple cells are engaged in the cross-orientation inhibition, so that the cells' activity at position (i,j,θ) , is inhibited by four neighbouring cells in the 3-D array (see caption to Fig.2) that are tuned for the orthogonal direction:

$$\begin{aligned} \hat{S}_{i,j,\theta}^n &= \left[S_{i,m,\vartheta}^n - (S_{i+1,m,\vartheta}^n + S_{i-1,m,\vartheta}^n + S_{i,m+1,\vartheta}^n + S_{i,m-1,\vartheta}^n) / 4 \right]^+ \quad (\text{A5}) \\ \begin{cases} \vartheta = \theta + \pi/2 & \text{if } 0 \leq \theta < \pi/2 \\ \vartheta = \theta - \pi/2 & \text{if } \pi/2 \leq \theta < \pi \end{cases} \end{aligned}$$

5. *Local amplification.* At each at position (i,j,θ) in the 3-D array, the excitatory input, $A_{i,j,\theta}^n$, from proximate ON cells, is an inverse function of squared distance:

Model 1:

$$A_{i,j,\theta}^n = \mu \sum_{l,m,\vartheta} \frac{S_{(on),l,m,\vartheta}^n}{D^2[(l,m,\vartheta),(i,j,\theta)]} \quad (\text{A6})$$

Model 2:

$$A_{i,j,\theta}^n = \mu \sum_{l,m,\vartheta} \frac{\hat{S}_{(on),l,m,\vartheta}^n}{D^2[(l,m,\vartheta),(i,j,\theta)]}$$

$$D^2[(l,m,k),(i,j,\theta)] = \omega[(l-i)^2 + (m-j)^2] + (\theta - \vartheta)^2$$

where μ - is a scaling factor set to: $\mu = 0.18$, and ω - is a weighting factor set to $\omega = 16$. Above computations are repeated twice for both polarities. Excitatory input B to an OFF-cell is obtained by substituting S_{off} .

Biological Motivated Multi-modal Processing of Visual Primitives

Norbert Krüger*

*Department of Psychology
University of Stirling
Scotland, UK
norbert@cn.stir.ac.uk

Markus Lappe†

†Psychologisches Institut
Universität Münster
Germany
mlappe@uni-muenster.de

Florentin Wörgötter*

‡Department of Psychology
University of Stirling
Scotland, UK
worgott@cn.stir.ac.uk

Abstract

We describe a new kind of image representation in terms of local multi-modal Primitives. These Primitives are motivated by processing of the human visual system as well as by functional considerations. We discuss analogies of our representation to human vision and concentrate specifically on the implications of the necessity of communication of information in a complex multi-modal system.

1 Introduction

In this paper, we describe a new kind of image representation in terms of local multi-modal Primitives (see figure 1). These Primitives are motivated by processing in the human visual system as well as by functional considerations. The work described here has been evolved from a project started in 1998 which has been focussed on the integration of visual information (ModIP, 1998—). The image representation described here is now a central pillar of the ongoing European project ECOVISION (ECOVISION: 2002–2004) that focusses on the functional modeling of early visual processes.

In the human visual system beside local orientation also other modalities such as color and optic flow (that are also part of our multi-modal Primitives) are computed in the hyper-columns of V1 (see, e.g. (Hubel and Wiesel, 1969; Gazzaniga, 1995)). *All these low level processes face the problem of an extremely high degree of vagueness and uncertainty (Aloimonos and Shulman, 1989).* This arises from a couple of factors. Some of them are associated with image acquisition and interpretation: Owing to noise in the acquisition process along with the limited resolution of cameras, only erroneous estimates of semantic information (e.g., orientation) are possible. Furthermore, illumination variation heavily influences the measured grey level values and is hard to be modeled analytically (Ikeuchi and Horn, 1981). Information extracted across image frames, e.g., in stereo and optic flow estimation, faces (in addition to the above mentioned problems) the correspondence and aperture problem which interfere in a fundamental and especially difficult way (see, e.g., (Ayache, 1990; Klette et al., 1998)).

However, the human visual systems acquires visual representations which allow actions with high precision and certainty within the 3D world under rather uncontrolled conditions. *The human visual system can achieve the needed certainty and completeness by integrating visual information across modalities and by utilizing spa-*

tial and temporal interdependencies (see, e.g., (Phillips and Singer, 1997; Hoffman, 1980)). This integration is manifested in the huge connectivity between brain areas in which the different visual modalities are processed as well as in the large number of feedback connections from higher to lower cortical areas (see, e.g., (Gazzaniga, 1995)). The essential need for integrating visual information in addition to optimizing single modalities to design efficient artificial visual systems has also been recognized in the computer vision community after a long period of work on improving single modalities (Aloimonos and Shulman, 1989).

However, integration of information makes it necessary that local feature extraction is subject to modification by contextual influences. As a consequence *adaptability* must be an essential property of the visual representation. Moreover, the exchange of information between visual events has necessarily to be paid for with a certain cost. This cost can be reduced by limiting the amount of information transferred from one place to the other, i.e. by reducing the bandwidth. This is the reason why we are after a *condensed* description of a local image patch, which however *preserves the relevant information*. Here relevance has to be understood not only in an information theoretical sense, but in a global sense (the system has to be subject to modifications by global interdependencies, in particular local entities have to be connectable to more complex entities) and action oriented sense (the transferred information has to be relevant for the actions the individual has to perform).

Taking the above mentioned considerations into account, the Primitives, which are the basic entities of our image representation, can be characterized by four properties:

Multi-modality: Different domains that describe different kind of structures in visual data are well established in human vision and computer vision. For example, a local edge can be analyzed by local feature attributes such as orientation or energy

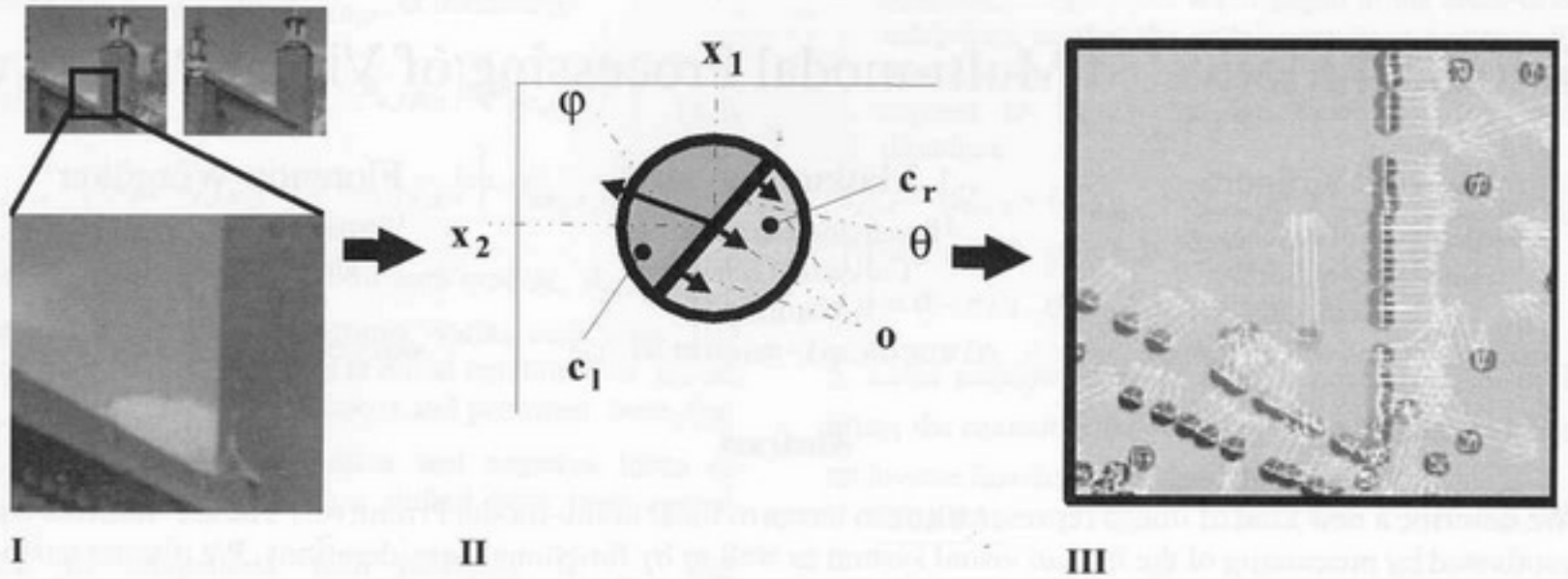


Figure 1: **I:** Image sequence and frame. **II:** Schematic representation of the multi-modal Primitives. **III:** Extracted Primitives at position with high amplitude.

in certain frequency bands (see, e.g., Krüger and Sommer (2002)). In addition, we can distinguish between line and step-edge like structures (contrast transition). Furthermore, color can be associated to the edge. This image patch also changes in time due to ego-motion or object motion. Therefore time specific features such as a 2D velocity vector (optic flow) are associated to our Primitives (see figure 1).

Adaptability: Since the interpretation of local image patches in terms of the above mentioned attributes as well as classifications such as ‘edgeness’ or ‘junctionness’ are necessarily ambiguous when based on local processing, stable interpretations can only be achieved *through integration* by making use of contextual information (Aloimonos and Shulman, 1989). Therefore, all attributes of our Primitives are equipped with a confidence that is essentially *adaptable according to contextual information* expressing the reliability of the attribute. Furthermore, feature attributes itself are subject of correction mechanisms that use contextual information.

Condensation: Integration of information requires *communication between Primitives* expressing spatial (Krüger and Wörgötter, 2002; Krüger et al., 2002b) and temporal dependencies (Krüger et al., 2002a). This communication has necessarily to be paid for with a certain cost (as will be made explicit in section 3). This cost can be reduced by limiting the amount of information transferred from one place to the other, i.e., by reducing the bandwidth. Therefore we are after a *condensed* representation. Also for other tasks it is essential to store information in a *condensed* way, e.g., for the learning of objects to reduce memory requirements.

Meaningfulness: Communication and memorization not only require a reduction of information. We want to reduce the amount of information within an image patch *while preserving perceptually relevant information*. This leads to *meaningful* descriptors such as our attributes position, orientation, contrast transition, color and optic flow.

We will describe our feature processing in section 2 and will compare it to early human visual processing in section 3.

2 Feature Processing and Application

In this section we describe the coding of modalities associated to our Primitives. In addition to the position x , we compute the following semantic attributes and associate them to our Primitives (see also figure 1).

Frequency: We describe the signal on different frequency levels f independently. Often the decision in which frequency band the relevant information does occur is difficult, therefore we leave this decision open to be decided at later stages of processing. It may be even that for the same position on different frequency levels there occur different kind of semantic information (for example, the top of the toy in figure 2:A on a high frequency level can be described as texture-like while on a lower frequency level it resembles an edge).

Orientation: The local orientation associated to the image patch is described by θ . The orientation θ is computed by interpolating across the orientation information of the whole image patch to achieve a more reliable estimate.

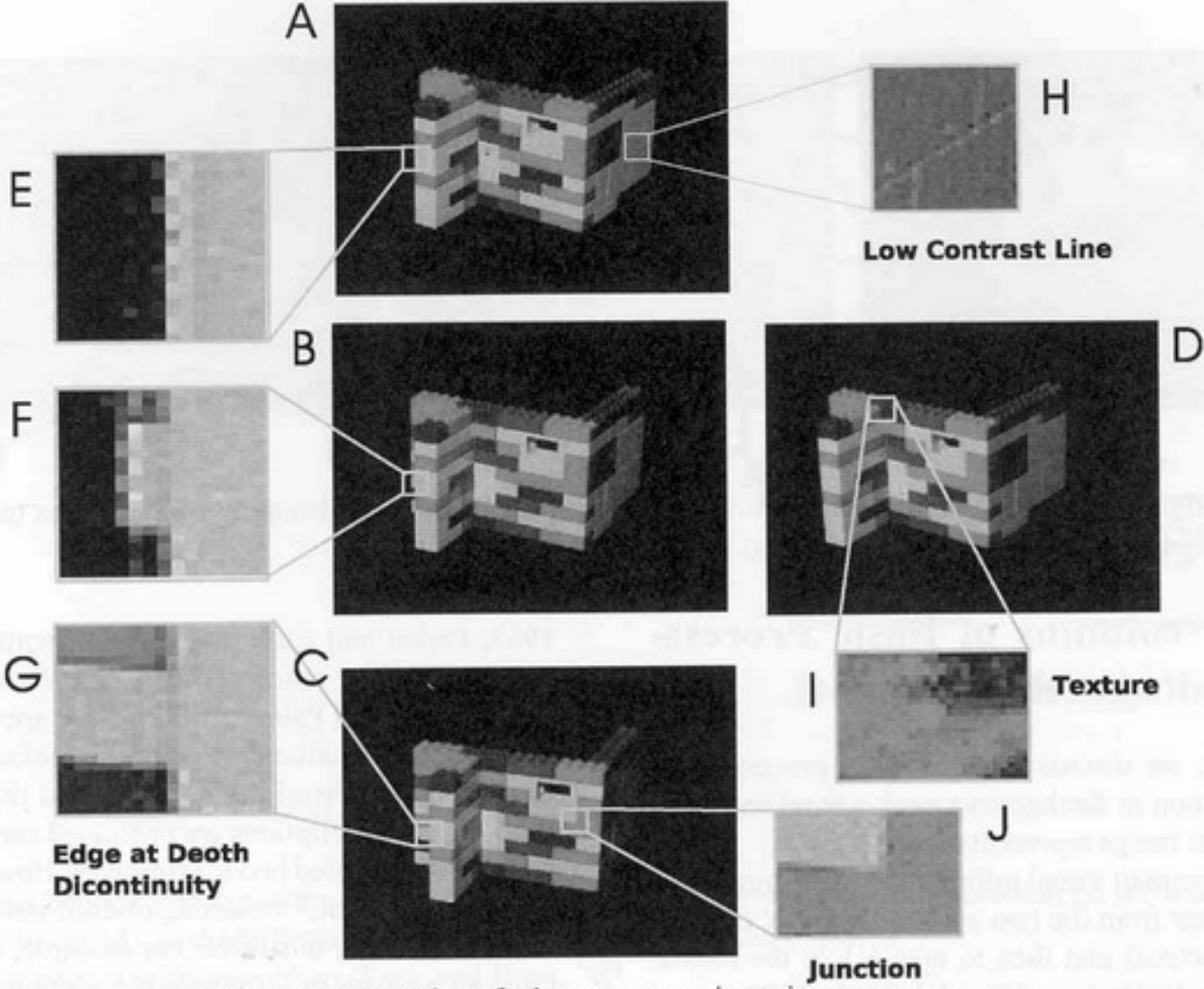


Figure 2: Examples of edge structures in an image sequence.

This holds also true for the following feature attributes contrast transition, colour and optic flow.

Contrast transition: The contrast transition is coded in the phase φ of the applied filter (Felsberg and Sommer, 2001). The phase codes the local symmetry, for example a bright line on a dark background has phase 0 while a bright/dark edge has phase $-\pi/2$ (in figure 3 the line that marks the border of the street is represented as a line or two edges depending on the distance from the camera). In case of boundaries of objects, the phase represents a description of the transition between object and background (see, (Kovesi, 1999; Krüger and Wörgötter, 2002)).

Color: Color (c^l, c^m, c^r) is processed by integrating over image patches in coincidence with their edge structure (i.e., integrating separately over the left and right side of the edge as well as a middle strip in case of a line structure). In case of a boundary edge of a moving object at least the color at one side of the edge is expected to be stable (see figure 2E–G) since it represents a description of the object.

Optic Flow: Local displacements $\mathbf{o}$ is computed by the well known optic flow technique (Nagel, 1987).

Furthermore, we represent the system’s confidence c that the entity e does exist. We end up with a parametric description of a Primitive as

$$E = (\mathbf{x}, f, \theta, \varphi, (c^l, c^m, c^r), \mathbf{o}; c).$$

In addition, to each of the parameters $\varphi, (c^l, c^m, c^r), \mathbf{o}$ there exist confidences $c_i, i \in \{\varphi, c^l, c^m, c^r, \mathbf{o}\}$ that code the reliability of the specific sub-aspects that is also subject to contextual adaptation.

We have applied our image representation in different contexts. First, an image patch also describes a certain region of the 3D space and therefore 3D attributes can be associated such as a 3D-position and a 3D-direction. In (Krüger et al., 2002b), we have defined a stereo similarity function that makes use of multiple-modalities to enhance matching performance. Second, the Primitives can be subject to spatial contextual modification. We define groups of Primitives based on a purely statistical criterion in (Krüger and Wörgötter, 2002) (see also figure 3III). Once these groups are defined, we modulate the confidences of our Primitives: confidences become increased if the Primitives are part of a bigger group, otherwise the confidences are decreased. Thirdly, we have stabilized features according to the temporal context. In (Krüger et al., 2002a,c), we make use of the motion of an object to predict feature occurrences and showed that we can stabilize stereo processing by modifying the confidences according to the temporal context.

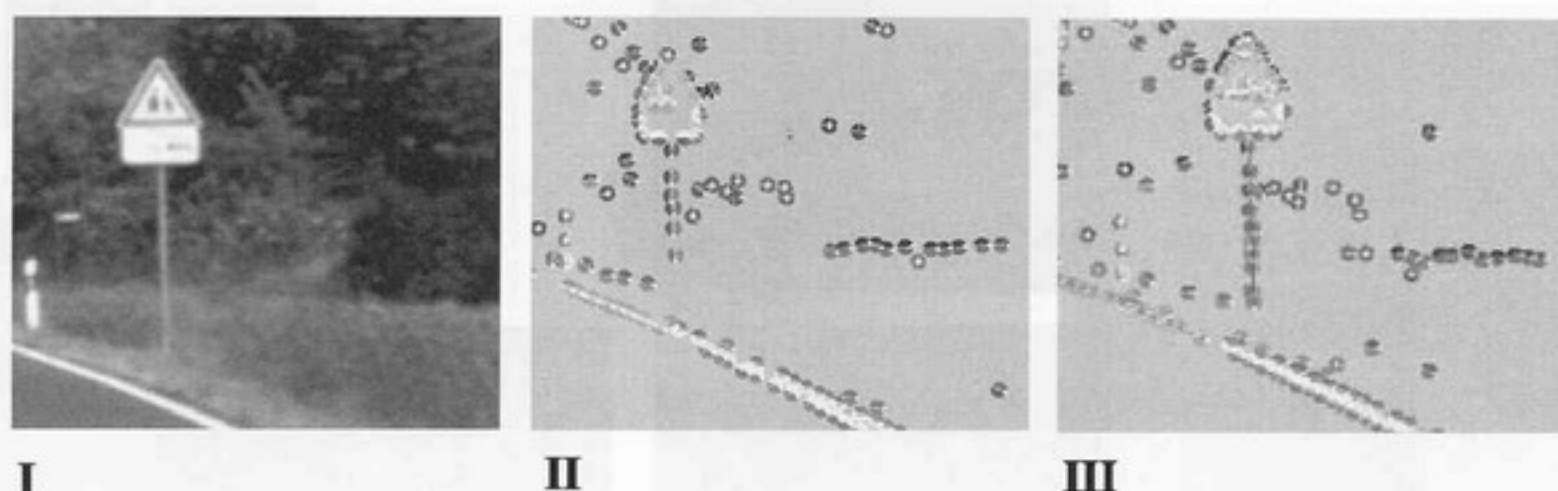


Figure 3: I: Original Image. II: Extracted Primitives with high amplitude. III: Grouping of Primitives (elements of the same group are indicated by the identical colours).

3 Hyper-columns of Basic Processing Units in early Vision

In this section, we discuss aspects of the processing of visual information in the human visual system and draw analogies to our image representation.

The main stream of visual information in the human visual system goes from the two eyes to the LGN (Lateral Geniculate Nucleus) and then to area V1 in the cortex. (see figure 4 and (Wurtz and Kandel, 2000a)). There are two kind of cell types involved, M (magnocellular) and P (parvocellular) cells that have different response characteristics: M cells have a low spatial but high temporal resolution and are not colour sensitive. In contrast to M cells, P cells have a low temporal and high spatial resolution and are colour sensitive. Both kind of cells project into two cortical pathways, the dorsal and ventral pathway (see figure 4). The ventral pathway goes from the cortical area V1 to V2 to the Inferior Temporal Area (IT) and is believed to be mainly responsible for object recognition (see, e.g., (Tanaka, 1993)). In the dorsal stream information is transferred from V1 to MT (Middle Temporal Area) to MST (Medial Superior Temporal Area) and is believed to be involved in the analysis of motion and spatial information.

V1 (or Visual Area 1) is the main input of both pathways. The structure of V1 has been investigated by Hubel and Wiesel in their ground-breaking work (Hubel and Wiesel, 1962, 1969). V1 is organized in a retinotopic map that has a specific repetitively occurring pattern of substructures called hyper-columns. Hyper-columns themselves consists of so called orientation columns and blobs (see figure 5). The main input of V1 comes from the LGN and targets to layer 4 to which information of both eyes projects (see figure 5Aiii).

The orientation columns are organized in an ordered way such that columns representing similar orientations tend to be adjacent to each other (see figure 5Ai). However, it is not only orientation that is processed in an orientation column but the cells are sensitive to additional attributes (see figure 5D) such as disparity (Barlow et al.,

1967; Parker and Cumming, 2001), motion (Wurtz and Kandel, 2000b), colour (Hubel and Wiesel, 1969) and phase (Jones and Palmer, 1987). Also specific responses to junction-like structures could be measured (Shevelev et al., 1995). Therefore, it is believed that in V1 basic local feature descriptions are processed similar to the feature attributes coded in our Primitives. However, since the processing is local,¹ the ambiguities of visual information is not resolved at this level. For example, response properties of neurons in V1 reflect the aperture problem (see, e.g., (Stumpf, 1911)). This holds also for our Primitives since the flow is also computed by a local operation.

It is believed that mainly form is processed in the ventral pathway. Neurophysiological equivalents of illusionary contours can be detected in V2 but not in V1 (von der Heydt et al., 1984). This is not surprising since illusionary contours like in the Kanizsa triangle (Kanizsa, 1976) presuppose an integration of information across a large spatial domain as well as across different feature types (e.g., edges and junctions) and can therefore only be processed at a later stage.

The different visual modalities are not computed independently but are combined. For example in V1 the processing of motion is necessarily intertwined with the processing of orientation because of the aperture problem. In V4, colour and orientation is combined (Wurtz and Kandel, 2000b). Accordingly, in our image representation the coding of colour is deeply intertwined with the coding of orientation: Colour is a feature that describes homogeneous surfaces. However, orientation describes discontinuities and can be used to separate the surfaces. In our image representation we therefore first compute orientation and then compute a left and a right colour according to this orientation.

In the dorsal pathway mainly motion is analysed. Like the occurrence of illusionary contours presuppose global interactions, the aperture problem can only be solved by taking the global context into account. This does not hap-

¹There is a high connectivity within a hyper-column. There exist also connections across hyper-columns. However their distribution falls sharply with distance.

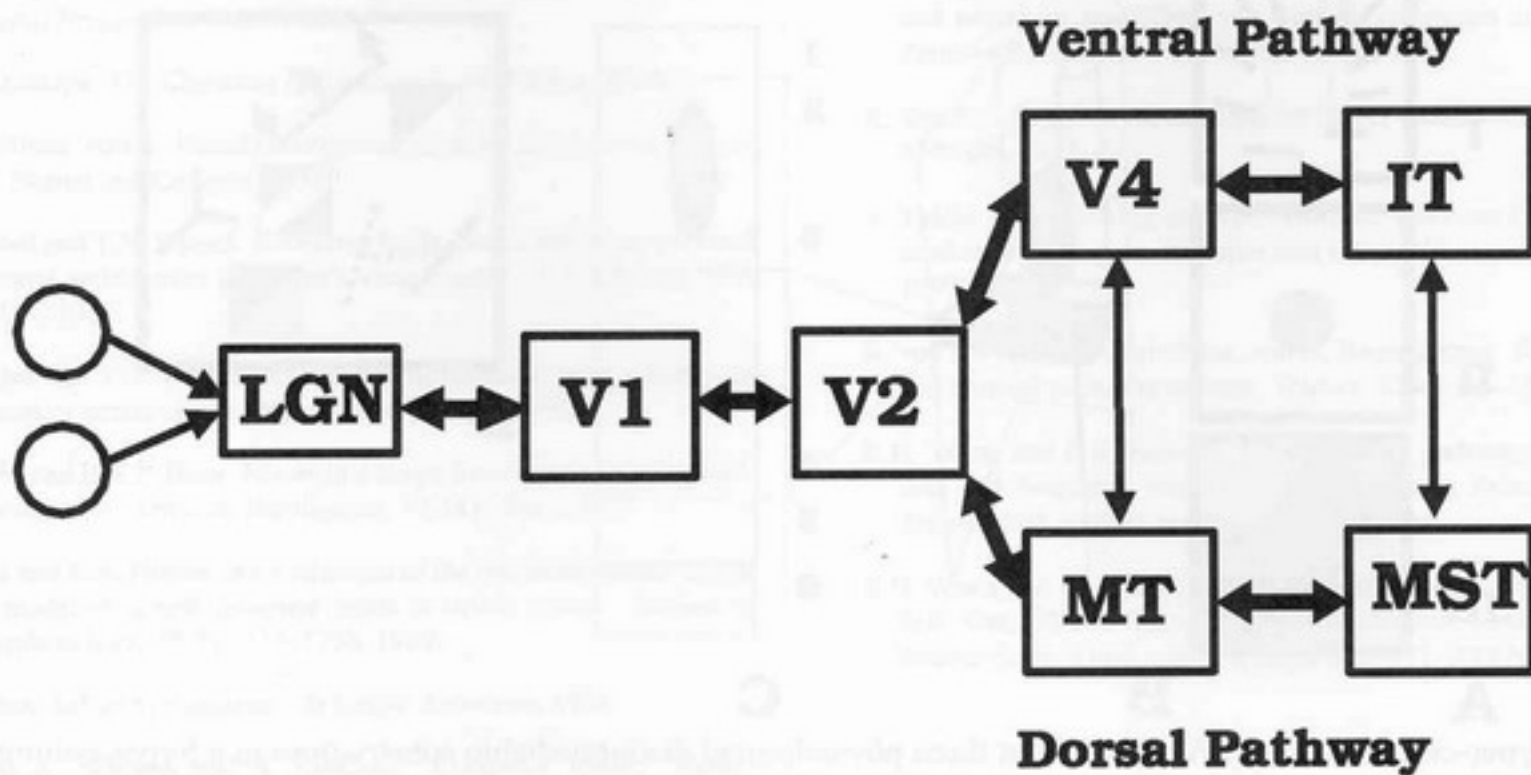


Figure 4: Flow of visual information in the human visual system (schematic).

pen (and can not happen because of the local processing) in V1. However, in MT and MST many cell responses indicate a solution to aperture problem (Pack and Born, 2001; Wurtz and Kandel, 2000b). Similar to the cells in V1, our Primitives also reflect the aperture problem. However, we can use the output of our Primitives to apply global mechanisms that disambiguate the local flow (see, e.g., (Krüger et al., 2002a)).

As in the ventral pathway, cells in the dorsal pathway show multi-modal response patterns. For example, a moving edge may be not visible as a luminance edge but can be constituted by colour or texture. MT cells respond to this kind of structures although they are not sensitive to colour alone (Thiele et al., 1999; Wurtz and Kandel, 2000b).

Let us summarize: In V1 visual information is mainly locally processed. However, some semi-local interactions exist. The ambiguities of visual information can not be resolved at this stage of processing. A specialization to form processing (along the ventral pathway V1–V2–V4–IT) and motion processing (along the dorsal pathway V1–V2–MT–MST) does occur.

As mentioned above, stable and reliable information can only be achieved by disambiguation through integration. However, this integration process makes the exchange of information within and across visual areas mandatory. As discussed before, intra-areal connections are very limited. However, inter-areal connection project to a much wider field of the next layer.

Regarding communication between visual areas we have to address two issues:

- 1) What is the bandwidth of information we want to transfer (“quantity”)?

- 2) What kind of information do we want to communicate (“quality”)?

The first question leads to a reflection about costs of communication. In any communication system transfer of information is associated to a cost which normally increases with the amount of information to be transferred and with the distance to be covered. This could concern the costs of “cables” but also the cost of the energy used for the transfer.

In the brain the communication between two neurons is realized by an axon docking to the soma or the dendrites of other neurons. Accordingly, the complexity and, thus, the “cost” of communication increases with the number of connections. This holds in a very general sense and may have been one driving force for the bandwidth reduction that is actually observed in neuronal visual processing. This bandwidth reduction most clearly manifests itself in mechanisms of visual attention and visual awareness. Focussed attention is often taken as one central mechanisms used to reduce the bandwidth of computation as well as of information transfer in the brain to a manageable degree. Anatomically the band-width limitation requirement may be reflected by the density of fibers which connect different areas which is smaller than that which connects cells within a hypercolumn.

A similar mechanism is used also in our image representation where we arrive at a significant reduction of information following the first processing stages. Compared to an average sized image patch of 15×15 pixels represented by a Primitive the output of a Primitives has less than 20 values, i.w., we have a compression rate of more than 90%. This rate becomes even higher when we compare the output of a Primitive to intermediate local stages of processing where feature attributes for all

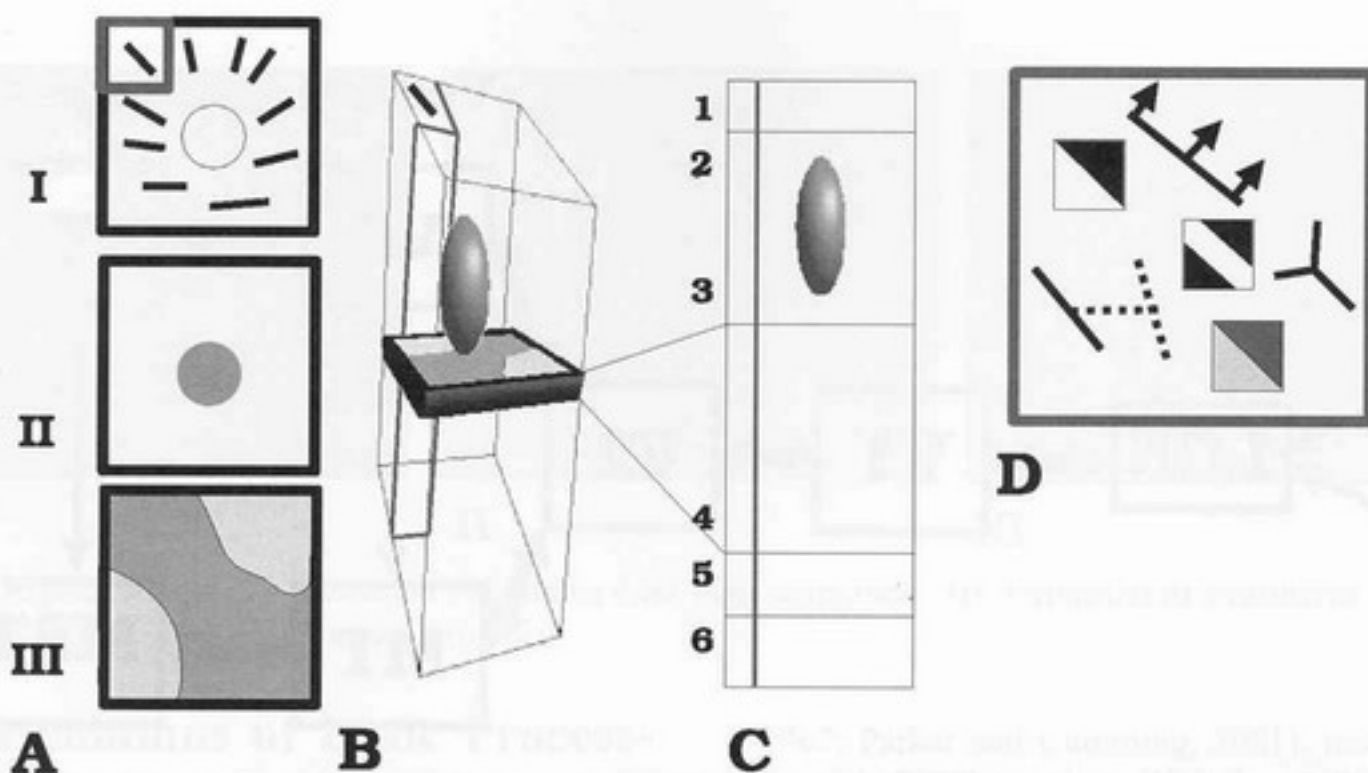


Figure 5: Hyper-columns in V1. A: There exist three physiological distinguishable substructure in a hyper-column: (i) In orientation columns information about oriented edge structure is represented in a topological way. (ii) Colour information is coded in so called ‘blobs’. (iii) Information of both eyes are input to the fourth cortical layer (see also B). B: Three dimensional structure of a hyper-column. C: Organisation in cortical layers. D: Feature attributes that are coded in a hyper-column.

modalities are derived for each pixel.

The second question above concerns the quality of information which needs to be transferred between the different stages of visual processing. Here we refer back to what we have said above noting that pre-processed visual information is exceedingly ambiguous as the consequence of fundamental problems in image data acquisition as well as resulting from the intrinsic structure of the detectors (receptive fields). This leads to the situation that redundant information must be transferred because only through redundancy it can be assured that erroneous information can be disambiguated. For this it is required that a visual event which is represented by the firing of neuron A has a relevance for the event represented by B. Since event A is supposed to be used to correct event B both events will have to be correlated which can be measured by the expression

$$\frac{P(B|A)}{P(B)}. \quad (1)$$

If this term takes a high value then there is a high likelihood of the occurrence of event B when we know event A has occurred compared to the likelihood of the occurrence of the event without prior knowledge. In this case, events A and B can be used to mutually correct each other because they are carrying shared (i.e.; redundant) information. The expression (1) has been called ‘Gestalt coefficient’ in (Krüger, 1998) where it was shown that applying binarised Gabor wavelets to natural images, a high Gestalt coefficient corresponds to the Gestalt laws Collinearity and Parallelism. As an extension of (Krüger, 1998) it has been shown in (Krüger and Wörgötter, 2002) that using by our multi-modal Primitives we can increase

the expression (1) significantly compared to using orientation only (Krüger, 1998). That means that by using our Primitives we can increase the interdependency of visual events. In this way in our Primitives not only information is condensed but transferred to *more meaningful descriptors*.

4 Conclusion and Outlook

We have introduced a novel way to compute visual Primitives which are motivated by early processing in the human visual system in analogy to the output of V1 hyper-columns. These Primitives are multi-modal and give a dense and meaningful description of a scene. Our Primitives can adapt according to the spatial and temporal context that is realized in the human visual system through a high synaptic connectivity. In this way the locally unreliable feature extraction can be disambiguated and stable feature representations can be achieved.

References

- J. Aloimonos and D. Shulman. *Integration of Visual Modules — An extension of the Marr Paradigm*. Academic Press, London, 1989.
- N. Ayache. *Stereovision and Sensor Fusion*. MIT Press, 1990.
- H. Barlow, C. Blakemore, and J.D. Pettigrew. The neural mechanisms of binocular depth discrimination. *Journal of Physiology (London)*, 193:327–342, 1967.
- ECOVISION: Artificial visual systems based on early-cognitive processing, EU-Project. <http://www.pspc.dibe.unige.it/ecovision/project.html>, 2002–2004.

- M. Felsberg and G. Sommer. The monogenic signal. *IEEE Transactions on Signal Processing*, 41(12), 2001.
- M.S. Gazzaniga. *The Cognitive Neuroscience*. MIT Press, 1995.
- D.D. Hoffman, editor. *Visual Intelligence: How we create what we see*. W.W. Norton and Company, 1980.
- D.H. Hubel and T.N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J. Physiology*, 160: 106–154, 1962.
- D.H. Hubel and T.N. Wiesel. Anatomical demonstration of columns in the monkey striate cortex. *Nature*, 221:747–750, 1969.
- K. Ikeuchi and B.K.P. Horn. Numerical shape from shading and occluding boundaries. *Artificial Intelligence*, 17:141–184, 1981.
- J.P. Jones and L.A. Palmer. An evaluation of the two dimensional Gabor filter model of simple receptive fields in striate cortex. *Journal of Neurophysiology*, 58(6):1223–1258, 1987.
- G. Kanizsa. Subjective contours. *Scientific American*, 1976.
- R. Klette, K. Schlüns, and A. Koschan. *Computer Vision - Three-Dimensional Data from Images*. Springer, 1998.
- P. Kovesi. Image features from phase congruency. *Videre: Journal of Computer Vision Research*, 1(3):1–26, 1999.
- N. Krüger. Collinearity and parallelism are statistically significant second order relations of complex cell responses. *Neural Processing Letters*, 8(2):117–129, 1998.
- N. Krüger, M. Ackermann, and G. Sommer. Accumulation of object representations utilizing interaction of robot action and perception. *Knowledge Based Systems*, 15:111–118, 2002a.
- N. Krüger, M. Felsberg, C. Gebken, and M. Pörksen. An explicit and compact coding of geometric and structural information applied to stereo processing. *Proceedings of the workshop 'Vision, Modeling and VISUALIZATION 2002'*, 2002b.
- N. Krüger, T. Jäger, and Ch. Perwass. Extraction of object representations from stereo image sequences utilizing statistical and deterministic regularities in visual data. *DAGM Workshop on Cognitive Vision*, 2002c.
- N. Krüger and F. Wörgötter. Multi modal estimation of collinearity and parallelism in natural image sequences. *Network: Computation in Neural Systems*, 13:553–576, 2002.
- V. Krüger and G. Sommer. Wavelet networks for face processing. *JOSA*, 19:1112–1119, 2002.
- ModIP. (Modality Integration Project). <http://www.cn.stir.ac.uk/ComputerVision/Projects/ModIP/index.html>, 1998—.
- H.-H. Nagel. On the estimation of optic flow: Relations between different approaches and some new results. *Artificial Intelligence*, 33: 299–324, 1987.
- C. C. Pack and R. T. Born. Temporal dynamics of a neural solution to the aperture problem in visual area MT of macaque brain. *Nature*, 409:1040–1042, 2001.
- A.J. Parker and B.G. Cumming. Cortical mechanisms of binocular stereoscopic vision. *Prog Brain Res*, 134:205–16, 2001.
- W.A. Phillips and W. Singer. In search of common foundations for cortical processing. *Behavioral and Brain Sciences*, 20(4):657–682, 1997.
- I.A. Shevelev, N.A. Lazareva, A.S. Tikhomirov, and G.A. Sharev. Sensitivity to cross-like figures in the cat striate neurons. *Neuroscience*, 61:965–973, 1995.
- P. Stumpf. über die abhängigkeit der visuellen bewegungsrichtung und negativen nachbildes von den reizvorgängen auf der netzhaut. *Zeitschrift für Psychologie*, 59:321–330, 1911.
- K. Tanaka. Neuronal mechanisms of object recognition. *Science*, 262: 685–688, 1993.
- A. Thiele, K.R. Dobkins, and T.D. Albright. The contribution of color to motion processing in macaque area mt. *J. Neurosci.*, 19:6571–6587, 1999.
- R. von der Heydt, E. Peterhans, and G. Baumgartner. Illusory contours and cortical neuron responses. *Science*, 224:1260–62, 1984.
- R.H. Wurtz and E.R. Kandel. Central visual pathways. In E.R. Kandel, J.H. Schwartz, and T.M. Messel, editors, *Principles of Neural Science (4th edition)*, pages 523–547. 2000a.
- R.H. Wurtz and E.R. Kandel. Perception of motion, depth and form. In E.R. Kandel, J.H. Schwartz, and T.M. Messel, editors, *Principles of Neural Science (4th edition)*, pages 548–571. 2000b.

ROBUST LIP MODELING AND EXTRACTION USING ACTIVE SHAPES

I. Shdaifat and R. -R. Grigat*

*TU Vision Systems

Technical University of Hamburg Harburg

shdaifat@tu-harburg.de, grigat@tu-harburg.de

Abstract

We present a new model for the lips using the Active Shape Model (ASM) based on Bézier Curves that can capture the dynamics of the lips efficiently. The simplicity of the model makes it possible to describe the lips just only with few points, where we use these points as visual speech features. The ASM is constructed using only few images of the lips of the speaker. Accurate detection of features of the lips is also implemented using multiple decomposed independent feature templates. Using some detected feature points and the ASM, we can recover the full shape of the lips robustly.

1. INTRODUCTION

Lip detection and tracking in image sequences and relating features of the lips to speech are two challenging problems. This is due to the lack of dominant image features defining the lips Lucey S. and Chandran (2000).

Several methods have been proposed for lip detection and visual feature extraction. For example, deformable templates are effective in lip tracking Ramos Sanchez and Kittler (1997). In Michael T. Chan (1998) shape contour using a cubic B-spline model was used efficiently. The Active Shape Model (ASM) was presented for lip modeling Matthews et al. (2002). Our work is similar to that of Matthews et al. (2002), but their model lacks an accurate definition of lip textures, and lip boundaries are defined using several points and not a simple parametric curve. Moreover, they use the normalized intensity derivative, we believe it is sensitive to noise and not sufficient for the detection of lip boundaries. Instead, we use template matching of several features.

Xiaozheng Zhang (2001) suggested a model with two feature points for the corner of the mouth, but these two points are correlated, also the inner corners of the mouth do not appear for all of the speaker, and they are not a robust feature especially for small images. For lip boundaries, they use r , g , and b color information, we use more features for robust detection.

In order to overcome the problem of insufficiency of lip features, we find some features of the lips which can be robustly detected. Then we use the method proposed in V. Blanz (2002) to correct the measurement using the model data.

For lip modeling, we constructed our own Visual SAMPA German database (SAMPA is Speech Assessment Methods Phonetic Alphabet). It consists of 43x10 sequences corresponding to the phonemes of the German

language. Each word is uttered twice by 10 speakers. In Sec. 2 the lip modelling, tracking, and recognition will be discussed. The results are presented in Sec. 3.

2. LIP DETECTION, MODELING, AND TRACKING

2.1. Finding the Lips' Features

Fig. 2 shows two lip templates and the optimal feature decomposition of the templates that maximizes localization accuracy when matching them. We define ζ as the ratio between the lip width w and the distance between the upper lip and the face midpoint d_{mf} , see Fig. 1. We assume that the face has been detected and the distance d_{mf} is known, and an approximate location of the mouth has been found from a preprocess. The width of the lips can be approximated with $w = d_{mf}/\zeta$, which is used to scale the templates. Four regions of interest can be set around the possible locations of the lips features of Fig. 2. As we have no idea about the shape of the lips, we match all the templates to find the four principal points of the mouth.

The corners of the mouth are good features to be detected better than the upper and lower lip especially in the case of rotation. We perform an affine transformation for the lip templates where the parameters of the transformation are obtained from a preprocess, see Fig. 3. For the first iteration, we search for the corners using the rotated templates with the angle obtained from the face recognition process. The face recognition is decoupled for the next frames, and the angle of the previous frame is used to rotate the template.

We match the templates as follows: If an object has n different characteristic features F_n , with ρ_i as similarity measure for the feature F_i , then the similarity between two objects can be defined as $\sigma = \sum_{i=1}^n w_i \rho_i$, where

$-1 \leq \rho_i \leq 1$ and $\sum_{i=1}^n w_i = 1$. For lip matching we use the Y_{uv} values in addition to the gradient. The combination of these features results in best localization and outlier rejection.

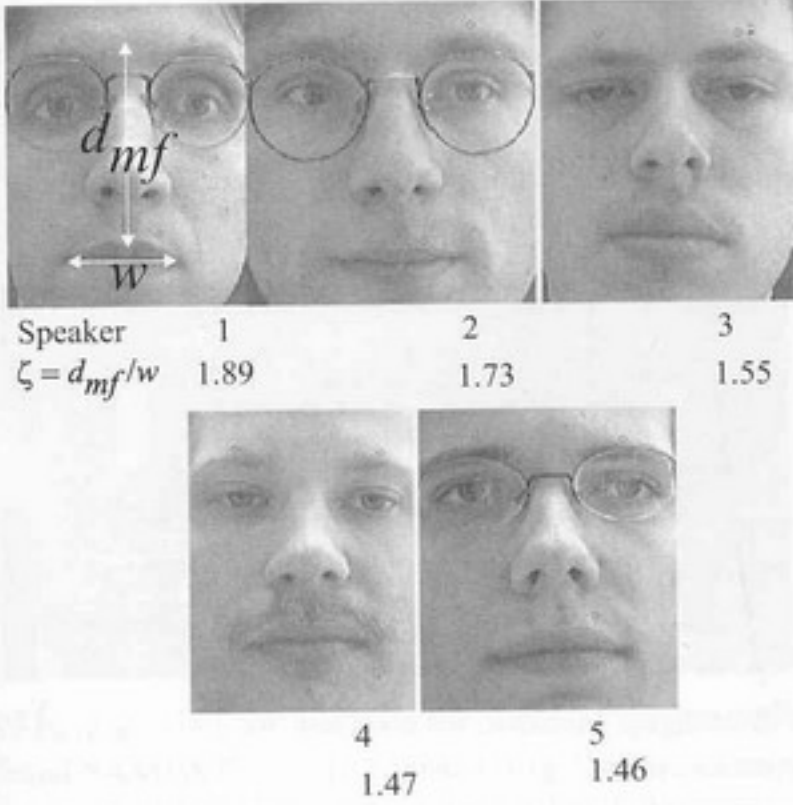


Fig. 1. ζ for different persons with closed mouth.

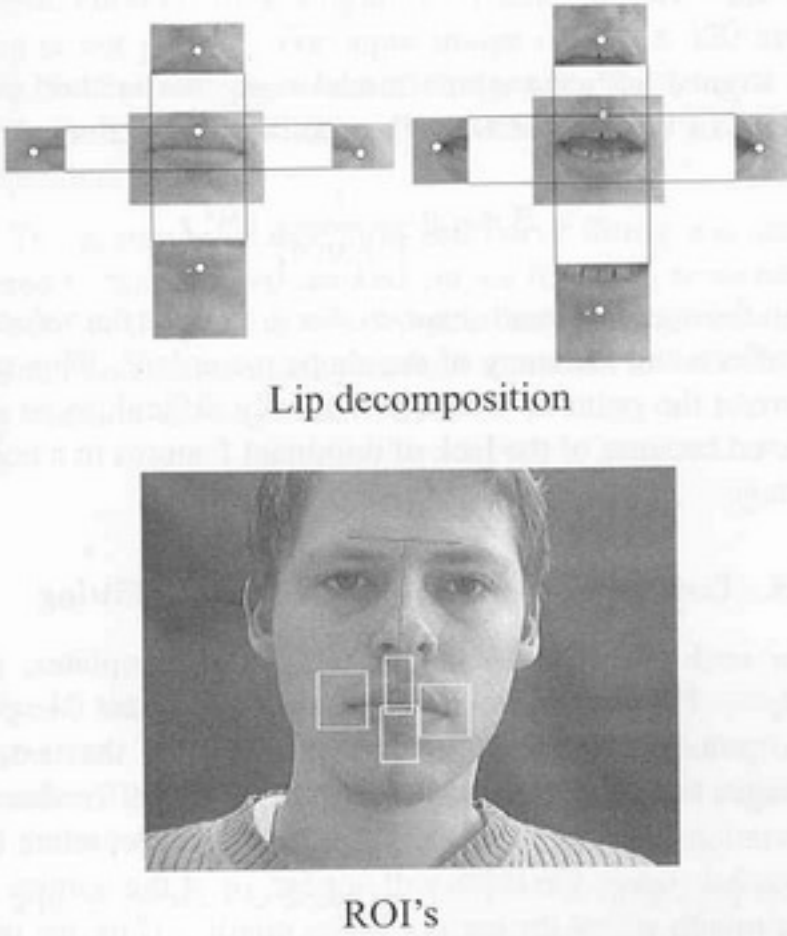


Fig. 2. Decomposition of the lips and ROI of lip features.

2.2. Modeling of the lips

Fig. 4 shows our proposed model for the lips. The model consists of 5 Bézier Curves. Each curve is defined by two

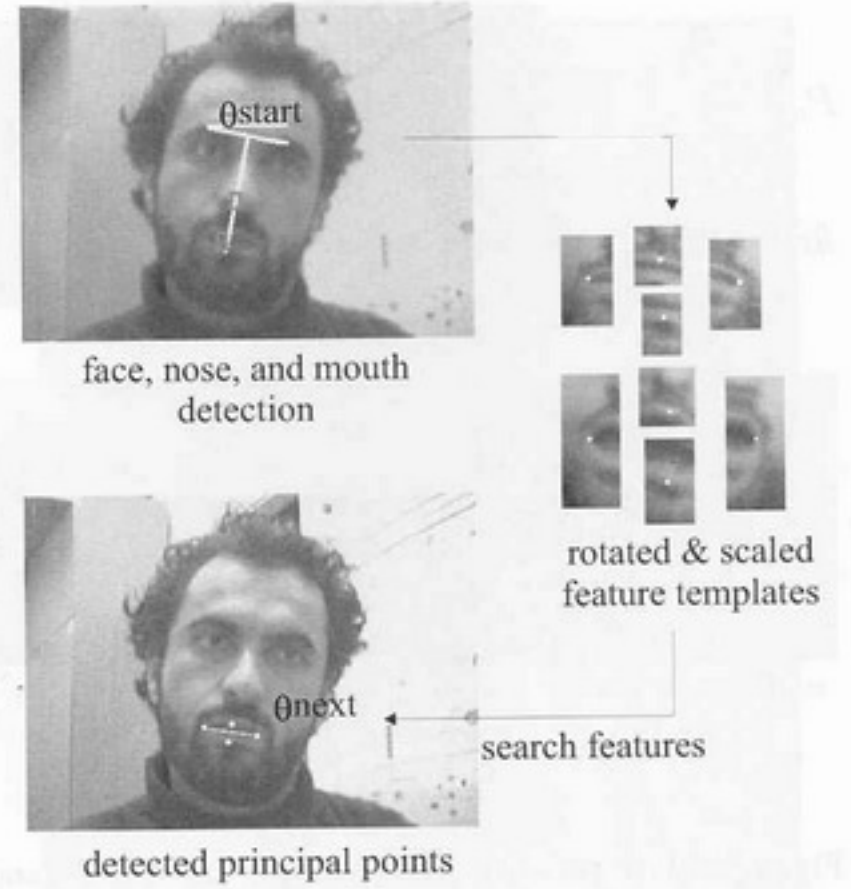


Fig. 3. Face detection and feature template scale and rotation.

end points P_0 , P_2 , and one control point P_1 , which can be written as

$$P(t) = \phi_0(t)P_0 + \phi_1(t)P_1 + \phi_2(t)P_2.$$

The model consists of three end points e_1 , e_2 , and e_3 and five control points c_1 , c_2 , c_3 , c_4 , and c_5 . These control points compact 150 feature points define the lip boundaries. The model is able to capture the dynamics of the lips and fit any shape of the lips in SAMPA German Database.

2.3. Active Shape Model (ASM)

The dynamics of the lips can be described by the ASM Matthews et al. (2002); V. Blanz (2002). We can write a given set of distributed points of a certain shape as a vector v_i , where

$$v_i = (x_{i0}, y_{i0}, x_{i1}, y_{i1}, \dots, x_{in-1}, y_{in-1})^T \in \mathbb{R}^n.$$

For the lip model, these are the coordinates of the control points of Bézier curves $e_1, e_2, e_3, c_1, c_2, c_3, c_4$, and c_5 , normalized to d_{mf} . The statistics of the distribution of these points can be captured as follows: The mean shape is $\bar{v} = \frac{1}{N} \sum_{i=1}^N v_i$. The deviation from the mean is $x_i = v_i - \bar{v}$. The covariance matrix is given by $C = \frac{1}{N} \sum_{i=1}^N x_i x_i^T$. C can be decomposed using singular value decomposition (SVD) $C = S \text{diag}(\sigma_i^2) S^T$ where $S = (s_1, s_2, \dots)$ are the eigenvectors and $(\sigma_1, \sigma_2, \dots)$ are the standard deviations within the data along each eigenvector s_i . By Principal Component Analysis (PCA) V. Blanz (2002) we can write any new

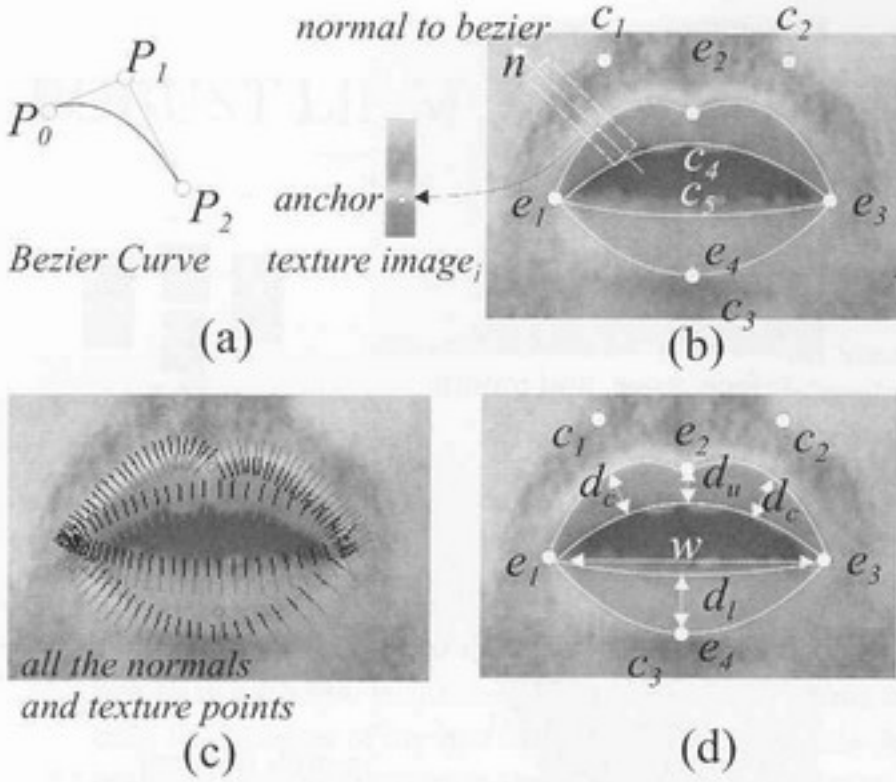


Fig. 4. (a) A Bézier curve, (b) lip model, (c) textures, and (d) distances used to complete the model from principal points.

element of the lip class

$$\mathbf{x} = \sum_{i=1}^M c_i \sigma_i \mathbf{s}_i = \mathbf{S} \text{diag}(\sigma_i^2) \mathbf{c}$$

Any shape of the lips can be approximated by $\mathbf{v} = \mathbf{x} + \bar{\mathbf{v}}$.

For the purpose of model construction, it is sufficient to start with two extreme cases of the lips' shapes as initialization denoted as $\mathbf{v}_1$ and $\mathbf{v}_2$, see Fig. 5. Any other shape of the lips can be approximated in the span of $\mathbf{v}_1$ and $\mathbf{v}_2$ by

$$\mathbf{v} = \frac{\mathbf{v}_1 + \mathbf{v}_2}{2} + c \frac{\mathbf{v}_1 - \mathbf{v}_2}{2}.$$

$\mathbf{v}_1$ is mouth fully closed, like the phoneme *m* in the word "Hammer" and $\mathbf{v}_2$ the mouth is fully open, like the phoneme *A* in "Aachen". For some speakers, we need an extra template $\mathbf{v}_3$ representing the viseme with rounded appearance like the phoneme *u* in "Blut" for a more accurate model. For each new frame in a sequence, the model is updated using the new automatically recognized shapes $\mathbf{v}_i$.

2.4. Completing the Model

In Sec. 2.1 we found the four principal points e_1, e_2, e_3 and e_4 of the lips. From these points we can approximate the rest of the shape. The average $\bar{\mathbf{v}}$ is used as a reference. Referring to Fig. 4 (d), w is the distance between e_1 and e_2 . The average $\bar{\mathbf{v}}$ is scaled with $w/\bar{w}$, consequently the distances d_c, d_u and d_l can be calculated from the scaled average. The rest of the shape c_1, c_2, c_3, c_4 and c_5 can be approximated from these values. We denote the resulting measured values with $\mathbf{r} = [e_1, e_2, e_3, c_1, c_2, c_3, c_4, c_5]$. $\mathbf{r}$

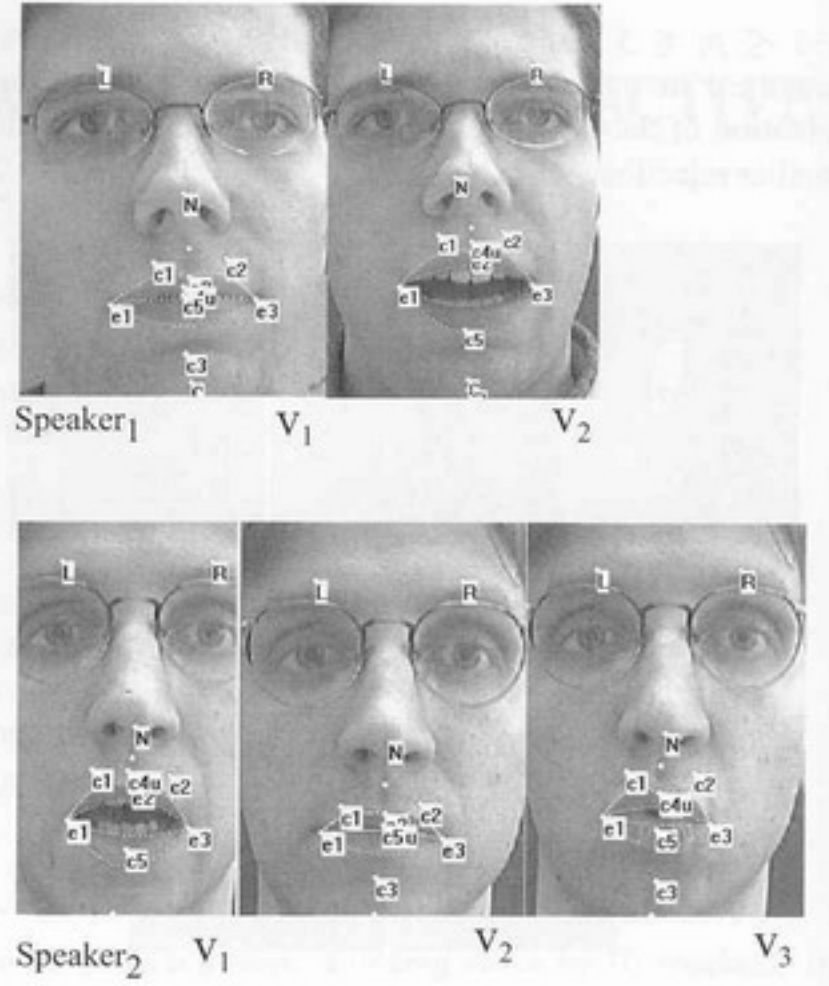


Fig. 5. Marking feature points of the lips and Bézier control points manually. The results are initial lip templates for two speakers

is aligned with the whole model using the method proposed in V. Blanz (2002). The resulting new value

$$\mathbf{x} = \mathbf{S} \text{diag}\left(\frac{1}{1 + \eta/\sigma_i^2}\right) \mathbf{S}^T \mathbf{r},$$

and the approximated shape $\mathbf{v} = \mathbf{x} + \bar{\mathbf{v}}$, where the value of η affects the tendency of the shape towards $\bar{\mathbf{v}}$. This will correct the point e_4 which is normally difficult to be detected because of the lack of dominant features in a noisy image.

2.5. Textures of the Shape and Boundary Fitting

For each curve in the model, we use 30 templates, see Fig. 4 (b) (c). Each template is normal to the curve at the point $\mathbf{P}(t_i)$. In Matthews et al. (2002) the texture images were assumed to be the same for all modes of variations, which is not the case. For example, when the mouth is open, the teeth will appear, or at the corners of the mouth where the texture varies rapidly. Thus, we take two classes of textures representing the extreme cases of open and closed mouth. Using the approximated shape $\mathbf{v}$ in Sec. 2.4, we select the nearest texture class suitable for this shape. We fix the endpoints for every Bézier curve, and iterate the control point stepwise as seen Fig. 6 until the maximum match score is reached. All templates are matched with the lip image for each step. If the results are consistent and the match score between the model and the

lips is above a certain threshold, we only search in a small ROI around each lip feature see Fig. 2.

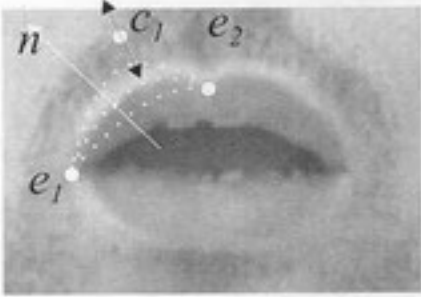


Fig. 6. Tracking of Bézier curve with texture templates.

3. RESULTS

We have applied our methods for different speakers of the Visual SAMPA German Database. Fig 7 shows examples of images of three speakers, the articulated phonemes are marked for each image.

Fig. 9 shows the four principal points as white squares, first approximated shape (gray), and the final fitted shape (white curves). In this figure for frame number 4 the fitting is not perfect. For input image of 240 x 320 pixel size, the lip tracking runs at 15 images per second on a 1.5 GHz Pentium computer using a webcam, with a speaker dependent profile.

The accuracy of detection and curve fitting was compared to that of hand marked curves for some sequences. We define the average relative error between the measured value r and that manually marked v relative to the average lip width in the sequence $\overline{w}$ as

$$e = \frac{1}{n} \sum_{i=1}^n \|v_i - r_i\| / \overline{w}.$$

Fig. 10 shows e for three speakers of three different image sizes. The error is less than 3% for $w > 30$ pixel for speaker 2 and 3, whereas speaker 1 has 4 % error because of the moustache.

Fig. 8 shows the distances $\overline{e_1e_3} = W$ and $\overline{e_2c_3} = H$ vs. time for the three words "Aachen", "Hammer", and "Blut", uttered twice from Visual SAMPA German Database. We can notice that for the phoneme "a" the mouth hight approaches its peak, whereas for the phonemes "m" and "b" the valley is reached. For the phoneme "u" the mouth width decreases remarkably for speakers 2, 5, and 6. A detailed study of the behavior of these waveforms with respect to speech is an ongoing research.

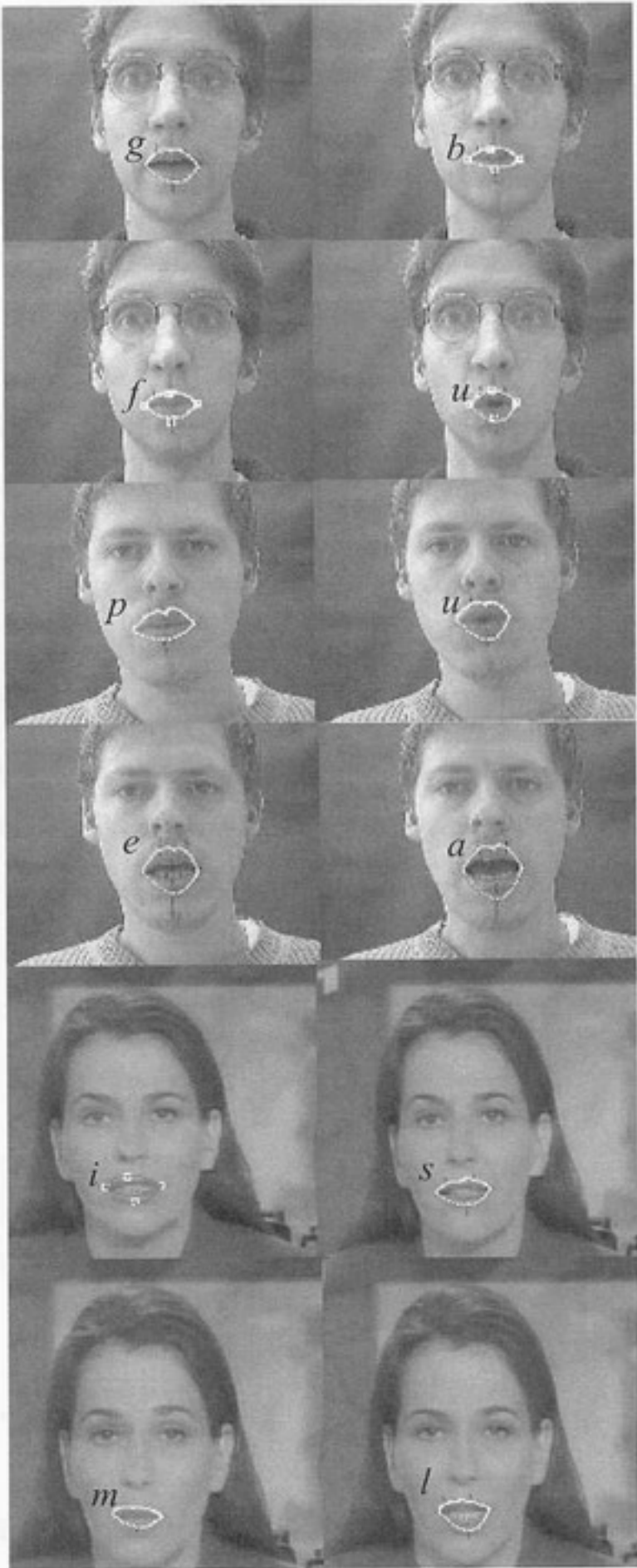


Fig. 7. Some uttered phonemes for different speakers and the results of lip detection.

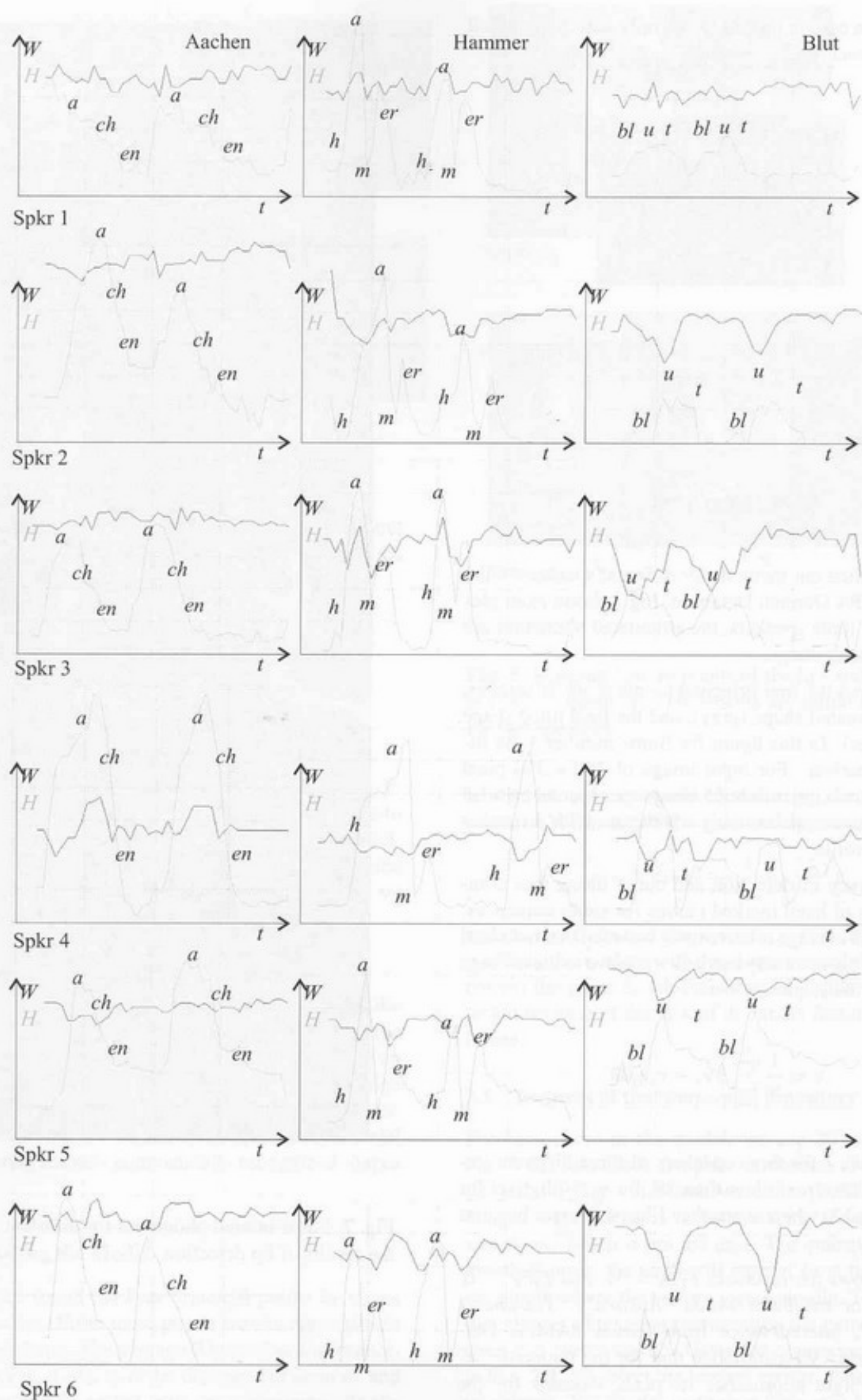


Fig. 8. Some samples of waveforms of the lip width W (black curves) and height H (gray curves) versus time t for 3 words of 6 speakers. W and H are normalized to the distance d_{mf} (see Fig. 1)

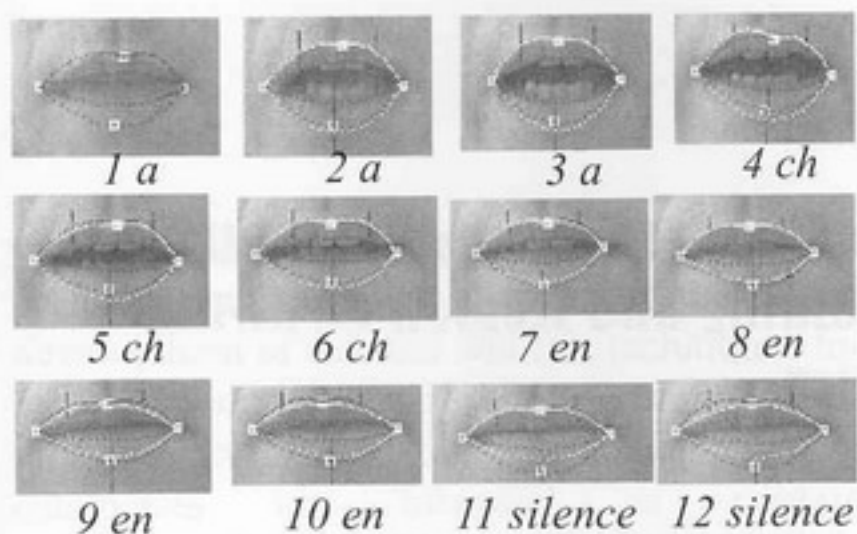


Fig. 9. Lip modeling and tracking for the word "Aachen".

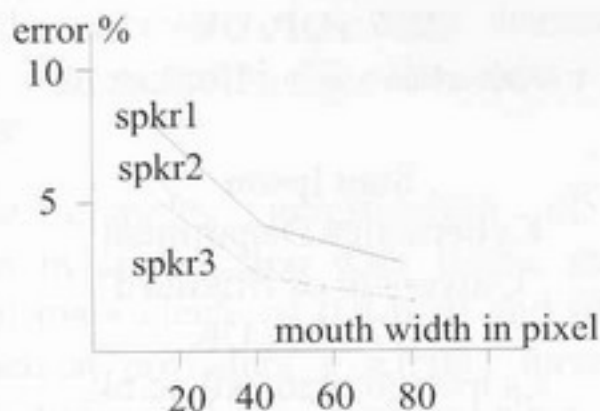


Fig. 10. Average relative error for different lip width.

References

- Srinidharan S. Lucey S. and V. Chandran. Initialized eigenlip estimator for fast lip tracking using linear regression. *ICPR*, 2000.
- I. Matthews, T.F. Cootes, J.A. Bangham, S. Cox, and R. Harvey. Extraction of visual features for lipreading. *PAMI*, 24(2):198–213, February 2002.
- Thomas S. Huang Michael T. Chan, You Zhang. Real-time lip tracking and bimodal continuous speech recognition. In *IEEE Signal Processing Society 1998 Workshop on Multimedia Signal Processing*, pages 65–70, Los Angeles, CA, Dec 1998. IEEE.
- J. Matas Ramos Sanchez and J. Kittler. Statistical chromaticity models for lip tracking with b-splines. In *Proceedings of the First International Conference on Audio- and Video-based Biometric Person Authentication, Lecture Notes in Computer Science*, Springer Verlag, pages 69–76, 1997.
- T. Vetter V. Blanz. Reconstructing the complete 3d shape of faces from partial information. *it - Information Technology*, 44(6):295–302, December 2002.
- Charles C. Broun Xiaozheng Zhang. Using lip features for multimodal speaker verification. In *A Speaker Odyssey - The Speaker Recognition Workshop*, Crete, Greece, 2001.

Automatic Identification of Active Regions (Plages) in the Full-Disk Solar Images Using Local Thresholding and Region Growing Techniques

Ali Benkhalil
Cybernetics Department
University of Bradford
BD7 1DP, UK
a.k.benkhalil@bradford.ac.uk

Sergei Zharkov
Cybernetics Department
University of Bradford
BD7 1DP, UK
s.zharkov@bradford.ac.uk

Valentina Zharkova,
Cybernetics Department
University of Bradford
BD7 1DP, UK
v.v.zharkova@bradford.ac.uk

Stan Ipson
Cybernetics Department
University of Bradford
BD7 1DP, UK
s.s.ipsen@bradford.ac.uk

Abstract

The paper presents techniques developed for the automated detection of Active Regions comprising bright areas on a darker background using $H\alpha$ and Ca KII 3 solar images from the Meudon observatory. Statistical information for each quarter of the solar full-disk is calculated and used to determine the intensity threshold values which are used for initial segmentation of the active regions within each quarter. Median filtering and morphological operations are applied to the resulting binary image to remove noise and merge the broken regions. The centroid of each labelled region is calculated for use as a seed for a region growing procedure. Statistically based local thresholding is used to calculate upper and lower threshold values defining the spatial extent of regions. This procedure has been tested on one month sequences (April 2002) of full-disk solar images from Meudon observatory and with active regions classified by the National Oceanic and Atmospheric Administration (NOAA) observatory in Hawaii. A very good correspondence of identified active regions was found.

1 Introduction

There are a growing number of the archives of digitised images of the Sun taken from ground-based and space instruments in various wavelengths. These archives are available from different locations and are to be included into a unified catalogue by the EGSO

project (Bentley, 2001). The digitized solar images have different sizes, resolutions instrumental and weather distortions. All are to be subjected to the automated recognition process in order to provide a reliable location of features and their evolution on different times of solar rotation. With growing demand for solar activity forecasts by the space

weather project and by many industrial organisations there is a great need for the development of reliable and fast techniques for feature recognition on solar disks and their presentation in feature catalogues. These catalogues are intended to contain comprehensive statistics of active events (sunspots, active regions, flares, filaments), overlapping in a given period of time and to allow the extraction of physical characteristics which are essential for the solar activity forecast.

The complex investigation of active regions in conjunction with flares, sunspots, coronal mass ejections (CME's) and filaments is aimed at providing a reliable forecasts of solar activity and the space weather. Active regions are the basic reference features for solar activity. Their reliable automated detection will allow the building of a major database of solar active features and for the first time enable analysis of the solar activity on a comprehensive database of active regions taken in various wavelengths. The techniques are developed for automated detection of Active Regions (plages) using the H α and Ca KII solar images from the Meudon observatory, aiming to replace the existing manual detection methods. This research is done as part of the European Grid of Solar Observations (EGSO) project.

Three different approaches have been identified in the literature for the automatic identification of bright active regions (plage). The first approach based on the selection of a threshold to separate an object from the background which straightforward if the intensity histogram is bimodal but otherwise can be difficult (Steinegger et al, 1998). The Second approach based on the region growing techniques of segmenting solar images in a number of wavelengths, including H α , into bright and dark regions which have been used by a number of authors for a variety of purposes (Hill et al, 2001; Veronig et al,

2001). Finally the third approach using the Bayesian inference method for automatically identifying various surface structures on the Sun (Turmon et al, 1998). The Bayesian based methods were concluded to be very computationally expensive. Intensity threshold based methods are simple and fast but are very sensitive to noise which affects the reliability of the segmentation results obtained. As a result, in the current work region growing techniques are used in conjunction with intensity thresholds which are used in the initial processing to find seed location and in the subsequent active region segmentation. The method used and the results obtained for active region detection are described in section 2 and the conclusions are given in section 3.

2 The recognition techniques for active regions.

A flow chart of the segmentation procedure developed to detect active regions is shown in Figure 1. The presented technique first standardizes the full-disk high-resolution solar images using procedures that include limb fitting, shape correction and limb darkening removal which will convert them to a format that can be easily used by all subsequent algorithms (Zharkova V.V, 2002).

Because of this pre-processing the first assumptions made before started the active region recognition procedure are that the input images are free of radial limb darkening and of standardised size (1024 pixel $\times$ 1024 pixel) and solar disk radius (420 pixel) with solar centre (511.5 pixel $\times$ 511.5 pixel). The second assumption made is related to the active region characteristics where the active regions are the brightest features on the solar disk which means that the intensity values inside the detected regions of interest are greater than the intensity values on the background.

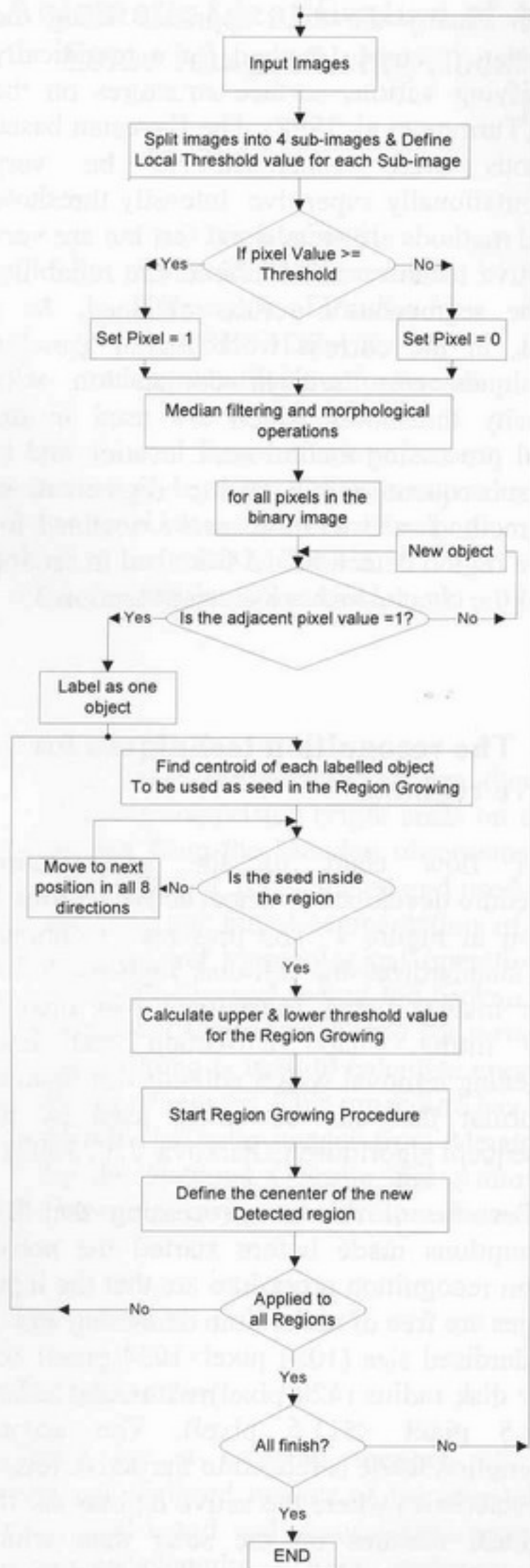


Figure 1: Flow chart of the used recognition technique for Active Regions detection

2.1 The Initial segmentation

The main stages of this technique using Ca KII 3 and H α Meudon full-disk images are illustrated in Figure 2 and Figure 3 respectively. Figures 2(a) and 3(a) are the cleaned initial images while Figures 2(b) and 3(b) shows the results of remapping these cleaned images to polar coordinates. After the remapping, the statistically derived local intensity threshold values are used for an initial segmentation of the bright plages by setting each pixel to 1 whose intensity value is over this intensity threshold. Otherwise, the pixel value is set to zero. The choice of these local intensity threshold values is very importation because a value that is too high may lead to real features being missed, whereas a value that is too low may lead to noisier binary images and hence spurious features. The optimum global threshold value also varies with the images brightness level and the non-radial large scale intensity variation which are the main problems in some Meudon images. To overcome this problem and to obtain the optimal local intensity threshold values (T), these values are calculated in iterations, for the quarter-sized regions of an image as follows:

$$T_1 = \mu_1 + (1 + \Delta_1) \times \sigma_1 \quad (1)$$

where μ_1 is the mean value for region one, σ_1 is the standard deviation for the same region and Δ_1 is a constant defining the range of values to look at, it is set to 0.4 after investigating more than 30 images. The results of this initial segmentation are shown in Figures 2(c) and 3(c).

2.2 The Median filtering, Morphological operations and Region labelling

The initial segmentation will include noise and unwanted small features caused by the

over-segmentation. The over-segmentation is preferable to under-segmentation as the former can be remedied using the Median filtering and Morphological operations while the latter could lose significant information. Firstly, in order to remove small features a 7×7 Median filter is used. The size was chosen through experimentation. Then the Morphological opening and closing operations are applied using a structure element of size 8×8 . This smoothes the features and fills in holes. Figures 2(d) and 3(d) show the detected regions after applying Median and Morphological processing and transformation back to Cartesian coordinates. As can be seen, the noise and over-segmentation problems have been completely remedied.

The result of this initial segmentation is a set of segments, each of which corresponds to an active region on the solar disk. Each segment is labelled, and its centroid is calculated for use as a seed in the region growing procedure. Prior to this, the intensity at the seed location is checked in order to ensure that it is inside the region and, if not, its position is adjusted. In this case a new seed is selected as illustrated in Figure 4 by investigating pixel values in the eight nearest neighbour directions in order until a new seed is found inside the region from which the region growing procedure can start.

2.3 The Region growing

Region growing is a basic procedure that starts with a set of seed pixels. The aim is to grow a uniform and connected region from each seed. A pixel is added to a growing region if and only if:

- It has not been assigned to another region
- It is a neighbour of the growing region
- A new region created by the addition of a new pixel is still uniform

The algorithm used for the region growing takes as input a clean image and a set of points from which the region-growing process starts. The algorithm begins at each starting pixel and scans the neighbouring 8 pixels in a circular fashion, to determine membership of the region around the central pixel that fall within specified constraints.

Two forms of constraint have been considered. The first uses a fixed threshold range (with an upper and lower pixel value) and second uses a variable threshold range set to a multiple of the standard deviation of the current region pixel values. After experimentation the fixed threshold range was chosen as it gives more accurate control in defining the outer boundaries of regions while also reducing the occurrence of holes in regions. Figure 6 shows the two detected active region grown by using the two types of constraint. Using upper and lower threshold values, the regions grown include all 8-connected neighbouring pixels that fall within the given threshold range. This is not the case using variable threshold range.

The upper and lower threshold values are determined by exploiting the statistical properties of the locally homogeneous background regions. As illustrated in the pseudo code below the lower threshold value is defined as $\mu - 0.3\sigma$ (where μ is the mean and σ is the standard deviation of that region) and the upper threshold is set to the maximum intensity value of the image. As pixels are added, the process repeats with the newly added pixels as the centre pixels. A list of tested pixels is maintained, in order to avoid unnecessary duplication of the tests. In this way, the region is constructed by using the membership criteria already discussed.

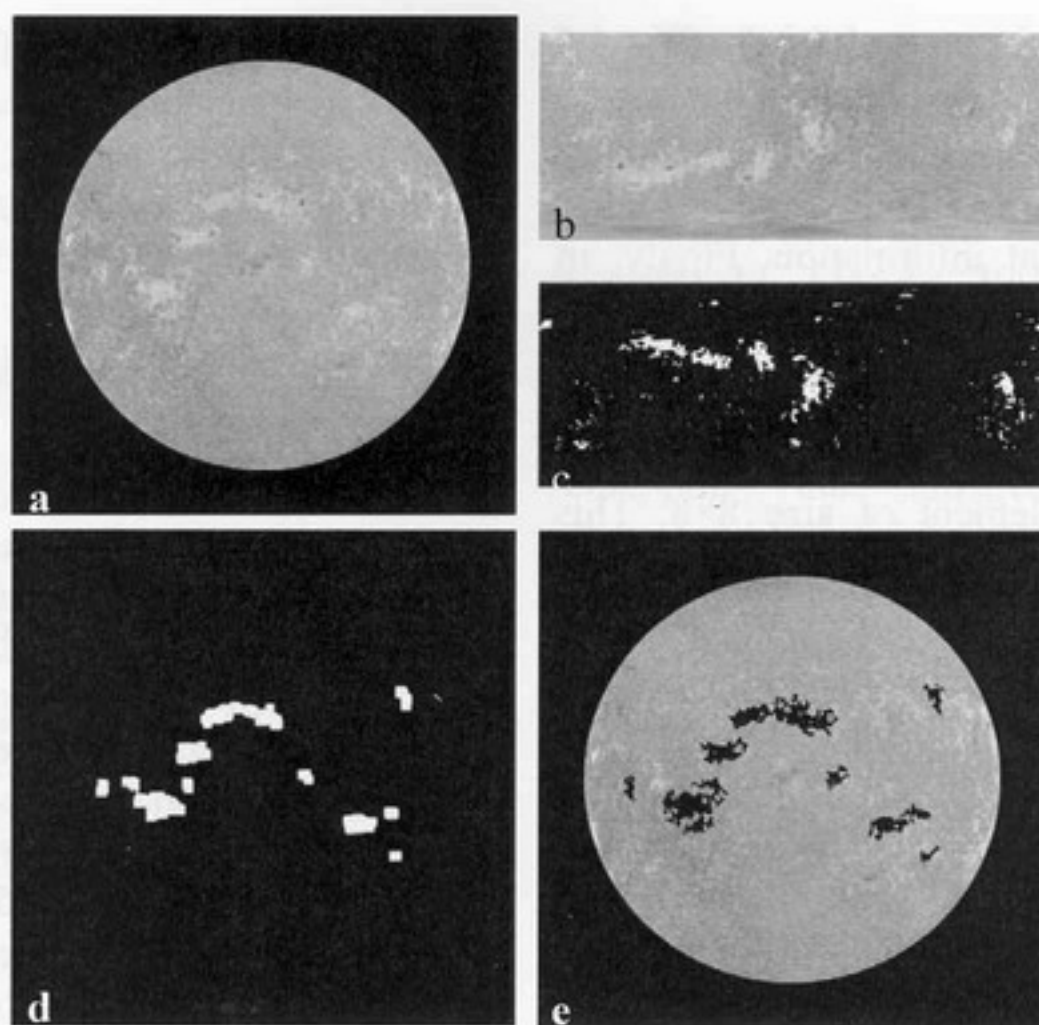


Figure 2: The segmentation procedure stages: a) an original Ca KII 3 image; b) after a transformation to Polar coordinates; c) after an initial thresholding; d) after a transformation back to the Cartesian coordinates, cleaning and morphological processing and e) a final result of the region growing.

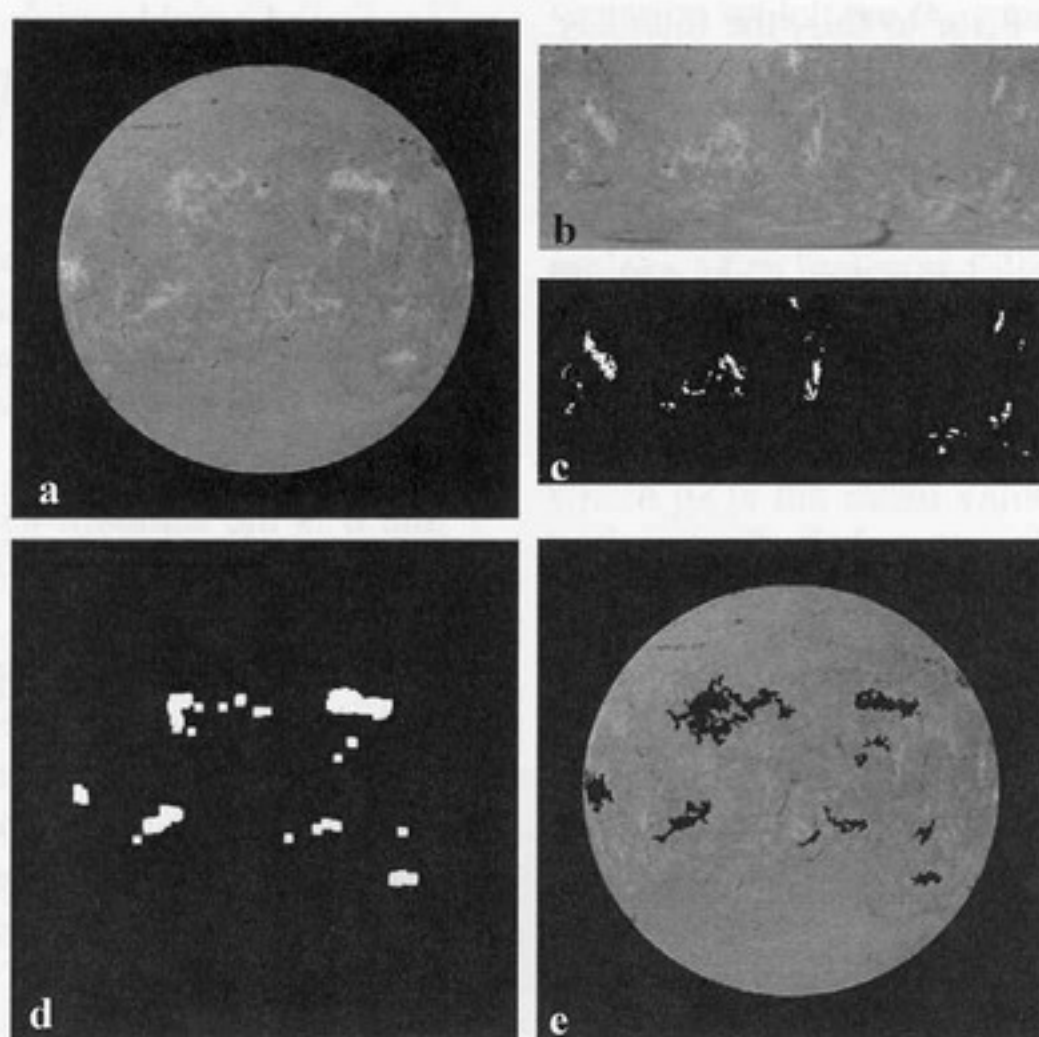


Figure 3: The segmentation procedure stages: a) an original H α image; b) after a transformation to Polar coordinates; c) after an initial thresholding; d) after a transformation back to the Cartesian coordinates, cleaning and morphological processing and e) a final result of the region growing.

The following pseudo code defines the region growing algorithm:

Let f be an image, and $R_1, R_2, \dots, R_n$ a set of regions each consisting of a single seed pixel.

```

Repeat
for i=1...n
  for each pixel  $p$  at the border of  $R_i$ 
    for all neighbours of  $p$ 
      Let  $x, y$  be the neighbour's coordinates
      Let  $lo_i$  be the lowest grey level of pixels in  $R_i$ 
       $lo_i = \mu_i - 0.3 \times \sigma_i$ 
      Let  $hi_i$  be the highest grey level of pixels in  $R_i$ 
       $hi_i = \max(f)$ 
      if neighbour unassigned and
        ( $hi_i \geq f(x, y) \geq (lo_i + \Delta)$ )
        Add neighbour to  $R_i$ , update  $hi_i$  and  $lo_i$ 
    Until no more pixels are being assigned to regions
  
```

If in the binary image there are more than one seed pixels obtained from an active region, the region growing method merges the detected pixels and forms a single active region area. Figures 2(e) and 3(e) show results of applying the region growing procedure.

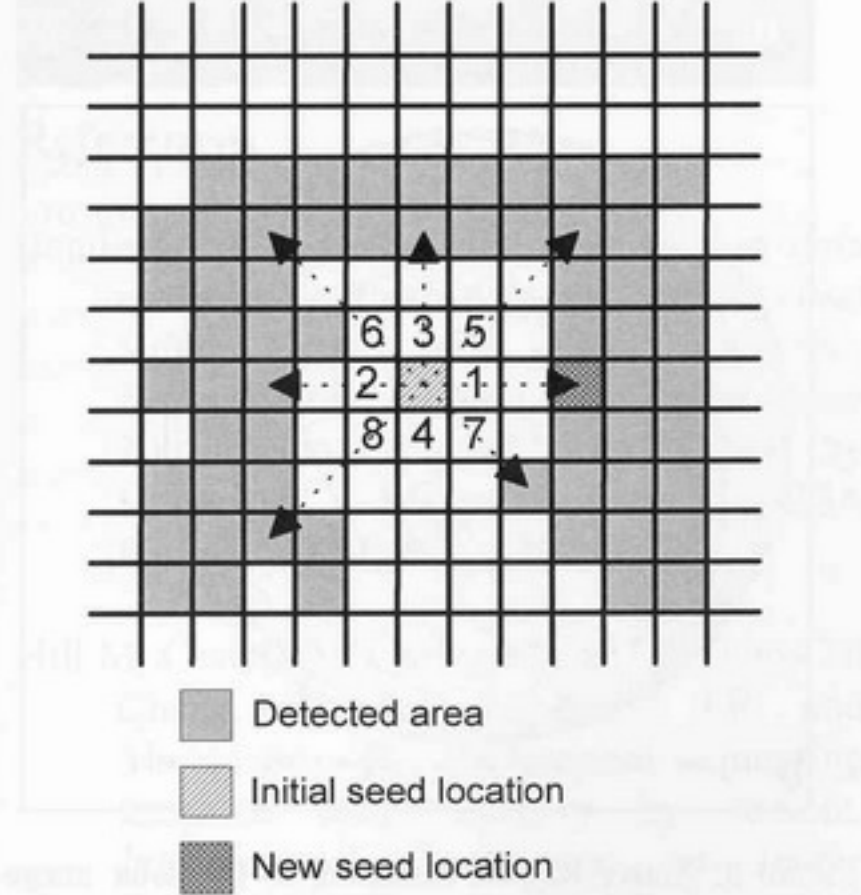


Figure 4: Seed pixel location

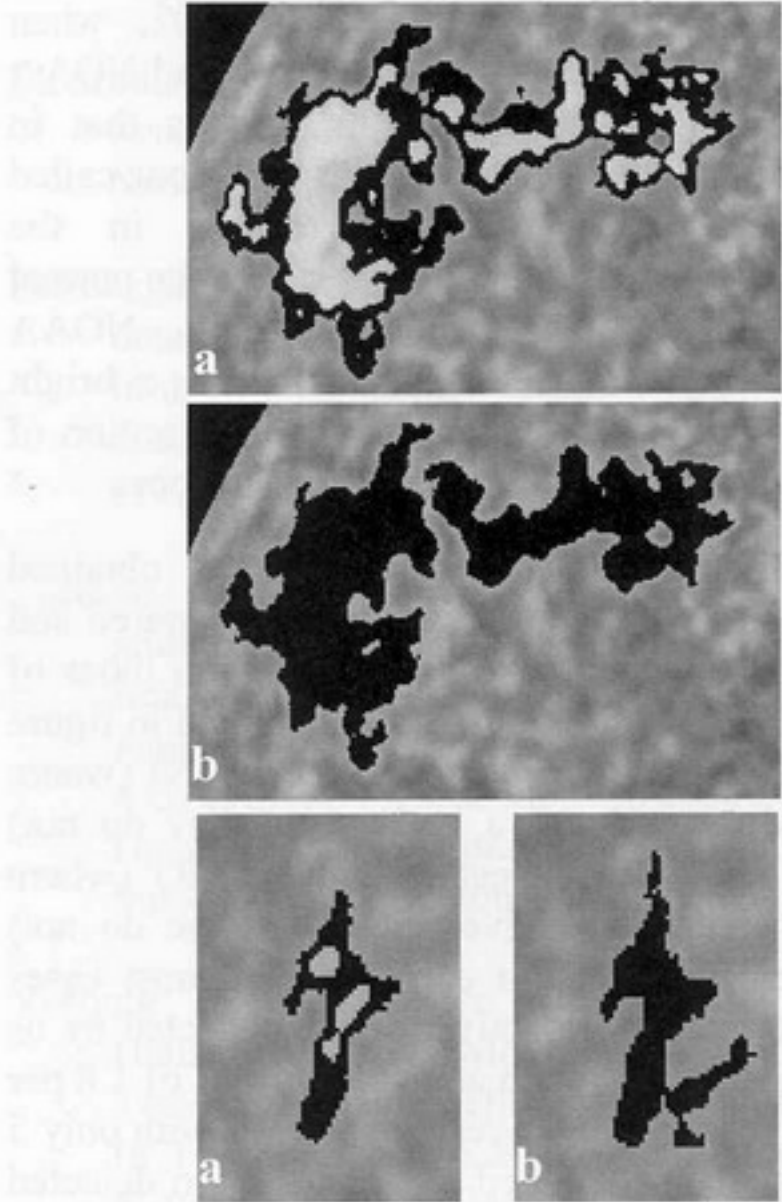


Figure 5: Results of region growing using the two forms of constraint: a) multiple of the standard deviation and b) threshold range

2.4 The accuracy of the technique

These procedures have been tested on the synoptic (28 days of April 2002) image sequences of full-disk solar images from the Meudon observatory. A comparison of the quantitative results obtained with our techniques with those done manually at the Meudon observatory (Mouradian Z., 1998) shows that in Meudon they detect 50% more active regions on most days. For example, on the 07/04/2002, there was 22 active regions in Meudon results while our procedure detected only 13 active regions and the NOAA observatory showed only 12 active regions. This was the case over the whole month (April 2002) except for 28/04/2002, when in Meudon there were detected 7, we detected 7 and

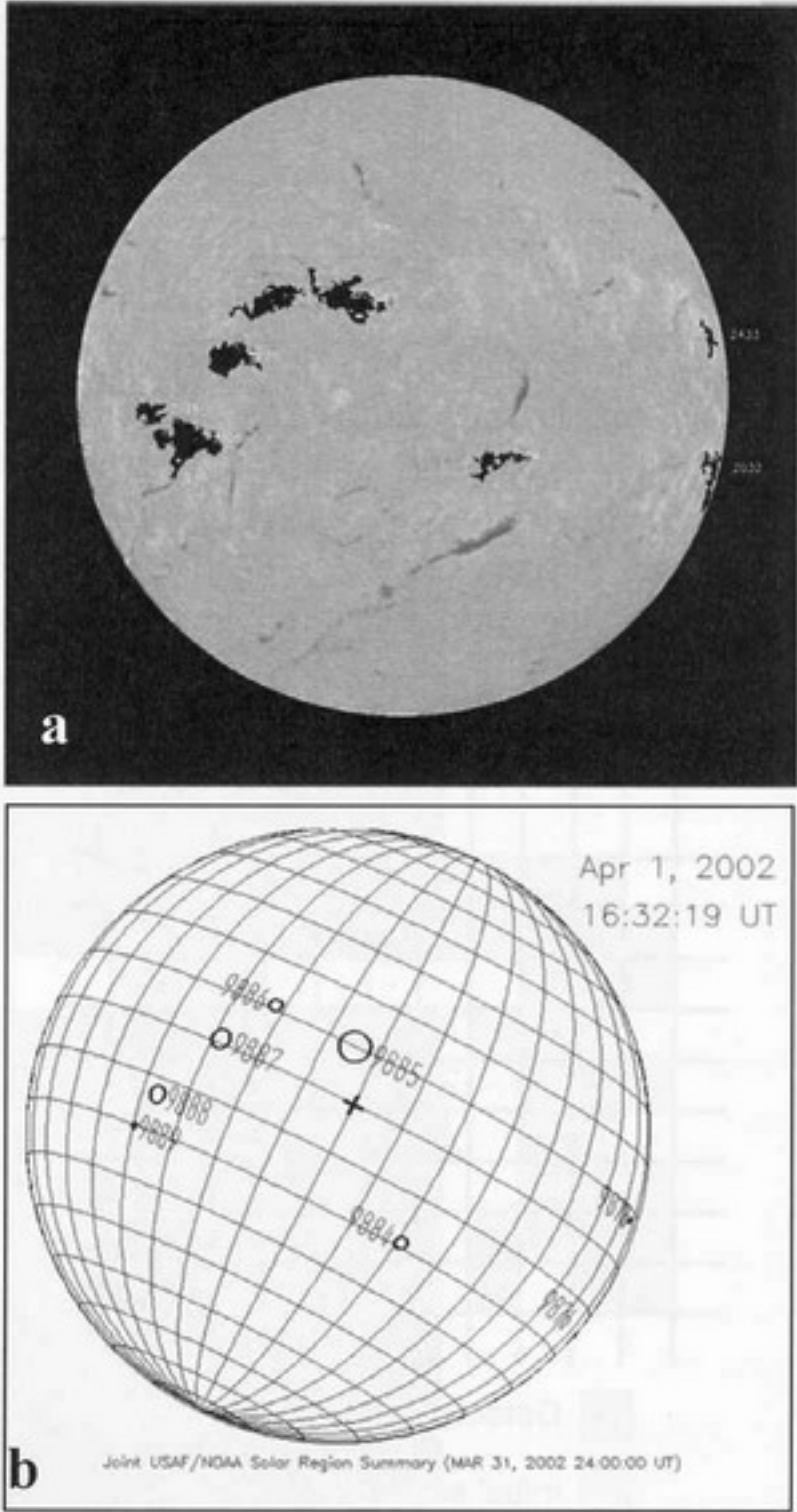
NOAA detected 8 and 29/04/2002, when Meudon detected 6, we detected 7 and NOAA detected 7. The reason for this is that in Meudon they detected the bright regions called plages, defined as the regions in the chromosphere that are brighter than the normal "quiet" Sun background, while NOAA detected the active region defined as a bright area on the Sun with a large concentration of magnetic field, often containing sunspots.

For further testing the results obtained using the Meudon images were compared and showed a reasonable agreement with those of the NOAA observatory (as illustrated in figure 6). The False Acceptance Rate (FAR) (where we detect an active region and they do not) and the False Rejection Rate (FRR) (where they detect an active region and we do not) were calculated on each day. In most cases there were more active regions detected by us than in NOAA with an average FAR of 1.8 per day. The FRR was very low, at 0.2 with only 5 days when we failed to detect region detected by NOAA. In some cases, and as shown in Figure 7, we detect an active region while NOAA splits it into two regions that does affect the quantitative comparison. In general, the agreement with NOAA is good, considering that NOAA bases its decisions on more information than we do in this stage

3 Conclusions

In this paper an efficient procedure of automated detection of solar active regions is presented. The statistically derived local intensity threshold values and region growing methods are used in the procedure. A local intensity threshold is applied to detect bright pixels that are considered to be the active region seed candidates. The subsequent region growing algorithm exploits statistical properties of the locally homogeneous

background regions in order to determine local upper and lower threshold values which control the spatial extent of the regions grown. The procedures used have demonstrated a satisfactory accuracy in the automated detection and segmentation of active regions from the Meudon H α and Ca KII 3 full-disk solar images. Future work is aimed at processing coronal EUV solar images from SOHO/EIT and building pseudo 3D structure for the active regions from these three wavelengths (H α , Ca KII 3 and SOHO/EIT).



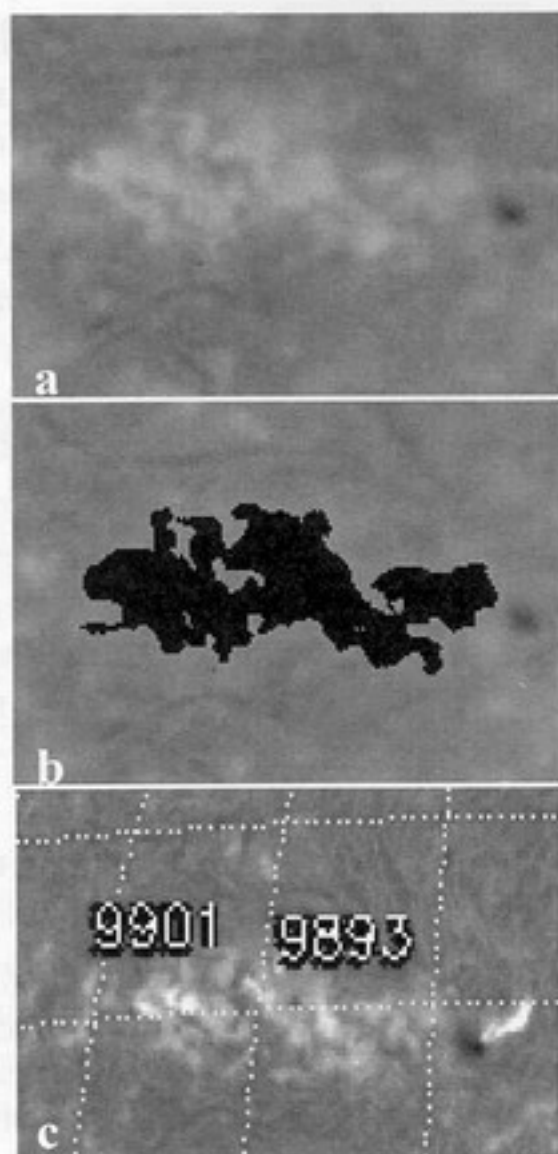


Figure 7: Two active regions merge: a) Meduon H α input image, b) the same image after applying region growing and c) Big Bear observatory (BBSO) H α image shown the same region as divided to two active regions

References

- Bentley, R.D. EGSO - the next step in data analysis. in *Proceedings of the Second Solar Cycle and Space Weather Euroconference*, 24 - 29 September 2001, Vico Equense, Italy, Edited by Huguette Sawaya-Lacoste, ESA Publication SP-477, 2002.
- Hill M, Castelli V, Chung-Sheng Li, Yuan-Chi Chang, Bergman L, Smith J.R. and Thompson B. Solarspire: querying temporal solar imagery by content. *International Conference on Image Processing*, Thessaloniki, Greece, 7-10 Oct. 2001, (1):834-837, 2001.
- Mouradian Z., Synoptic Data Findings, *Synoptic Solar Physics ASP Conferences Series*, (140):181-204, 1998.
- Steinegger, M and Brandt, P.N. On the determination of the quiet Sun centre-to-limb variation in Ca K spectroheliograms. *Solar Physic Journal*, (177):287-294, 1998.
- Turmon, M., Pap J. M., and Mukhtar, S. Automatically Finding Solar Active Regions Using Soho/Mdi Photograms And Magnetograms, *Proc. SoHO 6/GONG 98 Workshop*, Structure and Dynamics of the Interior of the Sun and Sun-like Stars, Boston, 1998.
- Veronig A., Steinegger M., Otruba W., Hanslmeier A., Messerotti M., and Temmer M. Automatic Image Processing In The Frame Of A Solar Flare Alerting System. *HOBUD7* 24(1):195-200, 2001.
- Zharkova V.V., Ipson S. S, Zharkov S. I, Benkhalil A. K and Aboudarham J. A full disk image standardisation of the synoptic solar observations at the Meudon observatory. *Accepted for publication in the Solar Physics Journal*. 2002.

An Automated Recognition of Sunspots on the Ca K1 and SOHO/MDI White Light Solar Images

Sergei Zharkov
Cybernetics Department
University of Bradford
BD7 1DP, UK
s.zharkov@bradford.ac.uk

Stan Ipson
Cybernetics Department
University of Bradford
BD7 1DP, UK
s.s.ipsen@bradford.ac.uk

Valentina Zharkova,
Cybernetics Department
University of Bradford
BD7 1DP, UK
v.v.zharkova@bradford.ac.uk

Ali Benkhalil
Cybernetics Department
University of Bradford
BD7 1DP, UK
a.benkhalil@bradford.ac.uk

Abstract

A new technique for an automatic identification of sunspots on CaII K1 line and "white light" full disk images is presented. The technique includes Sobel edge-detection, median filtering, morphological operations as well as an extraction of statistical and geometric properties of the detected objects for classification. This permits a robust recognition of the features on noisy images of the satellite or ground-based observations which are distorted either by a data transmission or weather conditions. This research forms a part of the solar feature recognition work package for the European Grid of Solar Observations (EGSO) project.

1. The Introduction

With a substantial increase of solar image databases an automated detection and verification of various features of interest is becoming increasingly important for a reliable forecast of the solar activity and space weather.

The solar images considered here have a "quiet Sun" background on which there are bright (faculae) and dark (sunspot) features superimposed. The sunspots are dark features in the solar photosphere and can be observed in the white light and CaII K1 line images. Visually, sunspots are the sites of strong magnetic field at the surface of the Sun. Sunspots consist of the two parts: a dark, roughly circular central disk called the umbra,

and a lighter outer area called the penumbra. Sunspots are best observed in "white light" images, while stronger spots can also be detected in CaII K1 line images due to the absorption in that line.

Sunspot identification is required for a quantitative study of the solar cycle and includes their locations, lifetimes, contrasts and other characteristics. It also plays an essential part in modelling of the total solar irradiance (TSI) and variations of sunspot properties with latitude and/or the solar cycle phase. Sunspots are also part of Active Regions, and their local behaviour is used for a study of the Active Region evolution and for the forecast of solar activity (Hathaway et al., 1993).

A number of techniques exist for sunspot area determination and umbra/penumbra separation. A brief history of this research is

provided in the paper by Steinegger et al. (1997) (see the references therein).

The ratio of the umbra-to-penumbral area depends on the spot size, but the study of its time variability is severely influenced by changes in the atmospheric seeing conditions. Some authors measured the umbral areas by superimposing on the dark umbrae, present on photographic plates, a quadrilateral consisting of two crosshairs. However, the accuracy of this method was not better than 30%.

In an analysis of full-disk white-light observations, a photometric method for area determination was used based on using a threshold estimated a priori for the sunspot intensity. A sunspot was defined to have a negative intensity contrast of -15% and simply counted all pixels beyond this value.

Some authors applied different photometric methods to manually determine the intensity boundaries between umbra, penumbra, and photosphere, individually for each spot. One consisted in taking the intensities half way up from the average penumbra intensity to the photosphere and half way down from the average penumbra intensity to the central part of the umbra. The other method was based on finding the extreme values of the derivative, i.e., the inflexion point, of the intensity profile of a one-dimensional scan across each sunspot. Steinegger et al. (1996) used the so-called difference histogram method to determine the intensity boundary between penumbra and photosphere. They define this value for each individual spot as the intensity at which the penumbral distribution falls to zero. Pettauer and Brandt (1997) have proposed a method to measure the umbral areas by determining for each particular sunspot the intensity value which defines the transition umbra-penumbra. It is based on the representation of the cumulative sum of the portions of sunspot area contained in successive brightness bins of the histogram.

However, these techniques in their original form were found to be not particularly well suited for a truly automatic detection and identification of sunspots. Some techniques rely on high resolution, while the technique suggested by Preminger et al (2001) requires clean images, something that cannot be guaranteed for ground-based observations susceptible to atmospheric conditions.

The compilation of the Zurich relative sunspot numbers is one of the most commonly used databases of solar activity. Beginning from 1981, the Zurich relative sunspot program was replaced by the Sunspot Index Data Center (SIDC) in Brussels, which is the World Data Center for Sunspot Indices (Hathaway et al., 1993).

In the current paper a new technique for an automatic identification of sunspots on the CaII K1 line and "white light" full disk images is presented. The initial objective of our research was to reliably identify sunspots on CaII K1 line images provided by Meudon Observatory with the possibility of extending the technique to data from other sources. Another application for this technique arose naturally from the fact that sunspots are used as a major indicator of solar activity and can replace the existing manual practices.

The presented technique allows the detection of sunspot locations, areas of umbra and penumbra with a high accuracy, the classification of sunspots and generation of Wolf numbers for every day, month and building the synoptic maps reflecting a sunspot evolution during every month of solar activity. The technique was developed as a part of the solar feature catalogue activity in the EC-funded EGSO project.

2. A Description of the Technique

2.1 The Observations and Data Reduction

There were two sets of solar full disk images used for this study. One was provided

by the Meudon Observatory, and the other was obtained from the SOHO/MDI data archive. Both sets cover the time period spanning April 1, 2002 to April, 30, 2002.

The CaII K1 line spectroheliograms from Meudon provide images of the solar photosphere (blue wing of the CaII K 3934 Å line, or K1). The data are acquired once a day by performing scans of the solar disk using entrance and exit slits. The images were pre-processed as described in Zharkova et al (2003). The FITS file headers were checked and amended where necessary. Images were resized to a standard size 1024 by 1024 pixels. The disk centre was set to be the centre of the image, while the disk radius was set to 420 pixels. Intensities of all pixels outside of the disk were set to zero. The general quality of these images is highly dependent on atmospheric conditions on the day of the observation. Also, a number of instrumental artefacts such as dust lines are often present. The original image data is 16 bit, with 12 bits actually used.

The SOHO/MDI provides almost continuous observations of the Sun in the white light continuum in the vicinity of the Ni I 6767 ? line with resolution comparable to the ground-based telescopes. SOHO satellite is not a subject to clouds, though there are dropouts from time to time due to spacecraft problems. For the SOHO/MDI intensity spectrograms the pre-processing stage was as follows. For the pixels outside of the solar disk, as specified by the FITS file header information, the intensity values were set to zero. The disk intensity was flattened by removing the limb darkening curve.

2.2 The Methods of the Automatic Sunspot Detection

Let us introduce main definitions used in the recognition. A contrast image is a pre-processed image with limb darkening removed.

Solar image consist of a “quiet sun” with bright and dark features superimposed, the histogram of the contrast image will have (possibly a continuous range of) the intensity values with peak counts corresponding to the quiet sun, i.e. the parts of the solar image not containing any features. (see Figure 1). The contrast value of a pixel is a ratio of the pixel’s intensity to quiet sun intensity.

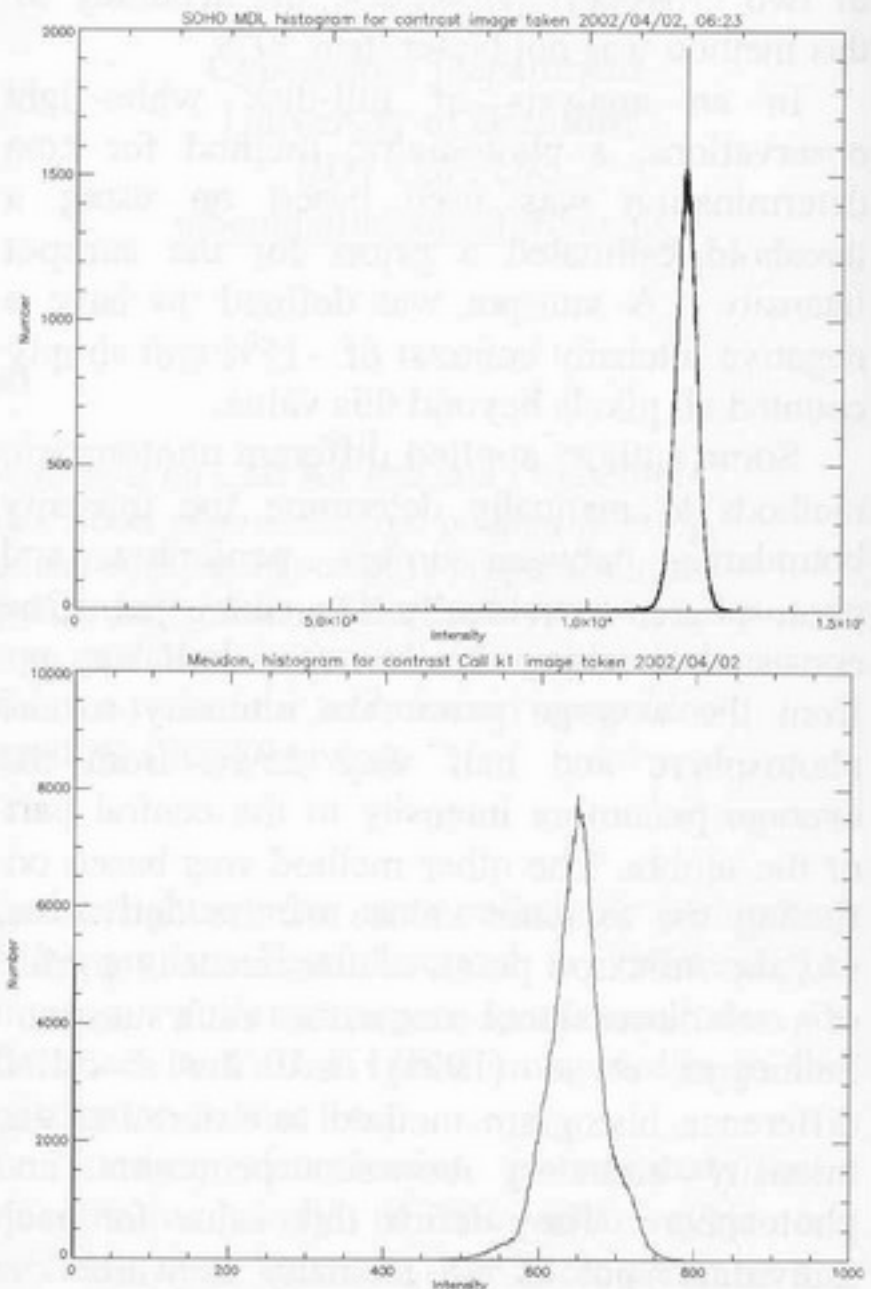


Figure1. Contrast image histograms for SOHO/MDI white light and Meudon Observatory CaII K1 line observations.

The first attempts were aimed at implementing the region growing technique described by Preminger et al, 2001: A candidate feature is identified by scanning a solar disk and finding a pixel whose contrast is greater than a given contrast criterion. Three contiguous pixels that meet the trigger criterion are considered to be a part of a feature (sunspot) and used as a seed for region growing. Subsequent adjacent pixels

must meet another, generally less stringent, contrast criterion, called the *threshold*, in order to be considered a part of the feature. Thus, all the contiguous pixels with the contrasts exceeding the threshold, including at least the three contiguous pixels with contrasts exceeding the trigger, are identified as a single feature.

By analysing the contrast image histogram (Figure 1), a quiet Sun intensity is determined. The image is then thresholded to a ratio of the quiet sun intensity, and the median filter is applied. The remaining non-zero pixels are used as seeds for region growing with the lower threshold set to zero and the upper one set to a chosen proportion of the quiet Sun intensity.

The technique performs reasonably well on clean images, particularly on the SOHO/MDI intensity spectrograms, consistently detecting the sunspots with well-defined umbral area. The ratio for threshold was determined empirically for each set of images. By increasing the ratio one can allow detection of smaller sunspots, although this leads to a much higher probability of false identifications. However, due to heavy atmospheric influence, the technique in this form is virtually unusable on the Meudon images as it gives a large number of false identifications and extracts erroneous information for the correctly identified features.

For this reason other possibilities for determination of threshold values were studied. In particular, the difference histogram techniques and cumulative histogram methods were considered as described by Pettauer, Brandt (1997) and Steinegger et al. (1997). Both methods were not designed to work on full disk images but rather on high resolution image fragments containing sunspots. They are also quite sensitive to blurring which often occurs in the ground-based observations due to atmospheric conditions. Thus a new technique for sunspot detection on full disk images had to be developed for our purposes.

In order to avoid dependency on a choice of global threshold, the edge detection is used instead. Good results were achieved using either the morphological gradient or Sobel operators. On the edge map image a global threshold is still have to be used, but it is significantly less sensitive. After removing the edge corresponding to the disk limb and filtering out the smaller artefacts by means of median transform and morphological closure, the watershed and dilation operators are used to define the regions of interest as those possibly containing the sunspots. Each region is then examined separately, and the region growing technique described by Preminger et al (2001) is used. The values of thresholds are determined as a function of full disk quiet Sun intensity value and statistical properties of the region (such as a mean intensity, standard deviation and absolute mean deviation).

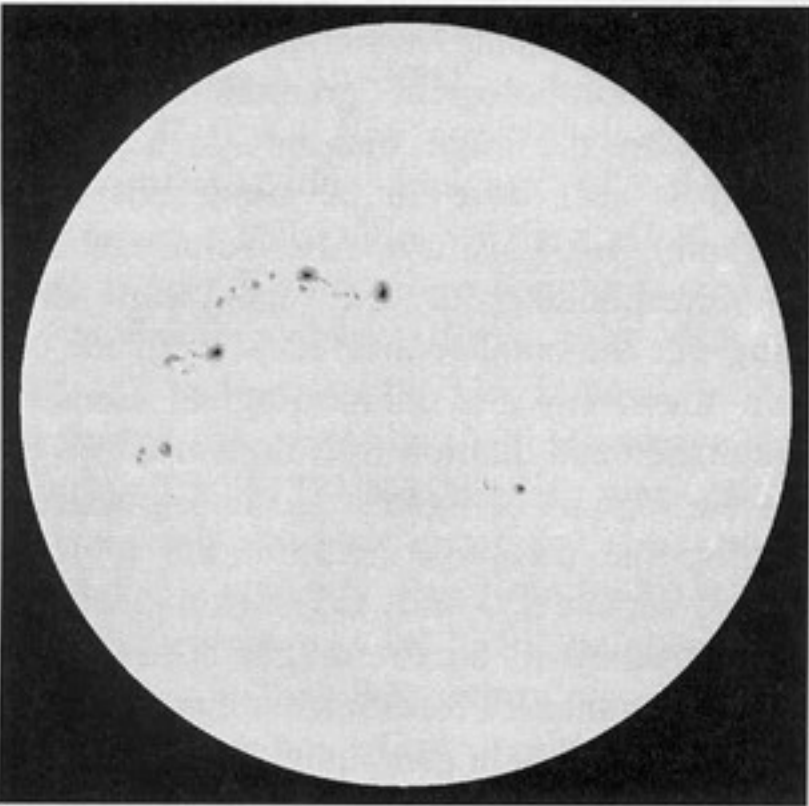
Alternatively, for each contour defined during the edge detection stage to be located within the region under examination, its mean, median and minimum intensities are determined and used as threshold values for the seed location and upper limit during the region growing procedure.

2.2.1 *The Automatic Detection on the SOHO/MDI white light images.*

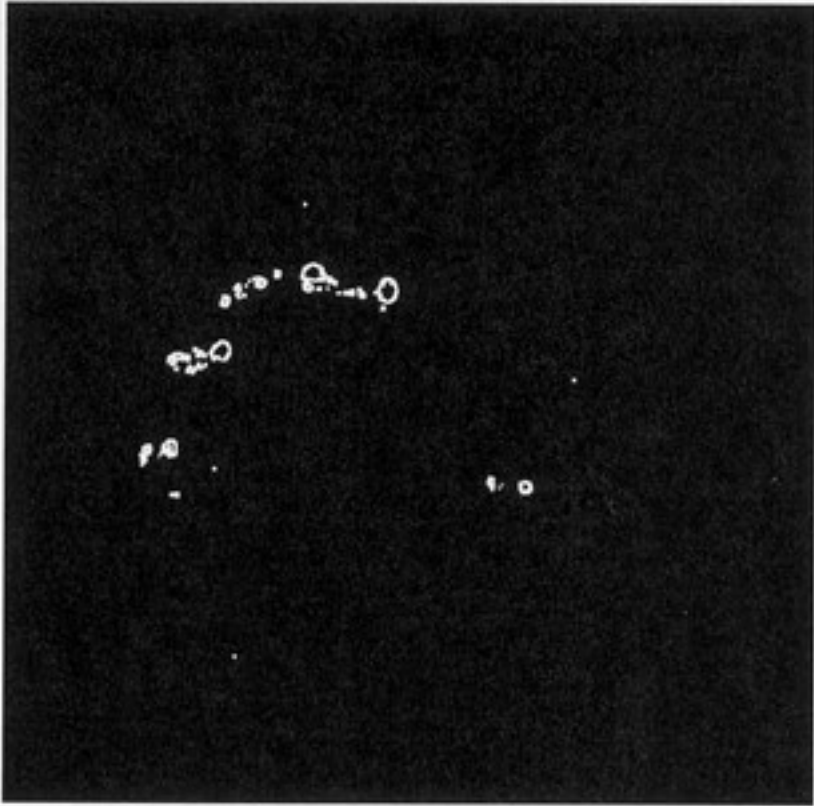
This technique developed for SOHO/MDI data relies on a good quality and continuity of the images. It results in a number of parameters to be set constant for the whole data set including a percentage of top and bottom values, deducted from the histogram equalisation method, and various thresholds.

1. First the image is pre-processed, the necessary information such as disk radius, disk centre etc is extracted from the header and the limb darkening is removed. Thus a contrast image is created.

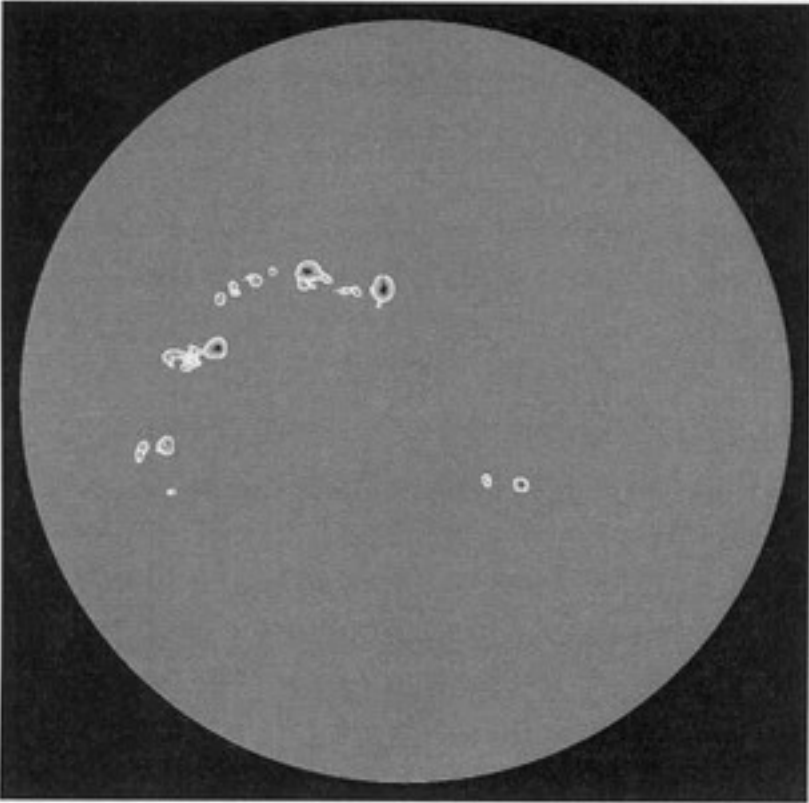
Figure 2. Sunspot Detection on the White Light full disk images obtained from SOHO/MDI



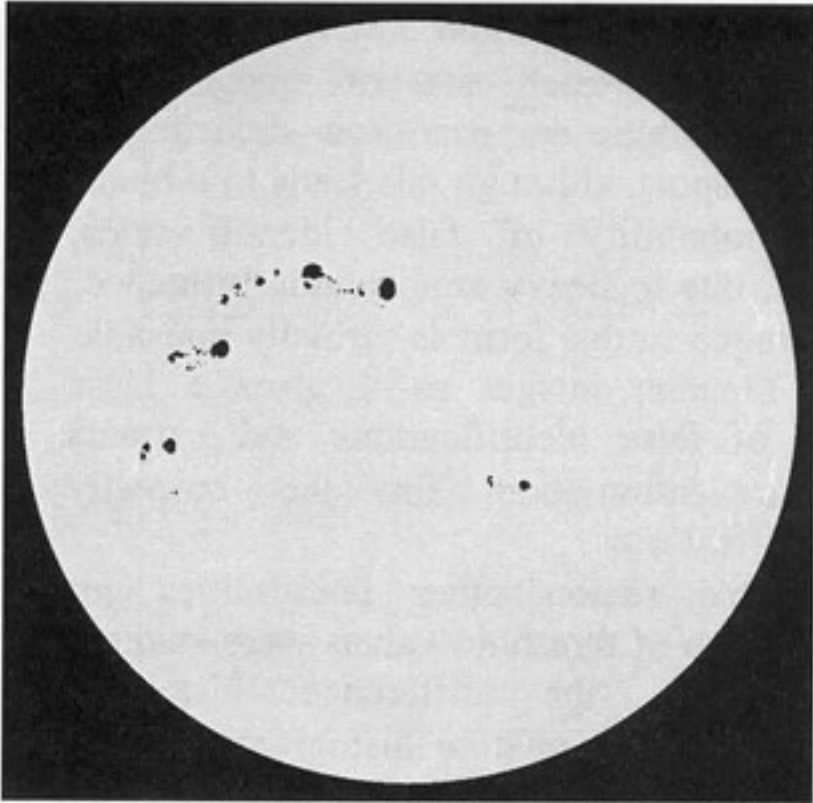
a) The original image



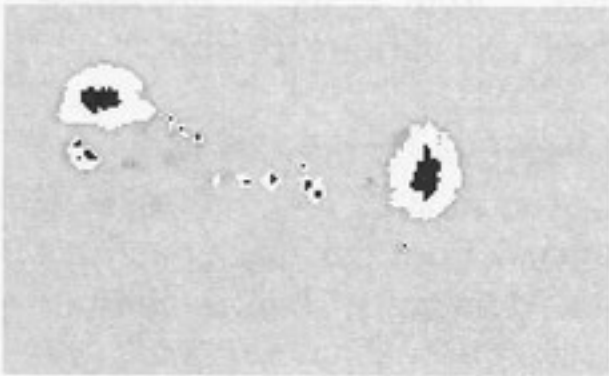
b) The detected edges



c) The regions of interests after filtering



d) The detection results



e) The close-up of sunspot group, with umbra pixels marked black and penumbra in white

2. The histogram equalisation is applied to amplify the edges.

3. A morphological gradient operator is applied to equalised an image. By thresholding and removing the limb edge we obtain a binary map containing sunspot and faculae contours and, possibly, some noise.

4. By applying the morphological closure and watershed operators we close the incomplete contours and fill the enclosed area and combine them together. The resulting map is then median-filtered in order to remove smaller structures.

5. Then dilation is used to define the regions of interest which are to be examined for a presence of sunspots.

6. Each region is examined in the following manner: the seeds are located by examining the pixels with intensities lower than the region's mean minus a quarter of the absolute mean deviation. There are, may be, several connected groups of the pixels in this region with this property; in each group the pixel with minimum intensity is chosen as a seed. Then the region growing operation is performed with the upper threshold equal to 80% of the 'quiet Sun' intensity value or the mean of the region, whichever is greater. The resulting region(s) is (are) sunspot or sunspot pores.

7. The results are combined into a binary version of the original contrast image with the detection results marked.

8. Then the umbra-penumbra separation is performed as follows: for sunspots bigger than five pixels, if it contains the pixels with intensities lower than 70 per cent of quiet sun intensity value, then the umbra is set to include the pixels with intensities lower than the sunspot mean intensity minus standard deviation (if there are any); for small sunspots (sunspot pores) the umbral threshold is set to 60 per cent of quiet sun intensity.

Some results of the application of the above technique are presented in Figure 2. SOHO/MDI also provides magnetograms taken practically simultaneously with the white light images. These are then used to determine the magnetic polarity of each sunspot and to combine them into the groups corresponding to the active regions.

2.2.2 Sunspot Automatic Detection on the CaII K1 line images.

The technique described above with some modifications can be also applied to CaII K1 images. Some results of sunspot detection with this technique are presented in Figure 3. First, a full disk high-resolution solar image is pre-processed (Figure 3a) by correcting, if necessary, the shape of the disk to a circular one and by removing the limb-darkening (Zharkova et al., 2003). Then a morphological gradient is applied to the edge enhanced processed image, followed by a thresholding applied in order to detect strong edges (Figure 3b). After removing the limb edge, a watershed operator is used on the binary image in order to fill the sunspot area enclosed by the edges. Further median filtering is used to get rid of noise and smaller features (Figure 3c). The regions' statistical properties are then used for a removal of false identifications such as the artefacts and lines, often present in the Meudon Observatory images. For the extraction of the area, shape, umbra/penumbra location of the detected sunspots and their basic classification the region growing, local contrast and contiguity techniques are used (Figures 3d and 3e).

As it relies on global equalisation this technique is not stable and can lead to erroneous results on images with certain kinds

Table 1. The accuracy of sunspot detection and classification for CaII K1 (see the text for description)

Date	#Sunspots Ideally	Visible	# Detected	#Detected mr=3	#Detected mr=1	#Detected mr=0
01/04/02	16	16	16	16	17(1 false)	19 (3 false)
02/04/02	21	20	19	19	19	19
03/04/02	15	14	15(1 false)	15(1 false)	15(1 false)	24 (9 false)
04/04/02	14	13	15(1 false)	16(2 false)	18(4 false)	20 (6 false)
05/04/02	17	16	18(1 false)	19(2 false)	19(2 false)	19(2 false)
06/04/02	12	10	10	10	10	15 (5 false)
07/04/02	11	11	9	13 (3 false)	13(3 false)	13(3 false)
08/04/02	15	14	15(1 false)	17 (2 false)	20 (5 false)	22(7 false)
09/04/02	17	16	17	17	17	17
10/04/02	14	12	12	12	13	14 (1 false)
11/04/02	15	12	9	9	9	10 (1 false)
12/04/02	18	18	20 (2 false)	20	21 (3 false)	21 (3 false)
14/04/02	21	20	22 (1 false)	26 (5 false)	27 (6 false)	34
15/04/02	16	13	15	16	16	18
16/04/02	11	10	12 (2 false)	14 (3 false)	14 (3 false)	19(6 false)
17/04/02	12	11	12 (1 false)	12 (1 false)	13 (1 false)	13 (1 false)
18/04/02	12	12	12	13(1 false)	13 (1 false)	18 (6 false)
18/04/02	11	11	10	11	11	12 (1 false)
19/04/02	14	13	14	14	14	15 (1 false)
20/04/02	11	9	10	10	10	11 (1 false)
21/04/02	9	8	8 (1 false)	11 (3 false)	11 (3 false)	12 (3 false)
22/04/02	12	12	13 (1 false)	13 (1 false)	15 (3 false)	15 (3 false)

of noise. Thus we have attempted to develop a more reliable method as described below.

This is the list of steps implemented in the presented technique.

1. First the image is pre-processed, the necessary information such as disk radius, disk centre etc. is extracted from the header and the limb darkening is removed. Thus a contrast image is created.

2. Gaussian smoothing is applied followed by Sobel edge detector. The edge map is then thresholded at an appropriate level determined from edge map histogram.

3. On the resulting binary files containing the edges of the features and noise on the image, the edge corresponding to the limb is removed.

4. Median filtering is applied to discard smaller structures and then dilation used to define the regions of interest. These regions are likely to contain sunspots or groups of sunspots. It is possible to have one edge encompassing several sunspots. These have to be separated. It is also likely that there are weaker sunspots that do not generate a sufficiently strong edge to be detected. Since sunspots appear in groups, we use the

morphological dilation to define the region of interest to be examined for a sunspot presence.

5. For each region of interest, we identify the edges present in the neighbourhood, and find the intensity values corresponding to these contours in the original image. The average intensity value is then calculated and used as a threshold for the region. In this way we are able to separate sunspots corresponding to a join edge as well as to detect other sunspots present in the neighbourhood.

6. The results are then combined into a Sunspot Candidate map consisting of a binary version of the original contrast image with the detection results marked.

7. Each sunspot candidate is examined and classified as a spot, spot with an umbra, noise and several other categories between spot and noise (a classification is in progress). At the moment the classification relies on the following parameters: quiet sun intensity, size, mean deviation, absolute mean deviation and principal axis ratio.

This technique performs well on noisy images as it does not heavily rely on global image statistics. This comes however at a price, as there are fewer potential sunspots detected compared to the technique using histogram equalisation. Some results of the application of the above technique are presented in Figure 4.

2.2.3 Verification and Accuracy

The technique was tested on one month's (April 2002) CaII K1 line observations from the Meudon Observatory and demonstrated a good correlation with the synoptic maps manually produced manually at the Observatory. The technique was also applied to the same month's observations from the SOHO/MDI white light images and confirmed the sunspot detection from the Meudon observations. Since sunspots are also characterised by a strong unipolar magnetic fields, the identification results are compared with the SOHO/MDI magnetograms

obtained for the corresponding time period and showed excellent agreement between the identified sunspot areas and observed magnetic features. In addition, the magnetogram data is used for a more detailed classification of sunspots i.e. separating them into the groups corresponding to the active region and adding a positive or negative polarity tag. The implementation of this technique in Interactive Data Language (IDL) can process 10-15 1024 by 1024 pixel images per minute on Intel 1.6GHz processor.

The application of the technique to the Meudon ground-based CaII K1 observations revealed good detection accuracy in comparison with the observations done manually (see Table 1). In Table 1, The "Number of Sunspots Ideally" is deduced by comparing the Meudon CaII K1 images with the SOHO/MDI white light intensitygrams with all possible candidates for sunspots in CaII K1 checked against the satellite data. The "Visible" corresponds to a number of sunspots that can be identified by a trained observer on a CaII K1 image.

The "Detected" represents a number of sunspot candidates detected by the algorithm with $mr=x$ indicating the number of regions classified as a sunspot depending on the classifier settings. Namely, the column corresponding to $mr=0$, shows a total number of candidate sunspots detected. The $mr=1$ column, shows a number of candidate sunspots with the following properties: mean intensity is less than the quiet Sun intensity, principal axis coefficient is less than 3.5 and the mean absolute deviation is greater than 5. The $mr=3$, shows a number of candidate sunspots with the principal axis coefficient less than 2.1 and mean absolute deviation greater than 15. The $mr=5$ column shows a number of candidate sunspots with principal axis coefficient less than 2.1 and mean absolute deviation greater than 21. This classifier is only approximate and does not perform reliably on blurred images.

The technique for sunspot detection in the Meudon CaII K1 images performs well, particularly, in comparison with other earlier methods (Preminger et al, 2001). Based only on the information represented in the image all sunspots are detected in most of the images. Erroneous identifications and missed sunspots are not unusual in manual detection. However, this is dealt with at the next stage of combining the data into the synoptic maps: the adjacent observations are compared and the most accurate information is then used. We are now proceeding to this stage in collaboration with the Paris-Meudon Observatory.

3. The conclusions

The automated techniques for sunspot detection capable of processing large numbers of white light and Ca II K1 line images were developed. The techniques allow us to detect sunspots in CaII K1 images from Meudon Observatory with good accuracy, to combine them into groups, and to extract their spatial location and area in order to store them in a feature database. These techniques also allow to build the synoptic maps (currently done manually) and to extract the data such as sunspot numbers, sunspot area etc.

We are now able to detect sunspots with excellent accuracy on SOHO/MDI white light images, combine them into groups using information extracted from SOHO/MDI magnetograms, extract sunspot locations, umbral/penumbral areas. We have reached our objectives within the EGSO project with respect to these data sets. Further work will be carried out to produce automatic sunspot detection in Meudon Observatory CaII K1 images.

4. Acknowledgements

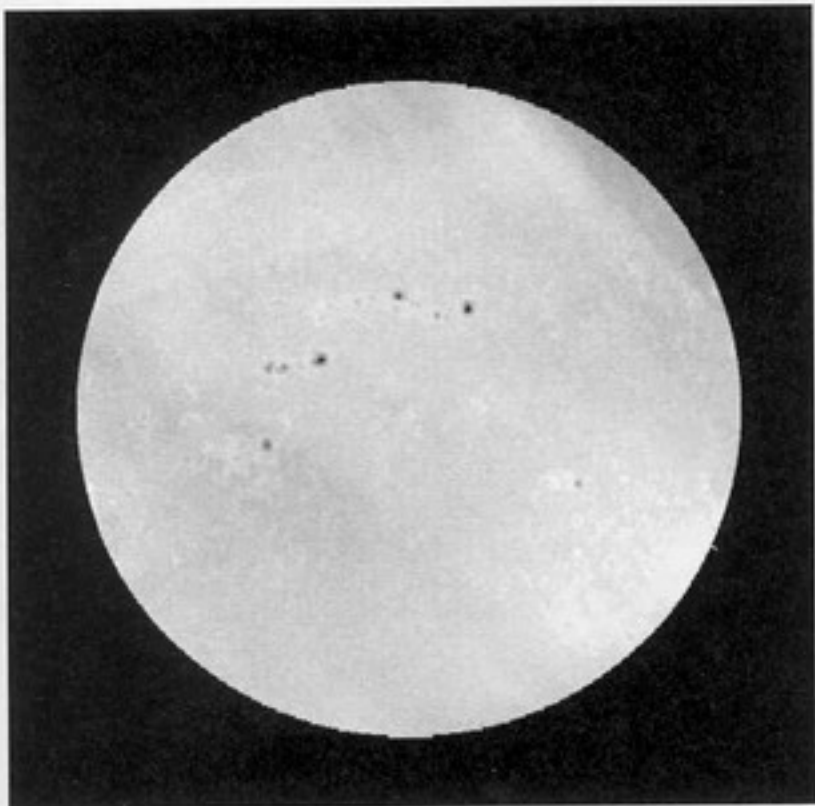
The research has been done for European Grid of Solar Observations (EGSO) funded by the

European Commission within the IST Fifth Framework, grant IST-2001-32409.

5. References

- Steinegger, M., Bonet J., A., Vazquez, M.: Simulation of seeing influences on the photometric determination of sunspot areas. *Solar Physics*, **171**, 303 (1997)
- Steinegger, M., Vazquez, M., Bonet, J.A. and Brandt, P.N.: On the energy balance of Solar Active Region, *Astrophys. J.*, **461**, 478 (1996)
- Preminger, D. G., Walton, S.R., Chapman, G.A.: Solar Feature Identification using Contrast and Contiguity, *Solar Phys.*, **202**, 53 (2001)
- Hathaway, D. H., Wilson, R.M., Reichman E.J.: "The Shape of the Sunspot Cycle", *Solar Physics*, **151**: 177-190 (1993)
- Pettauer, T., Brandt, P.N.: On novel methods to determine areas of sunspots from photoheliograms, *Solar Phys.*, **175**, 197 (1997)
- Zharkova, V.V., Ipson, S. S., Zharkov, S.I., Benkhalil, A., Aboudarham, J., Bentley, R.D., A full disk image standardisation of the synoptic solar observations at the Meudon Observatory, *Solar Phys.*, accepted (2003)

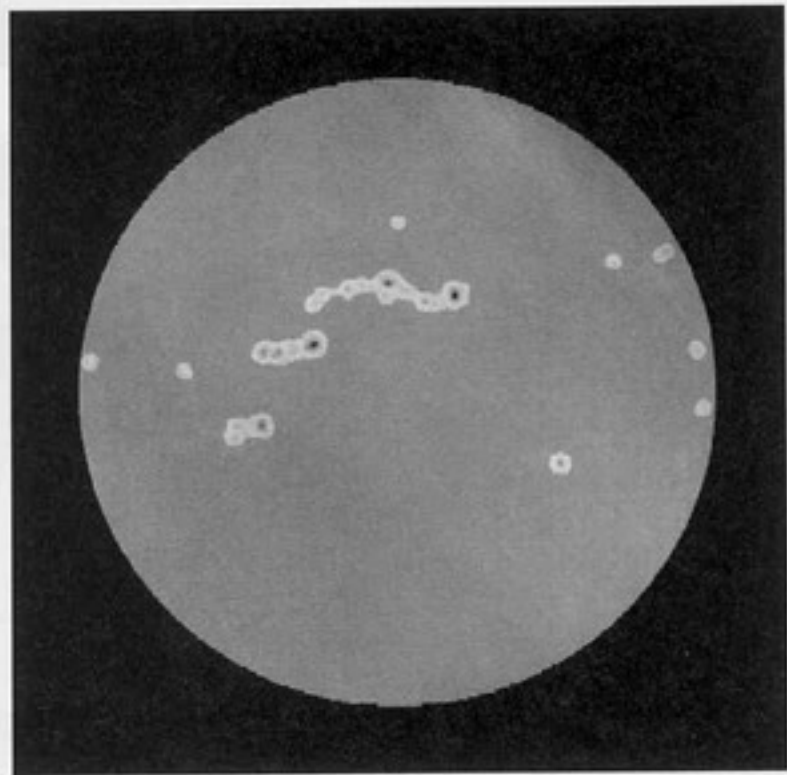
Figure 3. Sunspot Detection on the CaII K1 line full disk images obtained from Meudon Observatory



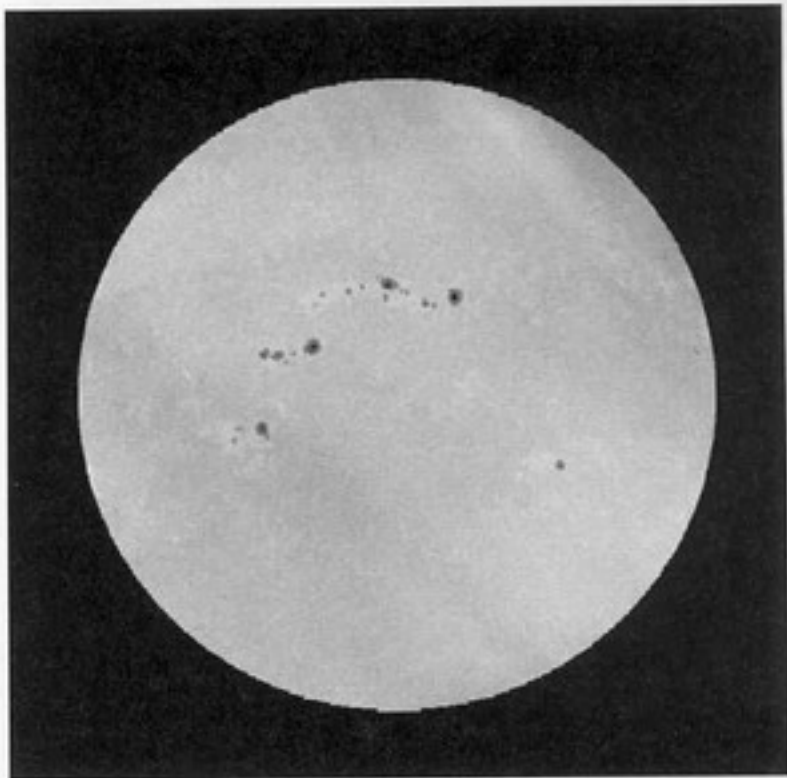
a) The original image cleaned



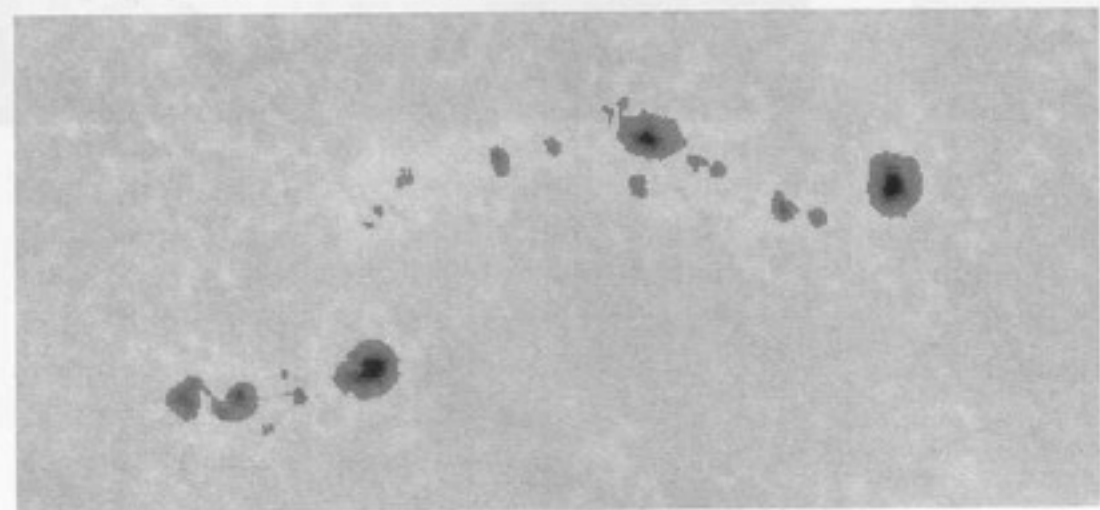
b) The detected edges



c) The found regions (dilated) superimposed on original image

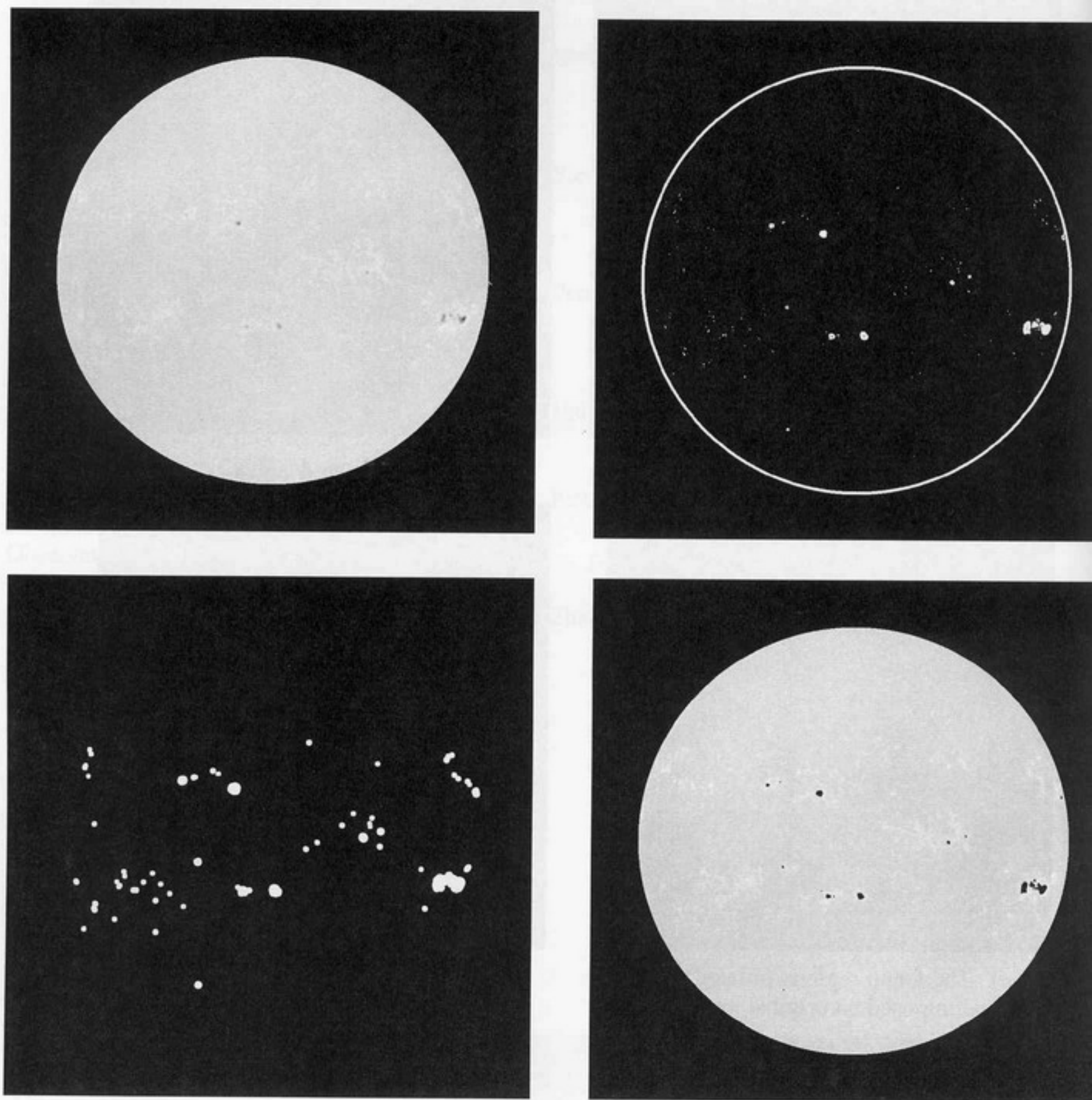


d) The final detection results



e) The extract from d)

Figure 4. Stages of Sunspot Detection on the CaII k1 line full disk images obtained from Meudon Observatory (top left - the original image, top right – Sobel edge detection results, bottom left – the regions of interest and bottom right – the sunspots detected).



Object Localisation and Tracking Through Subsymbolic Classification

Hugo Vieira Neto and Ulrich Nehmzow

*Department of Computer Science
University of Essex
Wivenhoe Park
Colchester, Essex CO4 3SQ, UK
{hvieir, udfn}@essex.ac.uk

Abstract

This paper presents results of experiments in subsymbolic processing of visual data, to achieve identification and tracking of *arbitrary* objects, which are intended to be used in autonomous robots for novelty detection and navigation purposes.

Artificial neural networks with unsupervised training are used as the classification stage for the vision system, in order to provide the robot the ability to develop its own representations from perceptual data without the need of any external human-provided information. We present an evaluation of the behaviour of the system when using very simple feature extraction techniques, such as horizontal and vertical average histograms, as well as average coarse coding.

1 Introduction

Animals can rapidly detect novelties in their normal environment using different sources of perceptual information. As it is relevant for survival, this particular ability to be aware of environmental changes is very important, for example, to find food or avoid predators.

The sense of vision plays an essential role in animal behaviour either for tracking movements or identifying objects of interest. It follows that novelty detection using artificial vision can be of equal importance to autonomous mobile robots. If such an ability is feasible to be implemented artificially, possible applications are in automated inspection and surveillance tasks.

Previous work done by Marsland et al. (Marsland et al., 2000, 2002) has given successful results in novelty detection, using habituation and sonar data in real robots. Habituation is a reduction in responses to stimuli that are repeatedly presented to the system and therefore can be used to highlight new perceptions.

The ultimate aim of the present research is to achieve a novelty detection system to be used in autonomous mobile robots using the known model of habituation and artificial vision as the main perceptual input.

Dealing with visual information is often a difficult task, because of the large amounts of data involved and the processing power required. As a consequence, it is necessary to arrange some sort of internal representation through models of the aspects of interest in the environment. These models, which are simplified abstractions of the original aspects, are expected to describe relevant features and eliminate unnecessary details.

However, "relevant features" are not always clear to the designer of such a system, nor what constitutes "unnec-

essary details". Nehmzow (Nehmzow, 1999) argues that in this case a method of model acquisition through robot learning is a more feasible solution, rather than explicitly supplying *a priori* models.

In this first stage of our research we are looking for suitable ways to extract relevant features from raw image data in order to manage processing in reasonable time and storage size, which are especially important for autonomously operating robots that have to respond to stimuli in real-time.

We are particularly interested in implementing an attention mechanism to be used in an active vision approach to localise and physically track objects of interest, as part of the whole novelty detection system to be developed.

2 Experiments

2.1 Objective

Our system was designed to determine where an arbitrary target object was located within an image frame. The target was provided by the user and its location was determined by the system as one of 25 non-overlapped sub-images of the frame (see Figure 3).

2.2 Experimental Setup

Our experiments were conducted using a Magellan Pro mobile robot, which is shown in Figure 1. The robot is equipped with a colour CCD camera, pan-tilt unit and framegrabber board, which are able to acquire RGB images with 160×120 pixels in size. Images were acquired while the robot was not in motion.



Figure 1: The Magellan Pro mobile robot

2.2.1 System Architecture

The functional blocks of our approach towards a system that can be trained to localise and track a small set of objects is given in Figure 2. Two very simple and computationally inexpensive image coding techniques were selected to be evaluated as the pre-processing stage.

As the classification stage, two different connectionist approaches with unsupervised training were chosen to be compared.

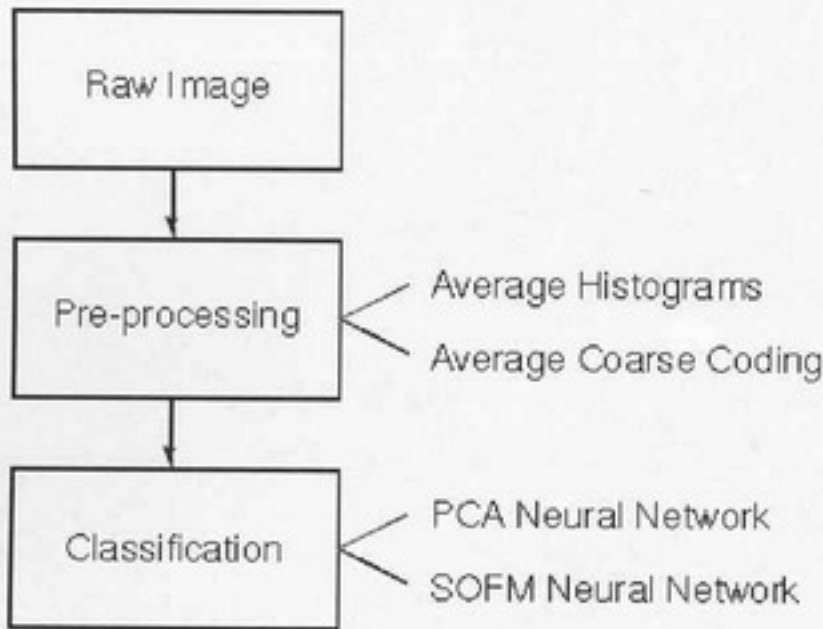


Figure 2: Functional blocks of the vision system

The pre-processing scheme consisted of partitioning the original image into 25 sub-images with 32×24 pixels in size, as shown in Figure 3. After partitioning, pre-processing continued with each raw sub-image serving as input for a feature extraction stage, in order to generate more compact input vectors to the classification stage.

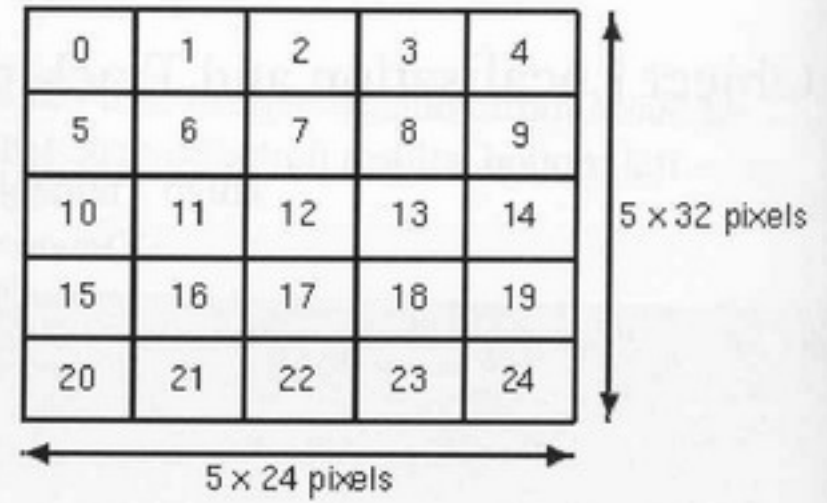


Figure 3: The 25 sub-images that result from partitioning the original image

2.2.2 Image Pre-processing

Two image coding techniques were used in different sets of experiments. The first consisted of computing the horizontal and vertical average histograms of the three RGB colour channels and stacking them into single feature vectors with $3 \times (32 + 24) = 168$ elements (Figure 4).

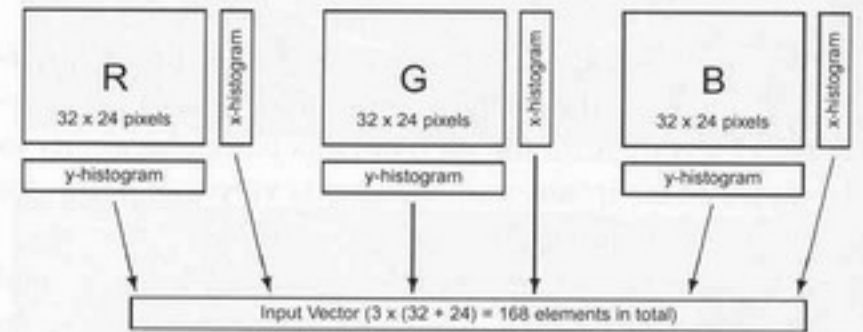


Figure 4: Image pre-processing using histograms

The second image preprocessing mechanism tested consisted of computing the average coarse coding of the three RGB colour channels within an 8×8 neighbourhood, resulting in feature vectors with $3 \times (\frac{32}{8} \times \frac{24}{8}) = 36$ elements (Figure 5).

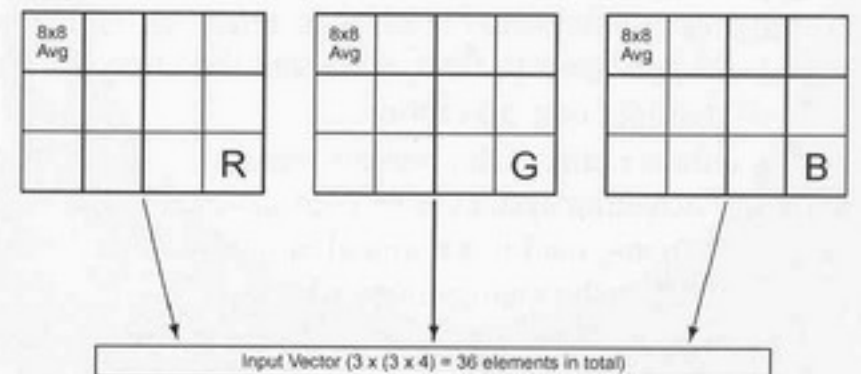


Figure 5: Image pre-processing using coarse coding

2.2.3 Image Classification

Two different neural network approaches were also tested in different sets of experiments. The first was a torus-

shaped self-organising feature map (SOFM) (Kohonen, 1984) with 10×10 units, the second was a single-layer feedforward neural network that performs principal component analysis (PCA) (Sanger, 1989; Ballard, 1997) with 16 units.

The SOFM network was trained according to the usual winner-takes-all approach (Kohonen, 1984), using the similarity matching given in Equation 1, where $\vec{w}_c$ is the winner among all $\vec{w}_i$ units for a given input $\vec{x}$.

$$\| \vec{x}(t) - \vec{w}_c(t) \| = \min_i \{ \| \vec{x}(t) - \vec{w}_i(t) \| \} \quad (1)$$

During the learning phase, the weight vectors of the winner and its neighbours were modified according to Equation 2, where α is the learning rate ($0 < \alpha < 1$) and N_c is the topological neighbourhood of the winner.

$$\vec{w}_i(t+1) = \vec{w}_i(t) + \alpha [\vec{x}(t) - \vec{w}_i(t)], i \in N_c \quad (2)$$

In our experiments, both the learning rate and the topological neighbourhood size decreased with the training cycles, as shown in Equations 3 and 4.

$$\alpha(m) = \exp(-10m/M) \quad (3)$$

$$N_c = \begin{cases} 3 & \text{if } 0 < m \leq 0.2M \\ 2 & \text{if } 0.2M < m \leq 0.5M \\ 1 & \text{if } 0.5M < m \leq M \end{cases} \quad (4)$$

M is the total number of training cycles (100 in all experiments).

The PCA networks were trained with the Generalised Hebbian Algorithm (GHA) (Sanger, 1989). To compute the output vector $\vec{y}$ for a given input vector $\vec{x}$, Equation 5 was used. Equation 6 describes how the weights w_{ij} of the network were adapted.

$$y_i = \sum_{j=1}^J w_{ij} x_j \quad (5)$$

$$\begin{aligned} \Delta w_{ij} &= \alpha y_i [x_j - \sum_{k=1}^i y_k \cdot w_{kj}] \\ w_{ij}(t+1) &= w_{ij}(t) + \Delta w_{ij} \end{aligned} \quad (6)$$

For the PCA network it was necessary to compute the average vector of the training data and then subtract it from all vectors in the set, in order to obtain zero-mean data. The learning rate for the PCA network was also made to decrease exponentially (Equation 7).

$$\alpha(m) = 0.1 \exp(-m/M) \quad (7)$$

Each neural network architecture was trained and tested for each image coding scheme, resulting overall in four different experiments. The data set for training consisted of 50 images, each of an orange football, a blue cylinder and a green box, acquired against the unstructured background of the Brooker Laboratory at the University of Essex. The objects were distributed in random positions and orientations, as shown in Figure 6.

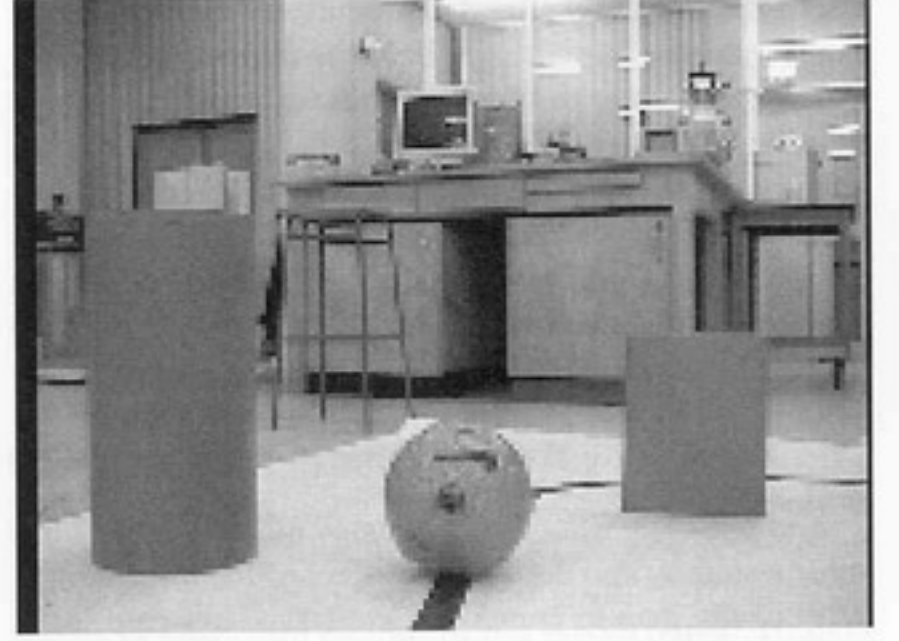


Figure 6: A sample image from the training set

2.3 Experimental Results

2.3.1 Test Images

A set of ten different images of each object in different locations within the image frame was used to test the neural networks. These test images were acquired in the same non-structured environment as the training images, although against a different background. Additional target images were also taken from each object positioned exactly to lie in the central sub-image. Figure 7 depicts a target image for the orange ball.

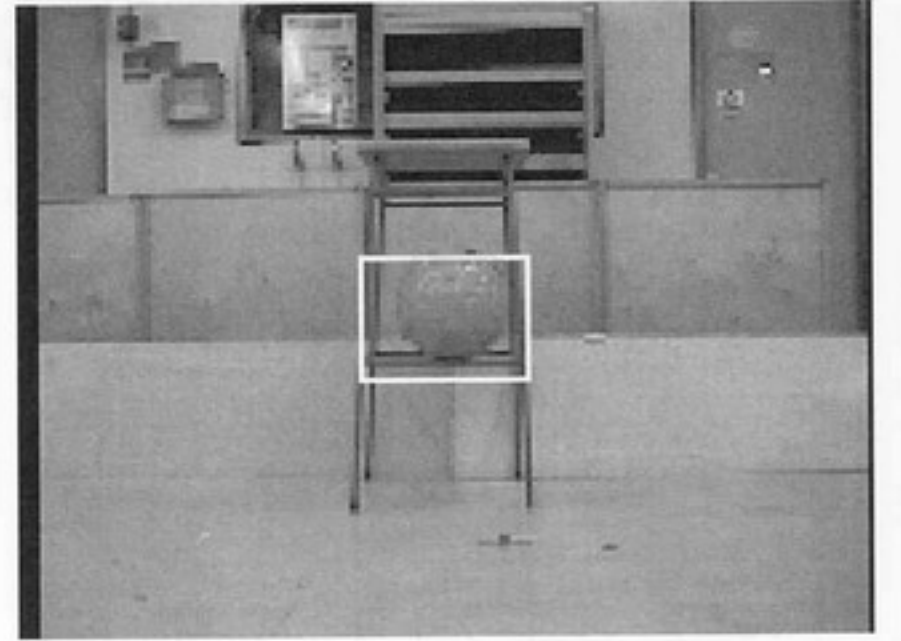


Figure 7: Target image of the orange football

2.3.2 Identification of Target Sub-images

For the SOFM architecture, the activation elicited by the target image was subsequently used to identify that sub-image in which the target object lay in each of the ten test images. The measure of similarity between the activations of the target sub-image and the other sub-images was a three-dimensional euclidean distance metric, which made

use of the two-dimensional coordinates of the winner unit in the map and its activation value.

For proper results, the activation value of the winner must be scaled according to some criteria to make its contribution to the distance measure proportional to the contribution of its coordinates in the SOFM. The scaling criteria employed in our experiments is given by Equation 8, where S is the side dimension of the square SOFM (10 in our experiments).

$$a_c = S \times \| \vec{x} - \vec{w}_c \| / \max_i \{ \| \vec{x} - \vec{w}_i \| \} \quad (8)$$

The two-dimensional coordinates of the winner units must also take into account the effect of the torus boundaries of the map to compute the distance between them. Ambiguous coordinates of the winner units were selected to minimise the euclidean distance between them.

For the PCA network, a similarity measure of the same nature was also used. The euclidean distance was used again to evaluate the output vector of the network for the target sub-image in comparison to the output vectors of the sub-images from the test images. However, in the case of the PCA architecture, the simple n-dimensional euclidean distance was used without the need of any scaling procedure.

Figure 8 shows an example of the results for the test images of the orange football using the histogram-based image coding and the PCA network. It can be seen from these results that the system is reasonably robust to small changes in scale and translations of the target object within the detected sub-images.

To evaluate results, we manually determined the correct system response as that sub-image within which at least 25% of the target surface lay. We then composed this correct target identification against the one the system produced. A summary of the results obtained for all the four experiments is shown in Table 1.

Table 1: Success rates of the experiments

Average Histograms			
	Blue Cylinder	Green Box	Orange Football
SOFM Network	80%	30%	20%
PCA Network	90%	30%	70%

Average Coarse Coding			
	Blue Cylinder	Green Box	Orange Football
SOFM Network	70%	20%	20%
PCA Network	90%	10%	70%

A set of χ^2 tests were also performed on the results of the experiments. As only a restricted amount of test images was available, it was necessary to group neighbouring sub-images to satisfy practical constraints of the

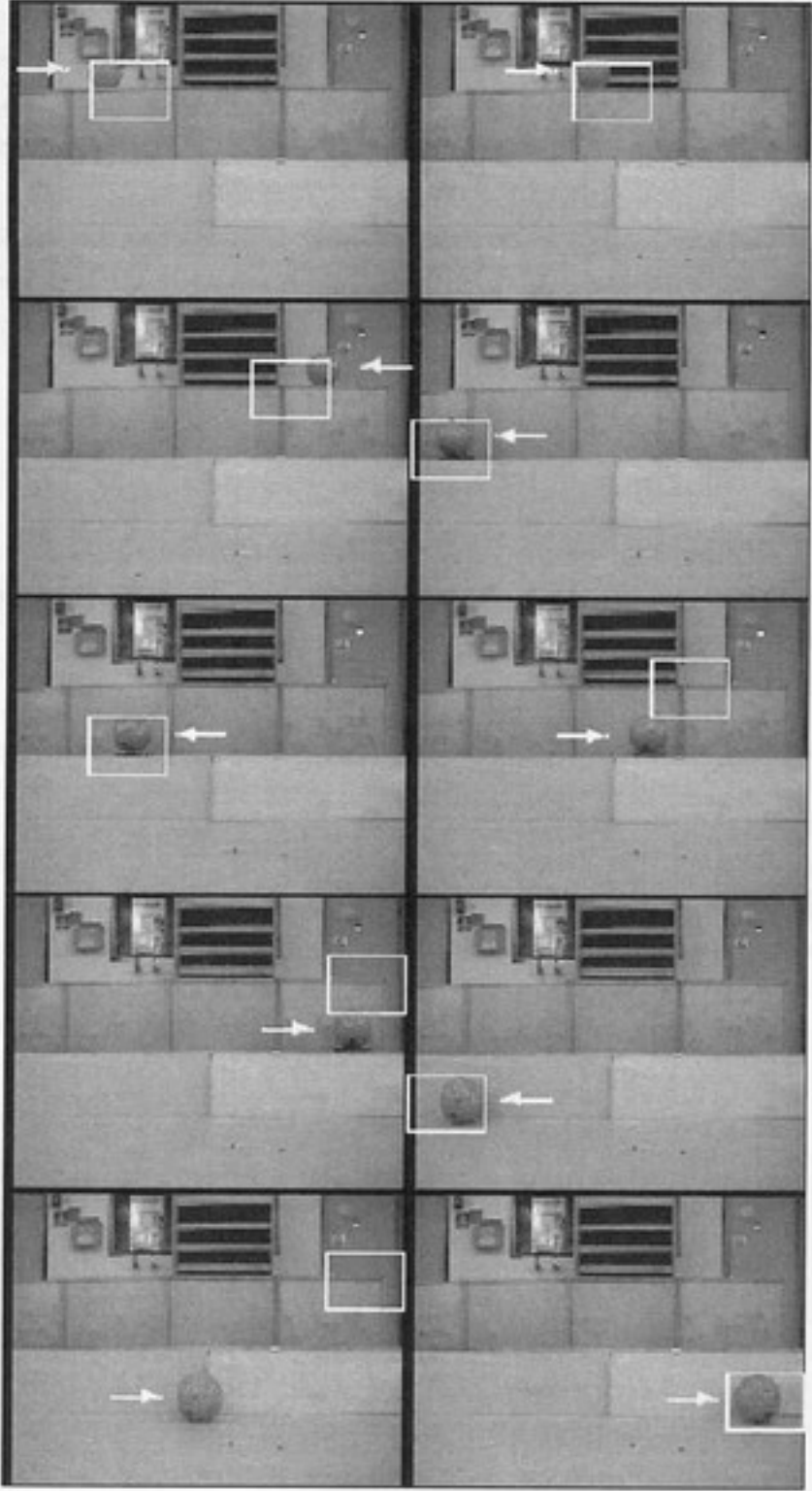


Figure 8: Results obtained using the histogram-based image coding and the PCA network (the arrows indicate the target object and the rectangles indicate the sub-image in which the target was detected)

χ^2 test. This roughly resulted in the division of the image frame into four regions, as illustrated in Figure 9.

The χ^2 tests for all experiments have shown statistically significant correlation between actual and identified target positions using all of the 30 test images ($p = 0.05$).

From the previous analyses, it can be inferred that the PCA network has a better performance for both image coding techniques in this particular task. In a similar way, the average colour histograms image coding scheme provide better results for both neural network architectures.

More importantly, however, is that the results given by the χ^2 test indicate that our approach identifies the target object's location at least within four large areas of the image frame. Although this location information might seem

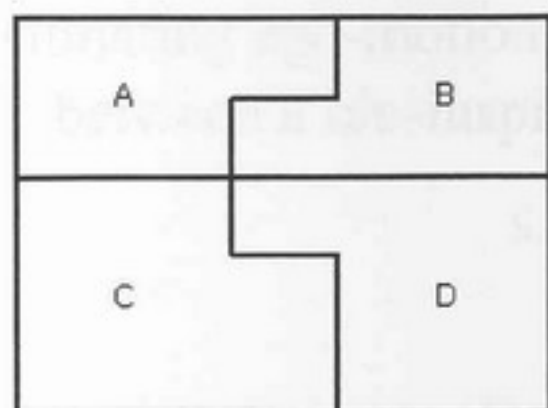


Figure 9: Sub-image groups used for the χ^2 test

very coarse at first, it is enough to be used to close the loop between perception and action, driving the camera's pan-tilt unit in two-dimensions to follow the target object and position it in the centre of the frame.

Our argument is that the simple fact of moving the camera in the direction of the target object in an active vision approach will not only be feasible, but will also improve the target localisation performance.

3 Conclusions

The eventual application this research is aimed at is that of novelty detection and navigation in mobile robots using visual information. In this paper, we presented a sub-symbolic, unsupervised mechanism to identify and track objects within the image frame using passive vision.

The results of our experiments have shown good performances for the blue cylinder when compared to the performances for the orange football and specially the green box, which was very poor. This might be attributed to the fact that the blue cylinder was the biggest object in the set and thus occupied a greater area within the image frame, increasing its probability of being identified. Another plausible explanation resides in the colour of the cylinder, which offered the greatest contrast against a mostly yellow background.

Feature extraction using average coarse coding with a neighbourhood of 8×8 pixels gives as a result a very blurry version of the original image, where basically only colour features are preserved. In spite of reduced details, the presented unsupervised neural network approaches were able to identify the target objects reasonably, indicating their suitability to constitute a simple colour-based object detector. Future work includes testing approaches using hue, saturation and intensity parameters (HSI colour model), which may result in improved performance in class separation with respect to colour features (Jain et al., 1995).

Average histograms seem to represent both shape and colour information and presented a good tolerance to scale and translation within the detected sub-image. This image coding scheme provided even better results than

coarse coding. Further tests need to be conducted with objects that are not present in the training data set, in order to evaluate the robustness of this method to objects that have never seen before by the system.

The PCA network was faster to train and also had the advantage of having a simpler similarity measure procedure than the SOFM. However, the convergence of the algorithm used to train the PCA architecture is very sensitive to several details. This includes the need of zero-mean data and small learning rate values for the training phase.

A special conclusion that arises from our χ^2 analysis is that the combination of techniques employed in our experiments are able to estimate the position of the target image in four main regions of the image frame. Despite being very simple, this characteristic is useful enough to drive an attention mechanism for an active vision system, according to some preliminary results of our next step in future work.

Our expectations are that active vision will contribute to improve the ability of the system to locate the target object. Several case studies that support better performances of active vision approaches are presented by Pfeifer and Scheier (Pfeifer and Scheier, 1999). Future work also includes the development of a "self-motivated" attention mechanism, in order to eliminate the need of user-provided target objects. The target is desired to be automatically selected, and the attention of the system directed to it, by the amount of novelty in its features.

Our initial analysis suggests that the approach presented in this paper is useful as part of the entire visual novelty detection system we have in sight. The feature extraction techniques employed so far are fairly sensitive to objects with saturated colours, although further experiments using monochromatic images need to be executed to confirm this hypothesis.

Acknowledgement

Hugo Vieira Neto is sponsored by a CEFET-PR and CAPES Foundation fellowship, whose support is gratefully acknowledged.

References

- Dana H. Ballard. *An Introduction to Natural Computation*. MIT Press, Cambridge, MA, 1997.
- Ramesh Jain, Rangachar Kasturi, and Brian G. Schunk. *Machine Vision*. McGraw-Hill, New York, NY, 1995.
- Teuvo Kohonen. *Self-Organization and Associative Memory*. Springer-Verlag, Berlin; New York, NY, 1984.
- Stephen Marsland, Ulrich Nehmzow, and Jonathan Shapiro. Detecting novel features of an environment

using habituation. In *From Animals to Animats: Proceedings of the 6th International Conference on Simulation of Adaptive Behaviour (SAB'2000)*, pages 189–198, Paris, France, 2000. MIT Press.

Stephen Marsland, Ulrich Nehmzow, and Jonathan Shapiro. Environment-specific novelty detection. In *From Animals to Animats: Proceedings of the 7th International Conference on Simulation of Adaptive Behaviour (SAB'02)*, Edinburgh, UK, 2002. MIT Press.

Ulrich Nehmzow. Vision processing for robot learning. *Industrial Robot*, 26(2):121–130, 1999.

Rolf Pfeifer and Christian Scheier. *Understanding Intelligence*. MIT Press, Cambridge, MA, 1999.

Terence D. Sanger. Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural Networks*, 2:459–473, 1989.

Estimating ego-motion using a panoramic sensor: Comparison between a bio-inspired and a camera-calibrated method

S. Gourichon*, J.A. Meyer*

*AnimatLab – LIP6

8, rue du Capitaine Scott

75015 Paris, France

stephane.gourichon@lip6.fr, jean-arcady.meyer@lip6.fr

S. H. Ieng[†], L. Smadja[†], R. Benosman[†]

[†]Laboratoire des Instruments et Systèmes

Université Pierre et Marie Curie

4 place Jussieu

75252 Paris cedex 05

rbo@lis.jussieu.fr

Abstract

We introduce and compare two algorithms related to ego-motion, applicable to a robot using a panoramic visual sensor in an unknown environment. The first method, computationally cheap, extends a family of bio-inspired navigation systems by providing an orientation estimation that previously had to be obtained from external references. The second method solves the general structure-from-motion problem and is here adapted to the panoramic sensor case. Both methods produce good results in a test scene, though the calibrated method has a greater precision. A step-by-step comparison of the methods is included, with mobile robotics applications in mind.

1 Introduction

In Gourichon et al. (2002), a biomimetic homing navigation method was applied to a robot, based on the work of Cartwright and Collett (1983, 1987) on searching behaviour in bees. Because this method assumed that the orientations of two snapshots – i.e., two one-dimensional projections of thresholded luminances on a circular retina respectively taken in a given goal place and in any other place close to the former – were known to the robot, only constrained moves were allowed, committing the robot to always face the same direction. To alleviate such constraint, a visual compass has been designed, thus making it possible to monitor the robot's changes of direction along any path, thanks to simple measures of parallax and to simple extensions of Cartwright and Collett's logic of snapshot comparisons.

Turns out besides that monitoring a robot's orientation is part of the more general issue of estimating its ego-motion (Chang and Hebert, 2000;

Faugeras and Maybank, 1990; Tomasi and Kanade, 1992; Spetsakis and Aloimonos, 1990), and that sophisticated 3D-vision algorithms may be used to monitor both the translations and rotations of the robot. Incidentally, the same algorithms may also serve for the 3D reconstructions of unknown environments (Ikeuchi et al., 2001; Narayanan and Kanade, 1998; Faugeras et al., 1998).

To assess the respective merits of these two independent lines of approach, an experiment has been set up to demonstrate their effectiveness at determining the orientation of a robot equipped with a catadioptric sensor. Two methods have been accordingly implemented. The first one calls upon simple bio-inspired parallax comparisons, while the second is more mathematically inclined and extensively relies on non Euclidian geometries and minimizations techniques.

This paper describes the basic principles underlying these methods, and the results that were obtained

in simple experimental conditions. The chances that these methods will prove to be efficient in more challenging environments are also discussed, and indications for future improvements are provided.

2 A visual compass from a biologically inspired context

The first method in this comparison is a simple and computationally cheap visual compass designed for an animat able to perceive the azimuths of some landmarks relative to its body. After an observation stage, done once and for all for a given location, there is an exploitation stage when, from any location in the vicinity, the visual compass can estimate the absolute orientation of the animat's body using only perceived relative angle measurements.

Only azimuths are used, no elevation, which makes sensor calibration unnecessary. A simple segmentation algorithm and a matching procedure by dynamic programming are used.

The essence of the algorithm is to take advantage of the local linearity of the variation of any azimuth with respect to the animat motion. This allows, at the observation stage, to capture relevant information without knowing the complete motion of the animat. At the exploitation stage, this also allows to very simply retrieve the animat's orientation from any later view taken in the vicinity of the observation place.

2.1 Early image processing

It is necessary to extract azimuths from the sensor's image. The image processing used here is voluntarily simple. It starts from a color panoramic image (see figure 7 right). The first step is to unroll a ring portion of the color input image according to rectangular-polar transformation to obtain a rectangular colored strip indexed by azimuth. Then a simple segmentation procedure is used, that selects portions of the panorama with highly saturated (colorful) pixels, and separates regions of nearly constant hue at azimuths where the average hue variation is beyond a threshold. The result is a set of regions sorted by azimuths, with possible holes caused by portions of the initial panorama without bright colors. A real-world panorama typically results in less than 100 regions. In this experiment, the regions correspond to the colored cardboard.

The segmentation algorithm, though very rough, is the same we already used successfully in real-world robotic short range guidance experiments in Gourichon et al. (2002).

2.2 Matching algorithm

The visual compass requires the ability to establish correspondences between objects appearing in different views. Here we called upon a dynamic programming algorithm that we already used for short range guidance. It is explained in more details in Gourichon et al. (2002). It globally maximizes a matching score from a local score function that takes color information (particularly hue) into account. This step remains reasonably cheap ($O(n^2)$ with $n < 100$) because the segmentation algorithm provided a small set of regions.

2.3 Using the parallax in two stages

As summarized above, the biologically-inspired visual compass operates in two stages: an observation stage, and an exploitation stage.

2.3.1 Notations

Let us introduce the notations illustrated in figure 1. Given a view taken at place X , we name $\theta_{a \rightarrow b}^X$ the signed angle between the azimuth of feature a and the azimuth of feature b , where a feature may be either a landmark number such as i , or a reference direction such as *north* or *body*.

We call parallax of a landmark during an animat's motion the variation of azimuth caused by this motion. Thus, for landmark i , for a motion from M to N , the parallax is $D_i^{M \rightarrow N} = \mu(\theta_{north \rightarrow i}^N - \theta_{north \rightarrow i}^M)$ where $\mu(x)$ is the number closest to zero in the set $(x \text{ modulo } 2\pi)$.

2.3.2 Observation stage

During the observation stage the animat observes the parallax caused by ego-motion. The animat has to know its orientation during this stage, but the translation can be unknown. It takes three views (named M, N, P) that are not aligned and have in common at least three landmarks (named a, b, c). The ideal situation is one distant landmarks and two close landmarks, as we will illustrate later. The worst situation is when the observation region and the three landmarks lie on a circle.

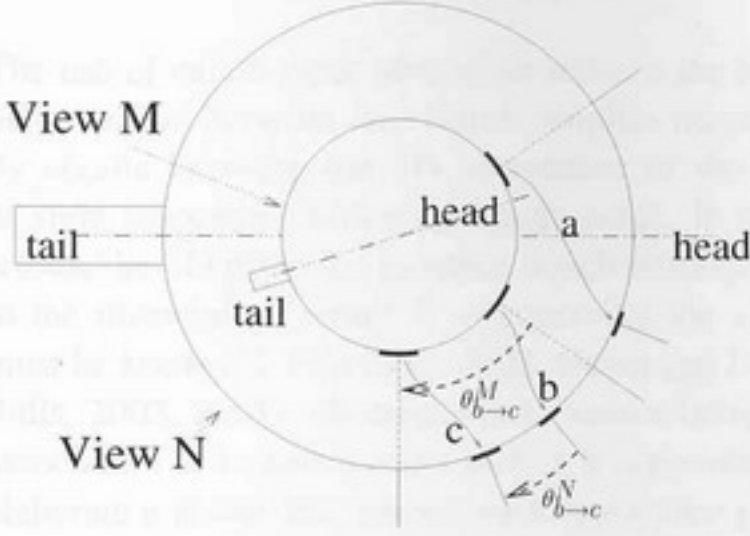


Figure 1: Illustration for notation $\theta_{a \rightarrow b}^N$. Curved arrows represent three landmarks a, b, c paired between views M and N by the matching process.

From the parallax measured we will calculate what we'll call a "bearing". It is a set of three real numbers $\lambda_a, \lambda_b, \lambda_c$ that are attached to the triplet and will be used to estimate a direction.

For $(i, j, k) \in \{(a, b, c)(b, c, a)(c, a, b)\}$, let

$$\lambda_i = D_j^{M \rightarrow N} D_k^{M \rightarrow P} - D_j^{M \rightarrow P} D_k^{M \rightarrow N}$$

If the sum of the three λ 's is close to zero, the triplet is unsuitable and simply ignored. The reason for this formula and details are given in Gourichon (2003). Usually, more than 3 landmarks are available in a standard situation. Then, all suitable triplets are enumerated and used to build corresponding bearings.

2.3.3 Exploitation stage

Each time a new view X of unknown orientation is taken, that has common landmarks with the views used at the observation stage, an exploitation stage can be conducted. It consists in estimating the animat's orientation, using a previously memorized bearing.

If the orientation of the new view is completely unknown, then it is first artificially rotated so that the landmark that has the highest lambda in the bearing appears to have the same azimuth as in view M. This prevents angular wrapping artifacts of pseudo-parallaxes in equation 1 below.

The denominator attached to the bearing is calculated:

$$d = \lambda_a + \lambda_b + \lambda_c$$

as well as a linear combination that depends on X and that we call the invariant:

$$I(X) = \sum_{i \in \{a, b, c\}} \lambda_i \underbrace{\mu(\theta_{body \rightarrow i}^X - \theta_{body \rightarrow i}^M)}_{\text{pseudo-parallax}} \quad (1)$$

Then the current orientation of the animat's body $\theta_{north \rightarrow body}^X$, that can be estimated from this bearing is simply:

$$\theta_{north \rightarrow body}^X = \theta_{north \rightarrow body}^M - \frac{I(X)}{d} \quad (2)$$

Let us summarize how it works. The pseudo-parallax is a real parallax if views M and X have the same orientation reference, i.e. $\theta_{north \rightarrow body}^X = \theta_{north \rightarrow body}^M$. The three λ 's are calculated so that the linear combination of parallaxes is constant, in first order approximation, in the vicinity of the observation places. As a consequence, if the view X is correctly oriented, $I(X) \simeq 0$. This is exact by construction, for $X = M$, $X = N$ and $X = P$.

At the observation step the linear combination $I(X)$ involves pseudo-parallaxes, because it is calculated as if the view X was correctly oriented, while it is not necessarily the case. This generates the same offset at the level of each parallax, which contributes to the linear combination. Since the linear combination is chosen to yield a constant result with real parallaxes, the difference between actual and expected values is the error in orientation (the only unknown value) multiplied by the sum of the linear coefficients d (which is known). Hence equation 2.

To make things better, more than a single invariant may be available, because more than one bearing may have been recorded at the observation step, from as many triplets of landmarks. To contribute to the overall direction estimate, each bearing yields a unit vector pointing to the estimated body direction. These unit vectors are summed up and their resultant provides the aggregated direction estimate. The length of the resultant vector is used as a self-confidence estimator: it is maximal when all bearings can be used and all indicate the same direction, while it is reduced if the different estimated directions contradict each other. Additional details are given in Gourichon (2003).

2.3.4 Example

For clarity of the example and figure, the equations in this example are simplified and use absolute az-

imuths θ instead of parallaxes, and would not always work properly *as is* because of angular wrapping effects.

Figure 2 shows an example animat motion and an observed invariant. Since $\theta_{north \rightarrow i} = \theta_{north \rightarrow body} + \theta_{body \rightarrow i}$, this linear equation it is very easy to solve for $\theta_{north \rightarrow body}$.

If the animat moves or rotates, or a combination of both, and takes a new view X , $\theta_{north \rightarrow body}^X$ is as simple to retrieve as:

$$\frac{43.25 \cdot \theta_{body \rightarrow a}^X + 92 \cdot \theta_{body \rightarrow b}^X + 55.25 \cdot \theta_{body \rightarrow c}^X}{43.25 + 92 + 55.25}$$

It is interesting to notice that the λ s are calculated based on a particular motion, which is unknown at any time. Only a constant orientation during this motion is assumed. In a sense, this method separates orientation from motion, or compensate motion to retrieve orientation.

One of the prerequisite for this method is that this separation is possible. For a particular triplet, there is a region where it is not true (figure 3).

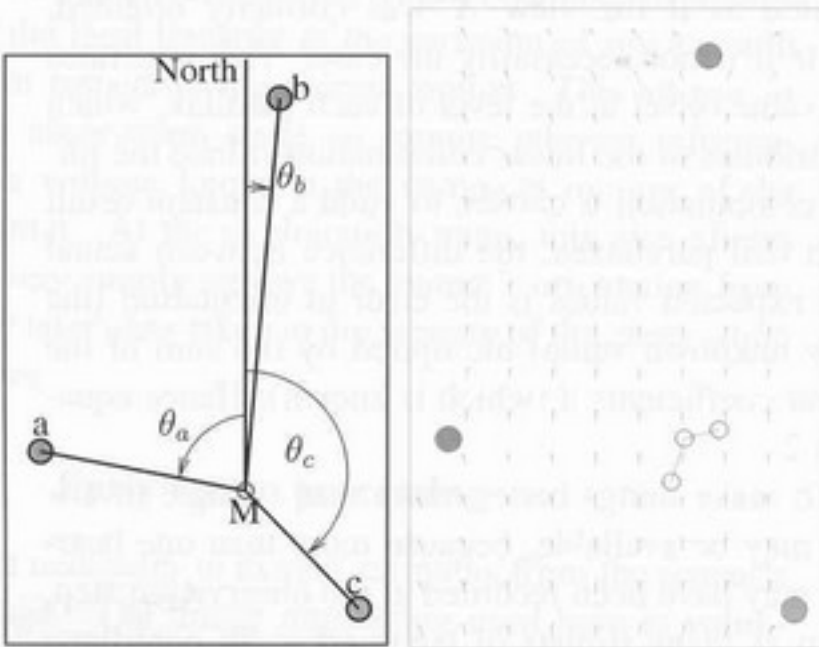


Figure 2: Example illustrating the principle of the bio-inspired visual compass. Shaded circles are landmarks. Moving along the three connected open circles (right) the animat observes that $43.25 \cdot \theta_{north \rightarrow a} + 92 \cdot \theta_{north \rightarrow b} + 55.25 \cdot \theta_{north \rightarrow c}$ is approximately constant. Since this is a linear equation it is very easy to solve for $\theta_{north \rightarrow body}$. Actually, this equation would suffer from angular wrapping effects, which is why parallax are used instead of azimuths, and the view is first pre-oriented in favor of the most stable landmark (the one with the highest lambda, see text). The vector field on the right shows the estimated north direction at points of a grid.

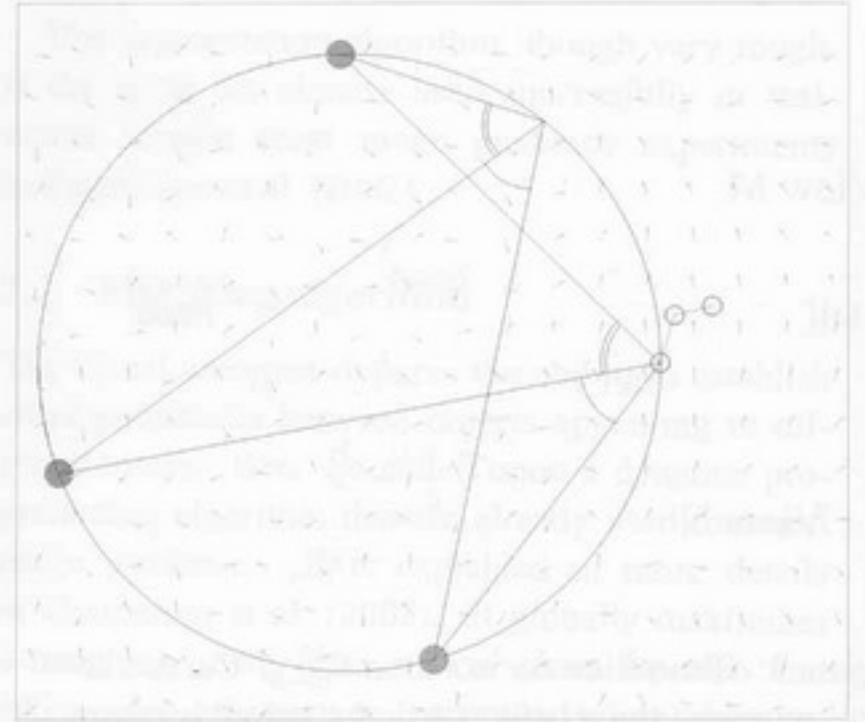


Figure 3: Example illustrating a degenerate case and a limitation of the bio-inspired method. The circle traversing any three landmarks has the geometrical property that the perceived relative angles are the same from any position (represented here at two positions). From the animat's point of view it means that orientation (turning around itself) cannot be separated from motion (moving on the circle). As a result, if the observation region is too close to this circle, the corresponding triplet will produce unreliable results. In this example the observed parameters are $\lambda_a = -31$, $\lambda_b = 14$, $\lambda_c = 16$ and $d = -1$. d is much smaller than any lambda, which is a measurable sign of a bad triplet in this region. In a real situation, the compass would have ignored it in favor of other triplets.

3 A camera calibration-based visual compass

This method estimates both rotational and translational components of motion from two views only. In a first stage, it calibrates the internal parameters of the camera, thus making it possible to get elevations from the camera image. In a second stage, a matching procedure based on robust methods is applied. It is rotation and scale invariant to cope with the non linear resolution of the sensor. Finally a structure-from-motion direct linear algorithm retrieves the motion parameters using a robust method, providing complete ego-motion (i.e. translation and rotation).

3.1 The necessity of calibration

The use of catadioptric sensors to retrieve the complete motion between two frames implies necessarily entails knowing the 3D orientation of the line of sight associated with each image pixel. In other words, the 3D direction in space which corresponds to the direction in which it is observing the scene must be known (J. Fabrizio, 2002; Geyer and Daniilidis, 2002, 2001). A catadioptric sensor being an association of a camera and a mirror, it is possible to elaborate a model that associates to each view point (camera, mirror) a 3D coordinates system. It is then important to retrieve the unknowns of the model, which are the camera internal parameters, the distance between the camera and the mirror and the orientation of the mirror. The mirror being generally manufactured with a very high precision, its parameters are assumed to be known. Moreover, by construction, most panoramic catadioptric sensors fulfill what is called the *single viewpoint constraint*, which means that all incident rays reflected by the mirror and intersecting in the camera focal point also intersect in a single point called the mirror viewpoint by Baker and Nayar (1998). Retrieving the unknown parameters makes it possible to check the validity of the hypothesis of the single view point constraint, and allows a better computation of the 3D vector of sight associated with each pixel (see figure 4). This is an advantage as the process of determining the motion from two consecutive images is very sensitive to noise and can provide distorted results if the calibration does not provide accurate parameters (see next section).

3.2 Matching catadioptric images

To be able to recognize common real-world features appearing in two images, it is necessary to perform a match.

Most of the existing matching methods rely on the same principle. They first extract features from two "rough" images, generally high curvature points. Then, they select around each point a neighborhood of points and compute a correlation between all the neighborhoods. The best score implies the best fit (Faugeras, 1993).

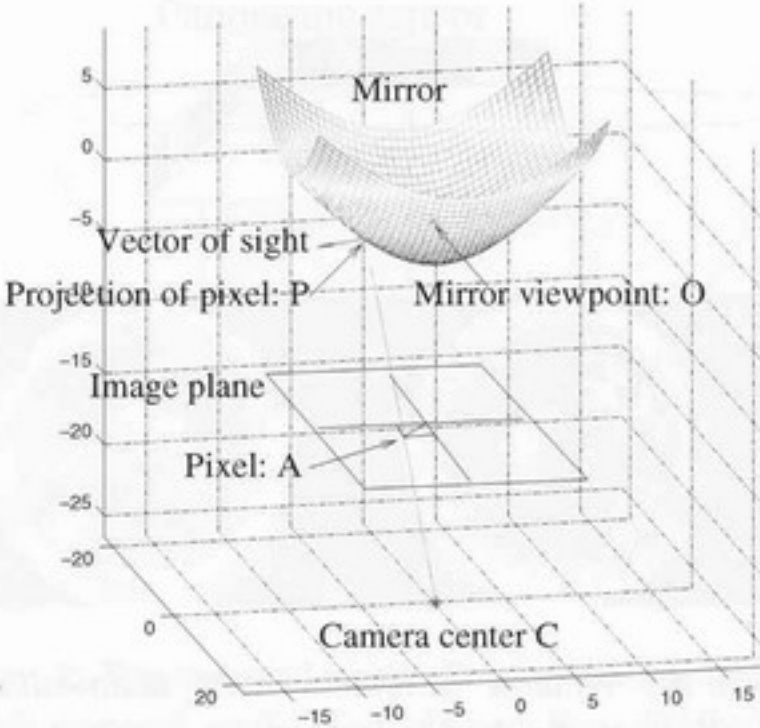


Figure 4: The aim of the calibration is to find the precise relationship between pixel coordinates in the image and the 3D direction of the vector of sight. This representation relies on the estimated model provided by calibration.

3.2.1 Feature extraction

The feature extraction normally used is a corner extraction, performed using a classical Harris detector (Harris and Stevens, 1988), directly applied to images, i.e. without paying attention to their catadioptric origin.

For this experiment, black spots were used instead (see figure 7). The feature points were the center of the spots.

3.2.2 Window definition

Matching interest points in stereoscopic views requires a sharp neighborhood extraction. Classical approaches use predefined size patches, often squares, centered on interest point. Unfortunately, in the present case, similar approaches would not give reliable results because the shape of the mirror does not guarantee an homogeneous resolution (it is maximum at the center and decreases when getting near the borders). This is why a dedicated procedure, derived from Svoboda and Pajdla (2001), which takes into account the spatial resolution variation by means of dynamically resized patches is used instead.

The shape of the neighborhood windows are not squares but diamonds defined by their vertices (see figure 5). Thanks to calibration, there is a two-way correspondence between each pixel in the image and

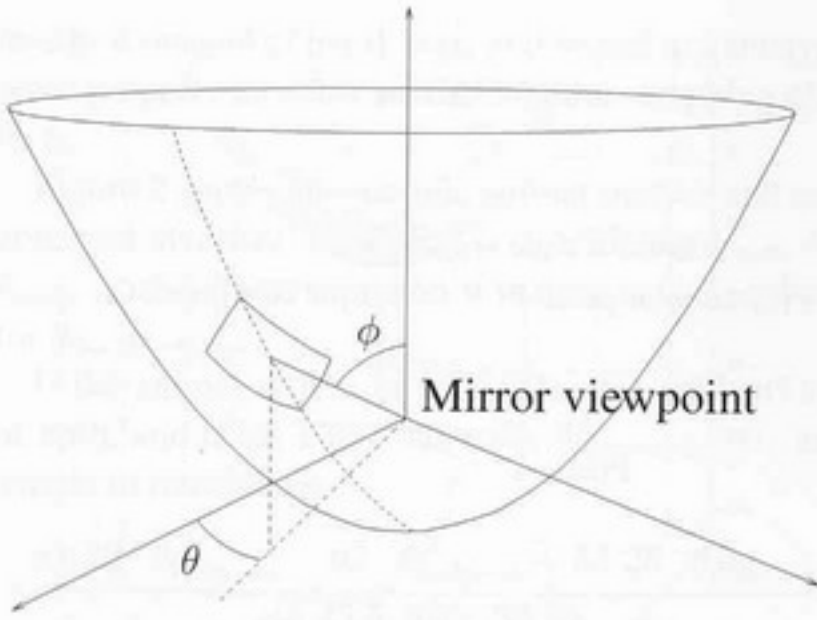


Figure 5: Window definition. The diamond is defined from the mirror surface, its vertices are obtained by setting azimuth (θ) and elevation (ϕ) ranges.

its projection on the mirror. Each feature point is projected on the mirror. The quadrilateral patch definition is an original method that defines the associated window on the mirror surface by fixing angular ranges around the point's azimuth θ and elevation ϕ (see figure 5). Then, the vertices of the diamond are projected again from the mirror to pixels in the image plane. By setting the neighborhood window on the mirror instead of the image, we are able to cope with the spatial resolution requirement. Dimensions of the window increase with the distance to the center, under the assumption of fixed angular parameters.

3.2.3 Matching and outliers removal

Optimal matching requires an appropriate sampling of the extracted neighborhoods which may have different size. Each window is accordingly resized using a classical bilinear interpolation.

Since the windows are projected on the mirror at a known azimuth, disparity angle for each couple of points can easily be computed. Thus, patches are correctly oriented before the matching process.

The correlation score computation should ensure one, and only one, best match for each point; therefore we use a centered and normalized cross correlation. Points are paired if they mutually give the best similarity scores.

This is still not enough to get reliable points. In order to remove outliers, a threshold extracted from

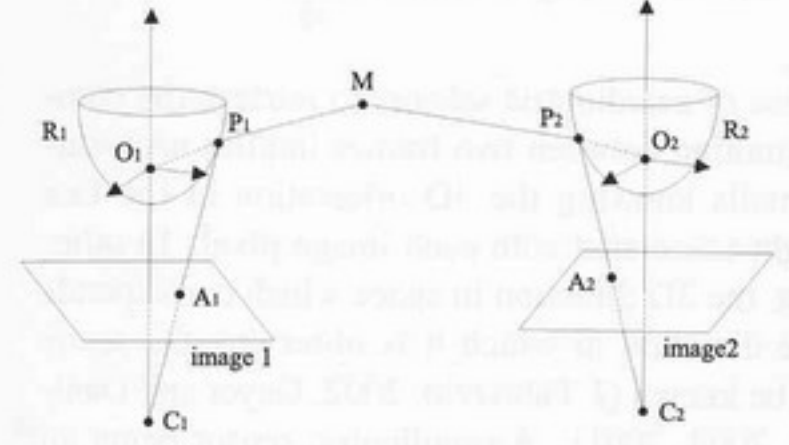


Figure 6: Motion estimation step. The sensor was moved from referential R_1 to referential R_2 between image 1 and image 2. If A_1 and A_2 are two correctly matched points there is a relation between O_1 , O_2 , P_1 , and P_2 . The coherence of this relation among many points allows at the same time to estimate the motion and reject mismatched points.

matched points is applied on the measured angular disparity set. Pair of matched points are removed if their relative angles are too far from the computed average value ($|\theta - \bar{\theta}| > 2.5\sigma$). Additional details are given in Ieng et al. (2003).

3.3 Extrinsic parameters estimation

The next step is to actually find the motion between our two matched images. This step is based on previous work from Faugeras and Maybank (1990); Chang and Hebert (2000) and Svoboda et al. (1998).

Let us consider a couple of matched points, A_1 in image 1 and another A_2 in image 2. If all the previous steps worked correctly, the two matched points correspond to the same location M in the surrounding scene (figure 6). If it is true, then O_1 , O_2 , P_1 , and P_2 lie on the same plane:

$$\overrightarrow{O_1P_1}_{|R_1} \cdot (\overrightarrow{O_1O_2}_{|R_1} \wedge \overrightarrow{O_2P_2}_{|R_1}) = 0 \quad (3)$$

Changing the form of the equation as followed leads to a well known property in stereovision, which is interesting because P_1 and P_2 are actually known:

$$P_1^t \cdot E \cdot P_2 = 0 \text{ assuming that } \begin{cases} P_1 = \overrightarrow{O_1P_1}_{|R_1} \\ E = \overrightarrow{O_1O_2}_{|R_1} \wedge R_2^1 \\ P_2 = \overrightarrow{O_2P_2}_{|R_2} \end{cases}$$

The only unknown is E , the essential matrix defined by Longuet-Higgins (1981). This matrix repre-

sents the coordinate transformation between referentials $R1$ and $R2$, that is the translation and rotation that was performed between the views we are working on, i.e. rotation R_2^1 and translation T_2^1 components.

Equation (3) is simplified to a more explicit form: $U.e = 0$, where $U = (p_{1x}p_{2x}, p_{1x}p_{2y}, p_{1x}p_{2z}, p_{1y}p_{2x}, p_{1y}p_{2y}, p_{1y}p_{2z}, p_{1z}p_{2x}, p_{1z}p_{2y}, p_{1z}, p_{2z})$ and e is a column vector containing all components of E .

This is a dot product with eight terms. In concrete terms, if all matching was perfect and measurements without noise, seven of those equations would be enough to know the motion. Thus, this method needs at the bare minimum, 7 matched points in the environment.

Actually, there are mismatched points and their position is not exactly known. So the problem is transformed into minimizing $\|U.e\|$. As estimating E means minimizing errors, we solved our optimization problem by weighting contribution of each interest point.

The reader can refer to the work of Zhang (1998) for a complete review of the robust existing techniques.

4 Experimental setup

In order to be able to compare both methods in the real world, we used the same testbed. Although both methods are meant to work on unprepared environment, we validated them here using a constrained, prepared one. This makes it easier to cope with experimental problems and have reproducible results.

The experimental setup is shown in figure 7. Around a flat and rectangular surface were displayed several colored patterns. A series of 2D positions forming a grid was determined. At each 2D position, we grabbed an image, forming a database of several view points. Each method was applied on the images of the database, with the goal of determining the orientation between selected frames.

5 Experimental results

5.1 Biologically-inspired method

After all the views have been acquired, three views were selected in the center of the arena for the observation step, which built 120 bearings.

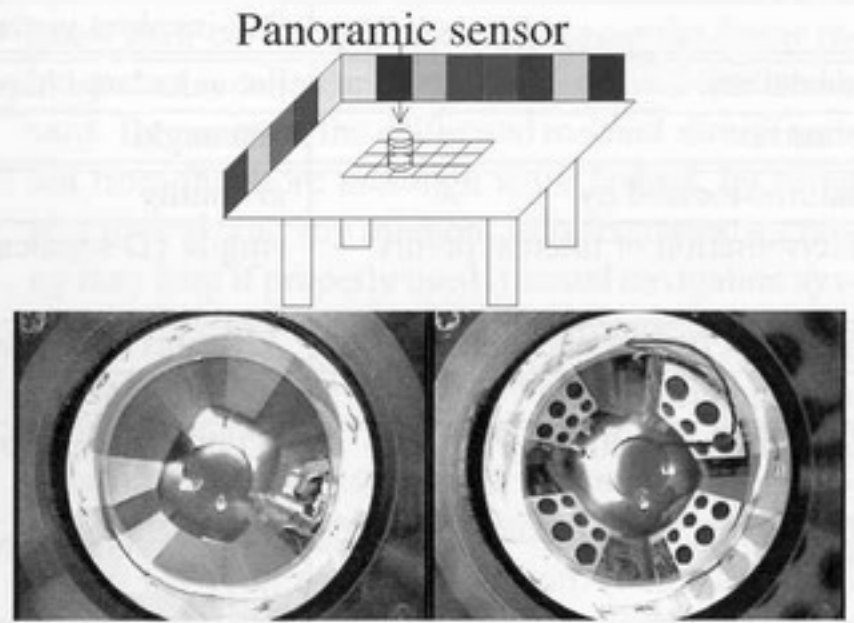


Figure 7: Experimental setup. For the bio-inspired visual compass, colored cardboard was attached around a table to provide environmental features that were easy to segment. For the calibrated method, black spots were displayed to ensure that points were available at different elevations.

Then, for each available view (including the ones used for observation) we applied the bio-inspired algorithm to estimate the orientation of the sensor when the views were taken. Results are shown on the left of figure 8. In this example, the mean error is 1.39 degrees and the standard deviation 3.54 degrees.

5.2 Sensor calibration-based method

After all the views have been acquired, one view was selected as reference (shown in red in figure 8).

Then, for each available view we applied the calibrated method, estimating its orientation with respect to the reference view.

In this example, the mean error is 0.155 degrees, and standard deviation is 0.156 degrees.

6 Discussion

In this run, the results of the camera-calibrated approach are considerably more precise (standard deviation 0,16 degrees versus 3.54) in this specific case. Naturally, additional experiments would be necessary to draw more generic conclusions. Furthermore, none of the two methods was tested in the case of an unprepared environment in which they are intended to be used in the future.

A number of differences are summarized in table 1.

	Bio-inspired visual compass	Calibrated method
translation	not calculated here	calculated
rotation	one angle	three angles
features located by	azimuths	azimuths and elevations
determination of interest points	simple 1D segmentation	center of displayed spots or corners
minimum features needed	3 regions (landmarks)	7 points
matching criterion	color (mainly hue)	2D neighborhood
matching procedure	1D dynamic programming	multi-resolution resampling and correlation
aggregation	simple vector sum	sophisticated minimization techniques

Table 1: Feature comparison between the two methods. Most differences illustrate the differences in the contexts in which those methods were developed.

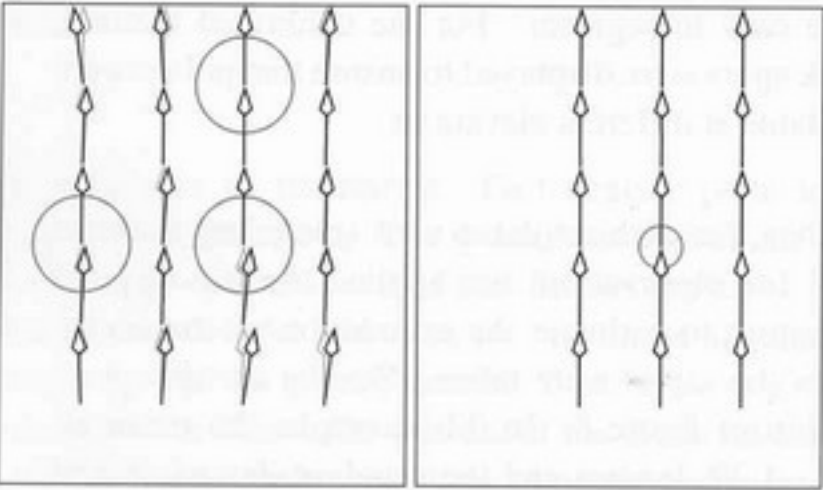


Figure 8: Experimental result. Left: biologically inspired method. The circles represent the three locations that were used for the observation stage. Each black arrow shows the orientation computed at its origin. The shaded arrows show the true reference direction. Right: Sensor calibration-based method. The error is too small to be noticed visually. Only three rows of images were taken.

The simplification affordable by assuming planar motion alone was experimented by Faugeras et al. (1998) in the case of classical projective cameras. In our case with a catadioptric sensor, further simplification can be done because the axis of rotation is the same as the camera-mirror axis direction. The choice of the bio-inspired visual compass to use only azimuths makes it independent of mirror shape and camera parameters, which makes calibration unnecessary.

One might wonder why the original bio-inspired ego-motion method focuses only on rotation. The reason is that the biomimetic navigation models

needs correctly oriented snapshots (hence this work) to produce a direction to go to reach the goal (where the reference snapshot was taken). Franz et al. (1998b) proved that the direction to go can be considered as an approximate ego-motion vector, which error in direction is less than 90° . This is of course insufficient for many applications of ego-motion, but actually sufficient for robot navigation.

The visual compass needs at the bare minimum 3 identifiable features, all visible from the 3 observation views. By contrast, the calibrated method needs at the bare minimum 7 points from the reference view. In both methods, the more visible features, the better the precision of the estimation. This difference is directly related to the number of parameters estimated.

The early image processing presented here for the bio-inspired visual compass are a bit too simplistic, particularly the segmentation algorithm which is too dependent on colored objects. It proved sufficient for real robot navigation, but according to recent experiments it is the first thing that needs improvement to do real robot orientation. The calibrated method normally uses more robust, though heavier, corner extraction techniques (although black spots were used for this experiment).

There is also room for improvement in the aggregation step of the bio-inspired method. We will test a modified aggregation algorithm, that weights bearing contributions according to their denominator. The calibrated method uses robust minimization algorithms which, again, are costly. An adaptation of those iterative techniques to robotics would probably reduce its cost by spreading calculation in time:

making a quick rough estimate and using as an input on next frame acquisition, which will at each new frame refine the position found.

Despite all the contrast between techniques, in complexity and computational cost, both methods produce comparable results in this experiment. The error of the calibrated method is one order of magnitude smaller than the bio-inspired method, which is an expected result, showing the benefit of the more costly methods. Not all robotics applications need such a great precision, though.

Of course all methods fail when their prerequisites are not satisfied. Since the bio-inspired method relies on local linearity of the variation of azimuths with respect to the animat motion, it fails when this assumption becomes too approximative. This happens in particular if the animat approaches one of the landmarks too closely: the associated azimuth varies more than during the observation because of shorter distance, causing I constancy to fail. In a real robotic application, an enhanced aggregation technique would ignore the too close landmark because it only results in outliers during the aggregation, but this would succeed only if enough other bearings are available. Another possibility, instead or in combination with the previous one, would be to call upon the complementarity of the visual compass property (observe once, use many times) with other source of information, for example odometry (used locally only to avoid drifts) to make other observations and extend the range of the visual compass as much as needed, including the information in a topological map, like Franz et al. (1998a) did with a different orientation strategy.

While we already used the same segmentation technique in Gourichon et al. (2002), we noticed that the bio-inspired visual compass is sensitive to wrong segmentations and wrong matchings, while the insect-inspired navigation strategies are usually very robust and forgiving in this matter. We think that there is an interesting parallel with the results of Svoboda and Sturm (1996) showing that the noise in the calibration parameters affects rotation estimation more than translation estimation.

Due to its simplifying assumption of a sole rotation around a vertical axis, the bio-inspired method makes itself vulnerable to situations where the motion is not even. It will fail if a rotation around a different axis occurs, much the same way many navigation models fail when they wrongly assume that they

know their correct orientation, because the linear relation that is called upon will be disrupted and irrelevant. By contrast, the calibrated method should benefit from the more thorough work it does, by being able to deal with any rotation. In both contexts, gravity may help if properly used (inertial navigation systems and/or low-cost mechanical arrangement maintaining the robot "head" upwards).

Synthetically, all this agrees with the common wisdom that robotics can take advantage of computationally much lighter methods than the ones used for 3D environment reconstruction. If an animat can deduce with a very cheap method an approximate direction to go, sufficient to reach the goal, there is no need for a computationally expensive one. On the contrary, the functional view, common in the 3D reconstruction paradigm, where pictures taken off-line are then fed into a mathematical algorithm that produces a reconstruction, cannot afford approximative results. There is room for hybrid methods, for example where a machine aiming at a 3D reconstruction would decide to move in a certain direction to overcome a perceptual ambiguity or refine a measurement.

In future work, we will test the visual compass on a real robot.

For the bio-inspired compass we will probably need to improve either the segmentation algorithm, the matching algorithm or add a method for selecting triplets that produce the most consistent results, to cope with an environment less easy than the test environment used here.

For the calibrated compass, it could be interesting to mathematically express the assumption about rotation that is done in the other method, and assess if the resulting simplification allows a better precision in difficult situations where too few points are available for example.

We are considering the use of the bio-inspired visual compass as an external direction estimator for the biomimetic models of head-direction cells currently developed by Angelo Arleo at LPPA, Collège de France.

After the visual compass is validated for itself in a robotic context, it should be tested as a complement to a navigation system that needs an orientation estimation, for example the work of Filliat (2001).

We noticed that the bio-inspired visual compass could be extended directly to also estimate a translation, at the cost of knowing the motion performed

during observation.

7 Conclusion

The issue of estimating the motion between two real-world views is simultaneously interesting for 3D world reconstruction as well as for robotics.

In this article, we presented and compared two new methods that estimate the orientation of a panoramic view with respect to a reference view in an *a priori* unknown environment, one biologically inspired in the continuation of the work on insect-inspired robot navigation strategies, the other in the continuation of research on 3D scene reconstruction.

Due to their different origins, the two methods differ in input used, results produced, complexity and computational cost. The 3D computer vision approach makes an extensive use of sharp models of sensors to retrieve a high precision. From a bio-inspired perspective, the calibrated method needs too much information and produces more results and with greater precision than really needed for a robotics application.

We think that there is room for inspiration from each other field, that would either offer more robustness where needed in the bio-inspired field, of more flexibility in the classical field by taking advantage of closer interaction with the real world.

References

- S. Baker and S. K. Nayar. A theory of catadioptric image formation. *ICCV*, pages 35–42, 1998.
- B. A. Cartwright and T. S. Collett. Landmark learning in bees. *J. Comp. Physiol*, 151:521–543, 1983.
- B. A. Cartwright and T. S. Collett. Landmark maps for honeybees. *Biol. Cybern.*, 57:85–93, 1987.
- P. Chang and M. Hebert. Omnidirectional structure from motion. In *Proceedings of IEEE workshop on omnidirectional vision*, 2000.
- O. Faugeras. *Three dimensional computer vision : a geometric viewpoint*. MIT press, 1993.
- O. Faugeras and S. Maybank. Motion from point matches :multiplicity of solutions. Technical Report RR-1157, INRIA, february 1990.
- Olivier Faugeras, Long Quan, and Peter Sturm. Self-calibration of a 1d projective camera and its application to the self-calibration of a 2d projective camera. In *Computer Vision - ECCV'98, 5th European Conference on Computer Vision, Freiburg, Germany, June 2-6, 1998, Proceedings, Volume I*, volume 1406 of *Lecture Notes in Computer Science*. Springer, 1998. ISBN 3-540-64569-1.
- David Filliat. *Cartographie et estimation globale de la position pour un robot mobile autonome*. PhD thesis, Université Pierre et Marie Curie (Paris VI), 2001.
- M. Franz, B. Scholkopf, P. Georg, H. Mallot, and H. Bulthoff. Learning view graphs for robot navigation. *Autonomous Robots*, 5:111–125, 1998a.
- Matthias O. Franz, Bernhard Schölkopf, Hanspeter A. Mallot, and Heinrich H. Bulthoff. Where did i take that snapshot? scene-based homing by image matching. *Biological Cybernetics*, 79:191–202, 1998b.
- C. Geyer and K. Daniilidis. Catadioptric projective geometry. *International journal of computer vision*, december 2001.
- C. Geyer and K. Daniilidis. Paracatadioptric camera calibration. *IEEE Transaction on pattern analysis and machine intelligence*, 24(4), april 2002.
- Stéphane Gourichon. A biologically inspired visual compass. Technical report, Université Pierre et Marie Curie (Paris 6), 2003.
- Stéphane Gourichon, Jean-Arcady Meyer, and Patrick Pirim. Using coloured snapshots for short-range guidance in mobile robots. *International Journal of Robotics and Automation*, 17(4):154–162, 2002.
- C. Harris and M. Stevens. A combined corner and edge detector. In *Proceedings of the fourth Alvey Vision Conference*, pages 153–158, 1988.
- Sio-Hoï Ieng, Ryad Benosman, and Jean Devars. Multi-angular dynamic matching for catadioptric views. In *Proceedings of ICCV2003*, september 2003. to appear.
- K. Ikeuchi, M. Sakauchi, T. Takahashi, M. Murao, I. Sato, and F. Kai. Panoramic image analysis for constructing virtual cities. *IPSJ Transaction on*

- R. Benosman J. Fabrizzio, J.-P. Tarel. Calibration of panoramic catadioptric sensors made easier. In *Proceedings of the IEEE workshop on omnidirectional vision*, 2002.
- H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, september 1981.
- P. J. Narayanan and T. Kanade. Virtual worlds using computer vision. In *Proceedings of the IEEE and ATR workshop on computer vision for virtual reality based human communications*, pages 2–13, january 1998.
- M. E. Spetsakis and J. Aloimonos. Structure from motion using line correspondences. *Int. Journal of Computer Vision*, 4(3):171–183, May 1990.
- T. Svoboda and T. Pajdla. Matching in catadioptric images with appropriate windows and outliers removal. *CAIP*, pages 733–740, 2001.
- T. Svoboda, T. Pajdla, and V. Hlavac. Motion estimation using central panoramic cameras. In *Proceedings of the IEEE International conference on Intelligent Vehicules*, pages 335–340, october 1998.
- T. Svoboda and Peter Sturm. What can be done with a badly calibrated camera in ego-motion estimation? Technical Report CTU-CMP-19, Center for Machine Perception, Czech Technical University, Prague, Nov 1996.
- C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *International Journal of Computer Vision*, 9(2), 1992.
- Z. Zhang. Determining the epipolar geometry and its uncertainty. *International Journal on Computer Vision*, 27(2):161–195, 1998.

Perception of Human Periodic Motion in Moving Light Displays

Baihua Li Horst Holstein

Department of Computer Science

University of Wales, Aberystwyth, SY23 3DB, UK.

bbl00@aber.ac.uk hoh@aber.ac.uk

Abstract

We present a feature-based frequency-domain technique for modelling and recognizing human periodic movements in moving light displays (MLDs). Periodic motions are modelled by motion templates composed of a set of feature power vectors. Feature power vectors are extracted from unidentified trajectories of feature points using motion power spectral analysis. Motion recognition therefore is carried out in the frequency domain by finding a best match among pre-stored templates. This method contrasts with common spatio-temporal approaches. Recognition is demonstrated through examples of human periodic motion with data obtained from a marker-based motion capture system.

Keywords: periodic motion recognition, motion-based recognition, Moving Light Displays (MLDs), power spectral analysis and FFT.

1 Introduction

The ability of humans to perceive structure and motion from sparse point feature motion cues has been demonstrated by Johansson's Moving light displays (MLDs) (Johansson, 1975). In the MLDs as shown in Fig.1, an image sequence was reduced to a set of moving light dots. These light dots were attached at the joint sites of a human subject, contrasted to a dark background. The dots acted as discrete feature-points presenting motion characteristics in spatio-temporal domain. The MLDs carried only motion information but no structural information, since the displayed points were discrete and unconnected. One frame of static dots remained meaningless to observers, while human observers were able to recognize activities such as walking, running or stair climbing from a sequence of relative movements of a small number of lights. Barclay et al. (1978), Cutting and Kozlowski (1977) also showed human observers can identify the actor's gender and even their friends by their gaits in MLDs. These pioneering works in psychology relating to human motion perception suggest that feature-based motion presentation plays an important role in recognition tasks. In the case of machine vision, the biological metaphor suggests that it is possible to use the reduced spatio-temporal information, such as embedded in MLDs, for recognition.

MLDs are used in studies of perception and motion classification (Johansson, 1975; Barclay et al., 1978; Cutting and Kozlowski, 1977; Campbell and Bobick, 1995; Goddard, 1992; Boyd and Little, 1997), clinical gait analysis (Angeloni et al., 1994), and more recently in humanoid robot design (Hill and Pollick, 2000). Motion analysis from MLDs uses concise and accurate data to

investigate what are the essential recognition features in modelling motion. This research is applicable in areas such as computer vision, biomedical studies, sports science etc., related to, for example, activity recognition, kinematics and motion analysis and synthesis.

In the last two decades, MLD perception theory has emphasized the recovery of structure from motion, employing derived concepts such as joint angle or phase. For example, Campbell and Bobick (1995) classified ballet dance steps using a phase space representation. The phase space was related to each degree of freedom (DOF) in identified human articulated structure using MLD data. Goddard (1992) proposed a computational model for visual motion recognition of gaits in walking, jogging, and running in MLDs. He used the joint angles and angular velocities as features for recognition. A difficulty in this work was the prior necessity to identify individual points in the images. He has argued the possibility for a perception directly from motion information. More works related to MLDs can be found in survey articles (Cédras and Shah, 1994; Cedras and Shah, 1995).

Very few researchers have attempted motion recognition directly from motion information provided by MLDs. However, recent work by Boyd and Little (1997), using global shape-of-motion features derived from MLD images, has shown that it is possible, by non-structural means, to recognize individual people by their gait. This approach avoids the complex vision problem of kinematic structure recovery.

In this paper we propose a frequency domain approach for recognition of human periodic movements in MLDs. We believe that common periodic movements, such as walking, running and skipping, present human body seg-

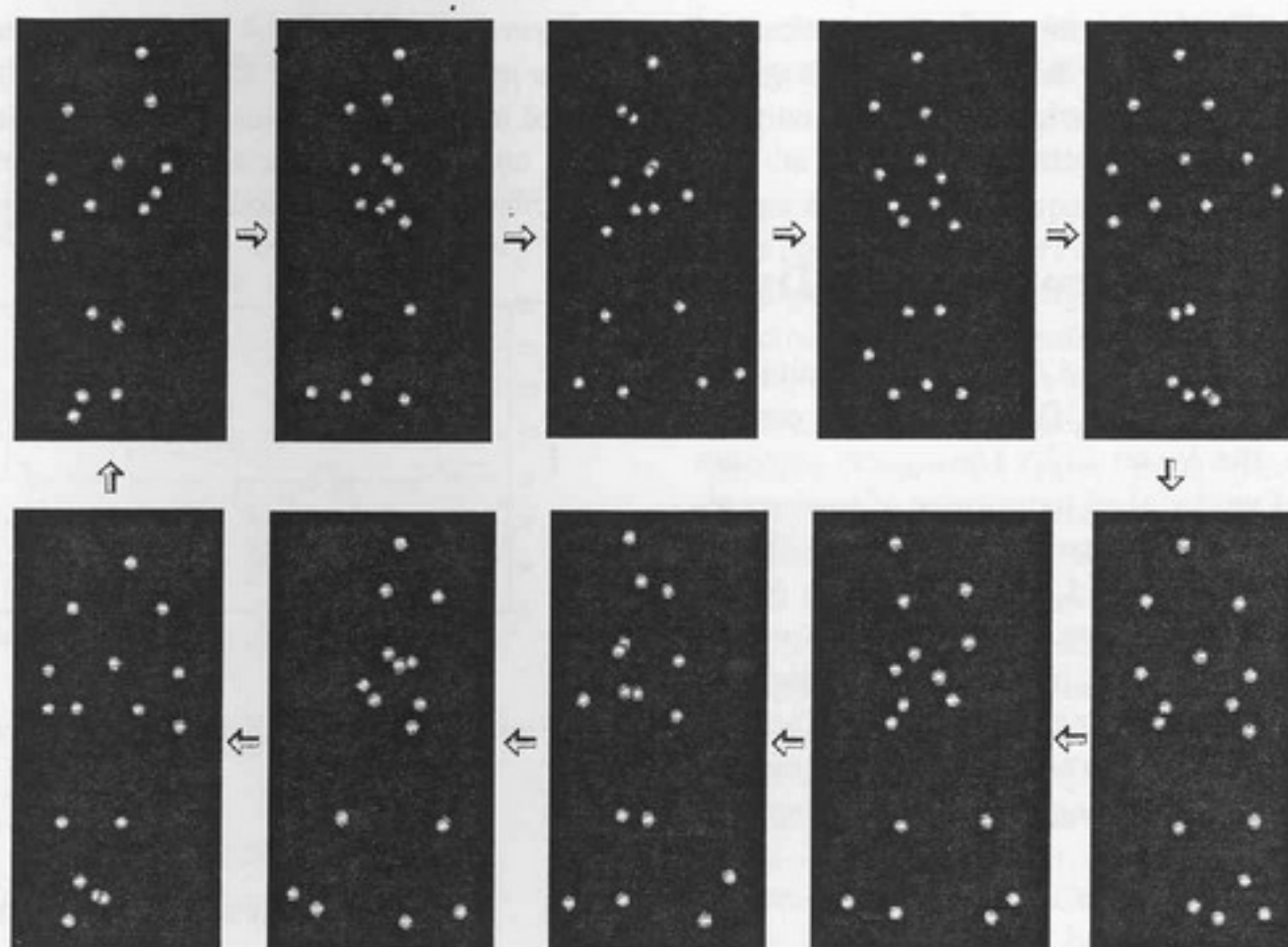


Figure 1: A clockwise circle-walking person in MLDs with 16 feature points.

ment movement in relative harmony. These motion characteristics exist not only in the spatio-temporal domain, but also in frequency domain. They are preserved even in the reduced information of MLDs, allowing motion recognition without knowledge of underlying structure.

The rest of the paper is organized as follows: Section 2 reviews related work on cyclic motion recognition. Section 3 states our method of data collection. Section 4 describes the frequency domain approach for recognition. Section 5 provides experimental results on recognition of human cyclic motion. We discuss and conclude our work in sections 6 and 7.

2 Related work

Human motion, specifically walking, has been studied extensively using various spatio-temporal cues from images or MLDs. Approaches directly using motion for periodic motion recognition are described in e.g. Polana and Nelson (1993), Tsai et al. (1994), Fujiyoshi and Lipton (1998), Boyd and Little (1997). Polana and Nelson (1993) propose a method for detecting periodic motion using Fourier transforms on several point trajectories. They indicated that in principle, the period of the movement could be inferred from averaging the fundamental frequencies of the point trajectories. Tsai et al. (1994) use the trajectory of one point on an object that performed some cyclic motion to compute curvature. An autocorrelation is performed to enhance self-similarity within the curvature function. The Fourier Transform is finally used

to detect the presence of a cycle and its period from the spatio-temporal curvature. Fujiyoshi and Lipton (1998) generate a "star" skeleton from the object boundary. They apply Fourier analysis to the skeleton for detecting periodic motion. Then they utilize both posture and motion cycle of the "star" skeleton to recognize activities such as walking and running.

In these approaches, Fourier transforms are usually used to detect or recognize periodicity. The detected periodicity is used to assist motion recognition. For spatio-temporal domain approaches, in order to deal with the problems of e.g. human motion irregularities or change in speed, techniques such as scale space or Dynamic Time Warping (DTW), considered computational expensive, are usually used to match portions of scale space to find repeated patterns from identified curves.

The direct application of frequency domain analysis for motion recognition has received much less attention. Works emphasizing frequency domain analysis are, for example, Angeloni et al. (1994), Köhle and Merkl (1996). Angeloni et al. (1994) use gait kinematic data from MLDs to analyze the frequency content of whole body movement. Their work presents the characteristic spectral distribution among articulated body parts. Köhle and Merkl (1996) demonstrate that the kinetic data from ground reaction force platforms can also be used to classify gait patterns in clinical gait analysis, through Fourier transforms of vertical force components and classification by self-organizing maps. The works of both Angeloni and Köhle show that motion frequency spectra may include cues suitable for motion recognition.

We are pursuing the development of interpretation and recognition of human periodic motion. We investigate a pure frequency domain approach to model human periodic motion using kinematic data from MLDs.

3 Data collection from 3D-MLDs

All human kinematic data used in our investigation are acquired from a marker-based 3D optical motion capture (MoCap) system, the Vicon 512. The system provides 3D coordinates of unidentified trajectories of markers attached to a subject, in the manner of a 3D-MLD system. The data are not affected by the projective distortions of particular camera views. In this respect, we differ from other classical MLD investigations, which detect data from 2D projected image sequences. The high quality of our data allows us to apply a frequency domain approach for direct motion recognition from the captured data.

In our motion capture system, the world coordinate system has its origin on the ground. The X-Y plane is parallel to the ground plane, and the Z-axis is vertical. Other conditions for data collection in our experiments are:

- Motions are captured in a control volume, about 4m(length)×4m(width)×2.5m(height). The measurement accuracy is to the level of a millimeter.
- Sixteen markers, regarded as feature points, are attached on human subjects at the following locations: head, torso, shoulders(L/R), elbows(L/R), wrists(L/R), AISs (on L/R hips), knees(L/R), ankles(L/R), toes(on each left and right foot). They are effective in indicating motion cues in MLDs.
- The trajectories are nearly always uninterrupted. Some small trajectory gaps arising from occlusion are filled by interpolation during post-processing.
- The sample rate is $f_s = 60Hz$. The length of each trial is $N \approx 350 \sim 1300$ frames.
- The correspondence between a 3D trajectory and the marker identity is not assumed to be known in the motion to be identified.

4 Method

The relative movements of feature points contain information of both motion and identity. For cyclic activities carried out on a level (horizontal) floor, the vertical components, which are the Z-coordinates of 3D-MLD trajectories, provide a simple input for Fourier analysis. They can be used without transformation, because they contain no secular motion component. In this study, we use the Z-data as the only motion cue to be analyzed. An example right toe (hallux) Z-trajectory of a walking person is

shown in Fig.2. In Fig.4 and Fig.5, we have indicated feature point identity for clarity, but this information is not used in the identification process. We find that cues from the unidentified tracks suffice to discriminate between a number of simple periodic human activities.

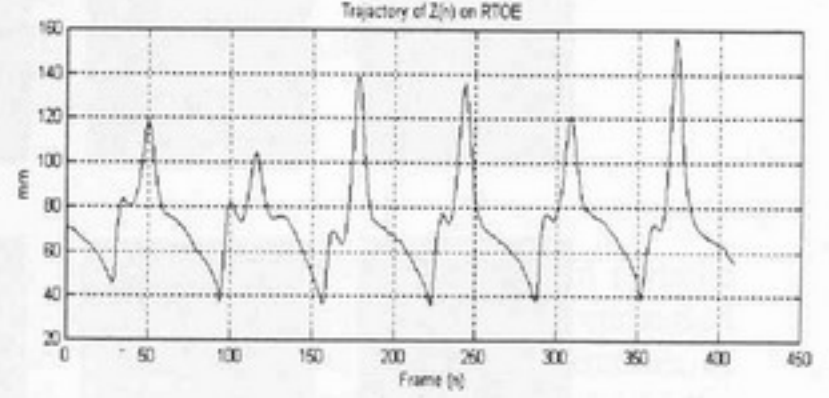


Figure 2: Right toe Z-trajectory of a walking person.

4.1 Power spectral analysis for whole body movement

We apply spectral analysis to a vertical-component trajectory $Z_{i(n)}$ in frames $n = 1 \dots N$, of each feature point $i = 1 \dots I$. The decomposition is expressed by

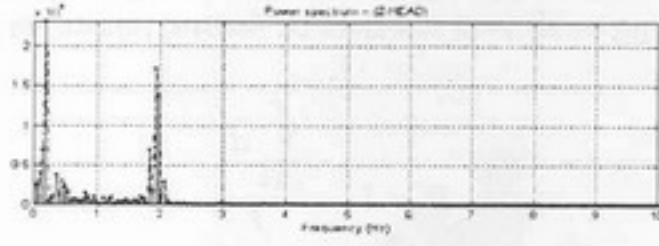
$$Z_{i(n)} = \frac{1}{2}a_{i(0)} + \sum_{k=1}^{N/2} a_{i(k)} \cos(2\pi nk\Delta f) + b_{i(k)} \sin(2\pi nk\Delta f) \quad (1)$$

where f_s is the sample rate and $\Delta f = f_s/N$ denotes the frequency resolution. The DC component $\frac{1}{2}a_{i(0)}$ denotes the average vertical position of this point relative to the origin on the ground during movement. When the sequence length N is an exact power of two, we adopt the Fast Fourier Transform (FFT) algorithm which is more efficient than the raw Discrete Fourier Transform (DFT). The power spectrum for the k th frequency multiple of feature trajectory i is

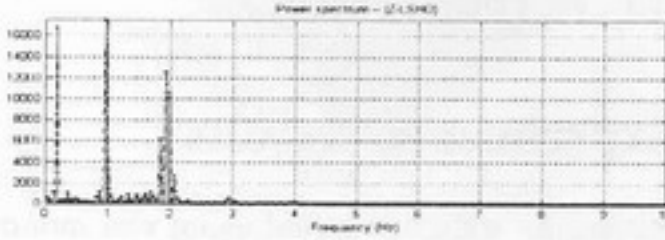
$$P_{i(k)} = a_{i(k)}^2 + b_{i(k)}^2, \quad k = 1, 2 \dots N/2 \quad (2)$$

Examples of such power spectra for a clockwise circle-walking person are given in Fig.3, in which $N=1024$, $f_s = 60Hz$, and the gait-cycle frequency is about 1Hz.

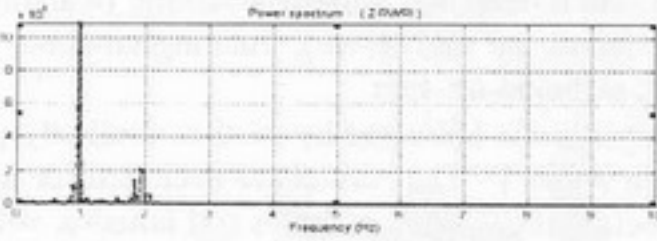
From power spectral analysis of whole-body feature points for a number of common cyclic movements, such as walking, running, jumping, skipping, we find the dominant power of human movements occupies only a narrow bandwidth, with the upper limit about 10Hz. The power spectral distribution shows clustering around some frequencies related to the fundamental activity frequency, such as the gait-cycle frequency in walking and running, and its multiples. The magnitude of the spectral component is time shift invariant. It retains its spectral envelope regardless of where in time the FFT is performed. We also find that power magnitude and spectral pattern are



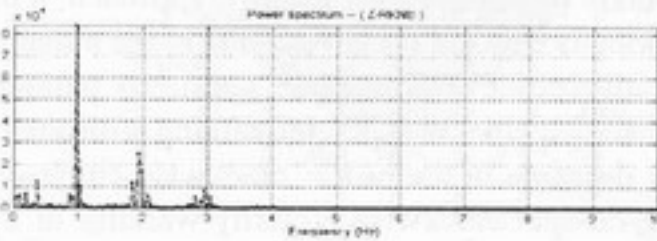
(a) head



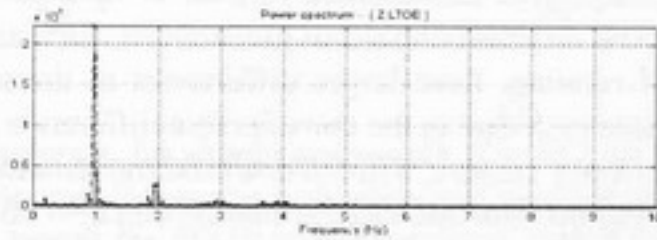
(b) left shoulder



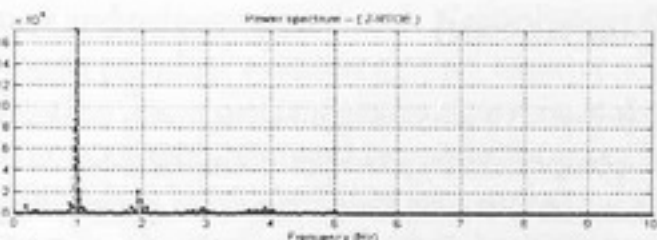
(c) right wrist



(d) right knee



(e) left toe



(f) right toe

Figure 3: Examples of vertical-component power spectra of a clockwise circle-walking person.

reflected by the active body parts. This is to be expected since they undergo different intensity and character of motion. For example, as shown in Fig.3, the spectrum for the head has a small power total and has a very different distribution compared with the spectra of toes. We also observe the power components of left toe in Fig.3(e) is larger than that of right toe in Fig.3(f), though their spectral patterns are similar. This is because in clockwise walking, left foot has a larger movement than right foot. For the same kind of motion in different subjects, spectral patterns of the same feature points are similar, hinting at the motion nature, differences being related to variation in individual speed and amplitude.

To obtain speed-invariant representation for the same kind of movement, we normalize these spectra to a fundamental activity cycle or generalized gait cycle G_c . To achieve an accurate G_c , we sum power spectra within a band-limited frequency $[0.4 \sim 5\text{Hz}]$. The frequency $G_c = k\Delta f$ corresponding to the maximum power magnitude in the summed spectra over feature points i ,

$$\max_k \left\{ \sum_i P_{i(k)} \right\} \quad (3)$$

is regarded as the activity cycle, or generalized gait-cycle (G_c).

The cycle frequency is subsequently used to normalize the frequency axis from Hz to generalized gait cycle (G_c). Fig.4 shows an example of G_c -scaled whole-body power spectra of a walking person.

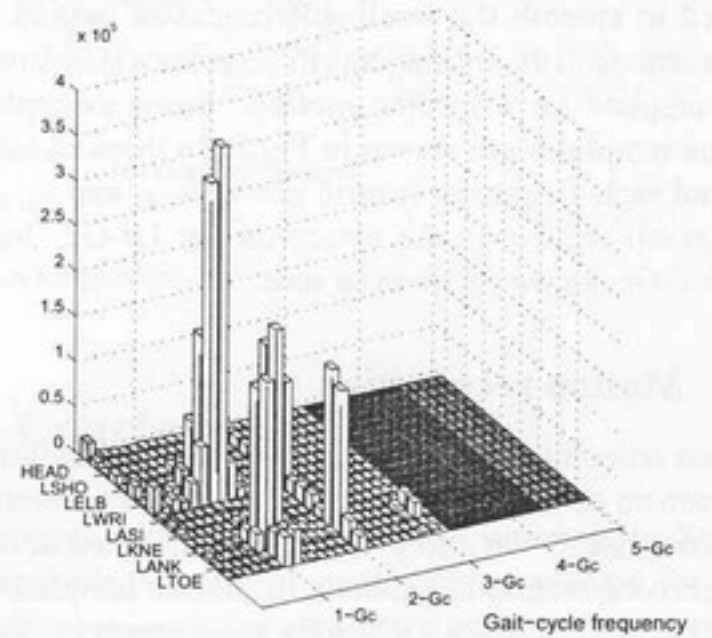


Figure 4: G_c -scaled whole-body power spectra of a walking person.

4.2 Feature power vector and motion template

In order to achieve an adequate frequency resolution $\Delta f = f_s/N$, long sequences are required. Moreover, a uniform frequency resolution between different trials is

needed in order to make a direct spectral comparison feasible. However, in practice, a random length sequence N may contain only a few cycles and result in an imprecise spectrum and different frequency resolution.

To mitigate this problem, we extract a set of powers from spectrum of each trajectory, regarded as a feature power vector, as shown in Eq.4,

$$\begin{aligned}\nu_{i(0)} &= DC_i \\ \nu_{i(n)}|_{n=1,2,\dots,5} &= \sum_{k \in W_n} P_{i(k)} \\ \nu_{i(6)} &= \sum_{k \leq 6Gc, k \notin W_{1,2,\dots,5}} P_{i(k)}\end{aligned}\quad (4)$$

where $\Delta Gc = \Delta f / Gc$ is the Gc-scaled frequency resolution. The first component $\nu_{i(0)}$ of the feature power vector $\vec{\nu}_i$ of each feature point i is its DC component. For $\nu_{i(1)}$ to $\nu_{i(5)}$, we utilize a shift sum-window $W_{1,2,\dots,5}$ to sum three power components around each of the 1st-Gc, 2nd-Gc ... 5th-Gc frequencies respectively. The last item $\nu_{i(6)}$ is used to compensate small power components lying up to the 6th Gc that have not yet been taken into account. Using feature power vectors also reduces the number of frequency components to be considered for matching.

We stack I feature power vectors of feature points together as a motion template, $V = \{\vec{\nu}_i | i = 1 \dots I\}$. The template can be viewed as an $I \times 7$ array. Each column in the motion template is scaled relative to the maximum value in this column. That means to normalize the power amplitude of individual motion. An averaging technique is used to smooth the small differences in motion templates among different subjects to generate a standard motion template for a specific motion. Some examples of motion templates are shown in Fig.5. In these examples, we find high-frequency feature power $\nu_{i(4)}$ and $\nu_{i(5)}$ are very small relative to the power on the 1st-Gc, 2nd-Gc and 3rd-Gc, so we set them to zero.

4.3 Motion recognition

Motion recognition is achieved by finding the best match between an observed motion template and pre-stored motion templates. We apply the algorithm stated above to an observed motion to generate its motion template $U = \{\vec{\mu}_j | j = 1 \dots J\}$, with J feature power vectors. We use an $I \times J$ match matrix $M = \{m_{(i,j)} | i = 1 \dots I, j = 1 \dots J\}$, see Eq.5, to store the weighted-difference between the i th motion vector $\vec{\nu}_i$ in a model template and the j th motion vector $\vec{\mu}_j$ in the observed template,

$$m_{(i,j)} = \sum_{n=0}^6 |\nu_{i(n)} - \mu_{j(n)}| \omega_n \quad (5)$$

where $\vec{\omega} = [\omega_0, \omega_1, \dots, \omega_6]$ is the weight vector. In our experiments, $\vec{\omega} = [0.4, 0.25, 0.25, 0.05, 0, 0, 0.05]$. The best match of point j should be the point corresponding to the minimum element $d_j = \min_i \{m_{(i,j)}\}$ in a row of

M . Subsequently, whole motion similarity S is measured according to all best matches of feature points, as shown in Eq.6.

$$S = 1 - \frac{\sum_{j=1}^J d_j}{J} \quad (6)$$

The motion with maximum similarity S for all the searched templates is taken to indicate recognition. The feature points of the observed motion can be an adequate subset ($J \leq I$) of standard templates.

5 Experimental Results

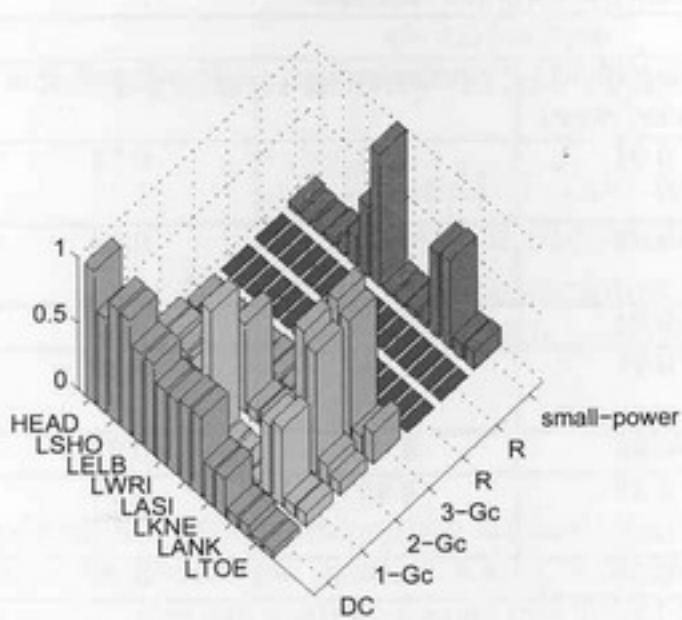
All experiments were conducted using real motion capture data from a marker-based optical motion capture system, the Vicon 512, as described in section 3. Recognition was tested on simple periodic activities, namely walking-on-spot, circle-walking, butterfly-walking (walking with waving hands up and down), running-on-spot, circle-running, skipping-on-spot.

Recognition is indicated by motion similarity parameter S in Table 1. The results are averaged for a group of subjects that consisted of males and females, with ages from 5 to 60 years. The highest column entries, highlighted, occur when the observed activity matches the motion template activity. Because the algorithm utilizes a whole-body power spectral analysis approach, a detailed recognition is realized by comparison of the movement on each feature point. For example, similarity parameters S of an observed clockwise circle-walking with each model activity decrease in the order: clockwise circle-walking, walking-on-spot, clockwise butterfly-walking in a circle, circle-running, running-on-spot, skipping-on-spot.

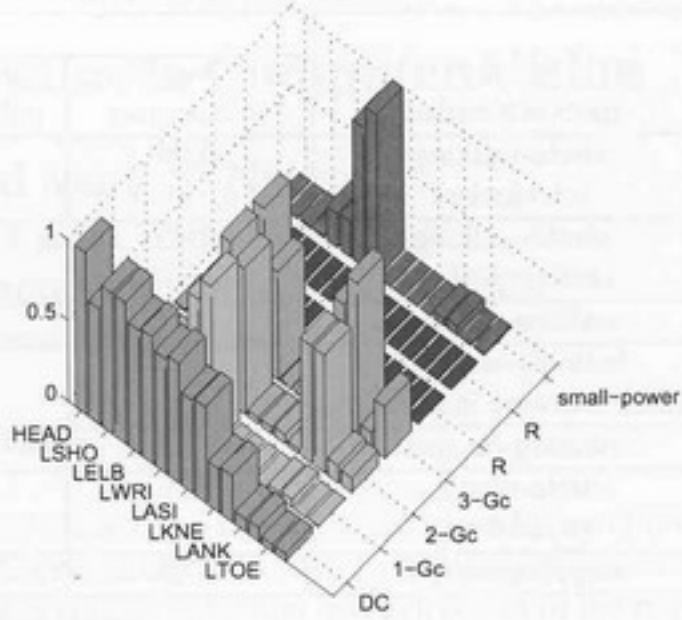
From the results above, we find that the proposed power spectral analysis approach is able to classify cyclic motions solely on unidentified vertical-component trajectories. The different classes of movements, such as walking and running, have larger differences in the similarity parameter S due to the considerable difference among feature power vectors; while the similar movements, such as running-on-spot and circle-running, can also be distinguished, because the magnitudes of power spectra on left limbs differ from that on right limbs in circular activities.

6 Discussion

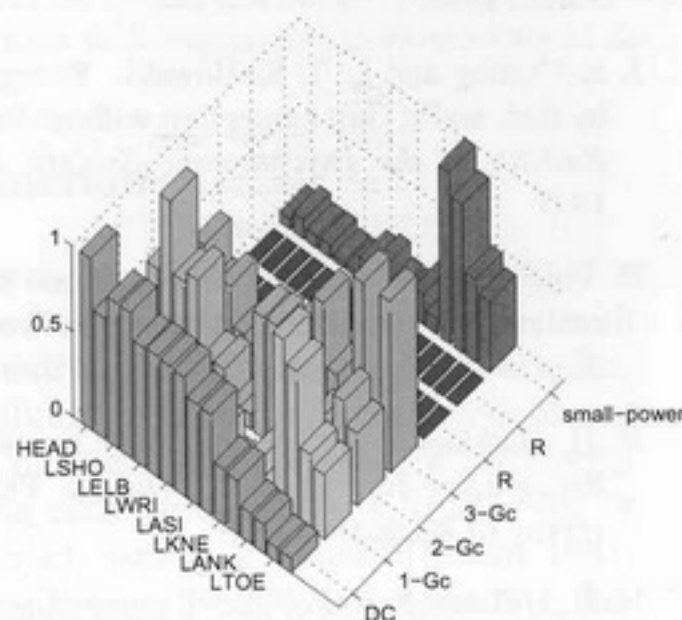
Our experiments assumed availability only of unidentified vertical-component trajectories obtained from 3D-MLDs. Normalization both on frequency and power magnitude allows recognition to be carried out in the frequency domain for a wide range of subjects. We can combine this approach with activity periodicity assisted recognition $S' = \{S, \xi_{Gc}\}$, where ξ_{Gc} is a activity periodicity (e.g. Gc) related recognition factor, that may present a more precise recognition.



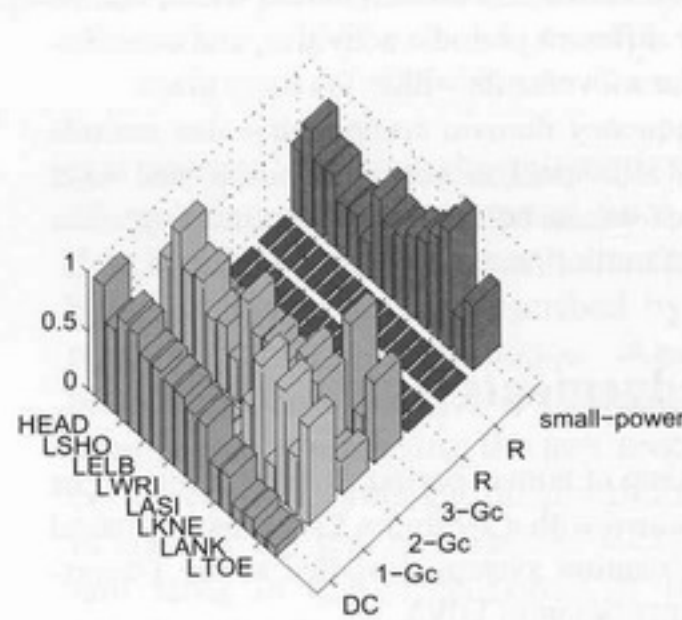
(a) anticlockwise circle-walking



(b) butterfly anticlockwise circle-walking



(c) running-on-spot



(d) skipping-on-spot

Figure 5: Examples of motion templates.

In practice, for shorter sequences $N < 512$, to guarantee a desired frequency resolution, we suggest using the whole length for the Fourier Transform, rather than cutting it to a length of an exact power of two. In this case, the FFT is reduced to the raw Discrete Fourier Transform (DFT) algorithm. This is a tradeoff between frequency resolution and computational cost. Nevertheless, because each motion template contains a small number of parameters by using feature power vector analysis, and because we also only use one standard template for indexing each kind of motion, a short matching time and small database are required. The method proved to be computationally efficient.

7 Conclusions

A frequency domain approach based on whole-body spectral analysis is proposed as an efficient method for modelling and recognizing articulated periodic activities. We address a motion-based approach using unidentified trajectories of feature points from MLDs. We apply power spectral analysis on each vertical-component trajectory and detect a set of feature power vectors. These feature power vectors are used to generate a motion template for given periodic activity, averaged for a number of subjects. Recognition of an observed motion involves comparison with stored motion templates to find a best match.

The experimental results indicate that using features derived from frequency domain analysis is successful for classifying periodic motions without identification of kinematic structure. It shows the inherent characteristics

Table 1: Recognition of human periodic movements.

activity motion template	observed activity			
	circle-walking (clockwise)	circle-walking (anticlockwise)	running-on-spot	skipping-on-spot
circle-walking (clockwise)	0.98	0.92	0.85	0.84
circle-walking (anticlockwise)	0.93	0.98	0.86	0.83
walking-on-spot	0.94	0.94	0.89	0.85
butterfly-walking (anticlockwise in a circle)	0.92	0.91	0.86	0.86
running-on-spot	0.86	0.88	0.97	0.95
circle-running (clockwise)	0.89	0.87	0.95	0.93
skipping-on-spot	0.85	0.86	0.94	0.97

of human periodic movements exist not only in the spatio-temporal domain but also in the frequency domain. Frequency domain features hint motion nature which can be used to classify different periodic activities, and even discriminate similar movements within the same class.

Whether frequency domain components also include information for individual subject recognition, and what kinds of features would be necessary for a more precise interpretation of motion, are still to be investigated.

Acknowledgements

All 3D-MLDs data of human periodic motion used in this paper were obtained with a 7-camera Vicon marker-based optical motion capture system, installed at the Department of Computer Science, UWA.

References

- C. Angeloni, P. O. Riley, and D. E. Krebs. Frequency content of whole body gait kinematic data. *IEEE Trans. Rehabilitation Engineering*, 2(1):40–46, March 1994.
- C. D. Barclay, J. E. Cutting, and L. T. Kozlowski. Temporal and spatial factors in gait perception that influence gender recognition. *Perception and Psychophysics*, 23(2):145–152, 1978.
- J. Boyd and J. Little. Global versus structured interpretation of motion: moving light displays. In *Proc. IEEE CVPR*, 1997.
- L. Campbell and A. Bobick. Recognition of human body motion using phase space constraints. In *Proc. IEEE Int. Conf. Computer Vision*, pages 624–630, Cambridge, 1995.
- C. Cédraş and M. Shah. A survey of motion analysis from moving light displays. In *Proc. IEEE CVPR'94*, pages 214–221, Washington, June 1994.
- C. Cedras and M. Shah. Motion-based recognition: A survey. *Image and Vision Computing*, 13(2):129–155, March 1995.
- J. E. Cutting and L. T. Kozlowski. Recognizing friends by their walk: Gait perception without familiarity cues. *Bulletin of the Psychonomic Society*, 9(5):353–356, 1977.
- H. Fujiyoshi and A. Lipton. Real-time human motion analysis by image skeletonization. In *Proc. IEEE Workshop on Applications of Computer Vision*, 1998.
- N. H. Goddard. *The Perception of Articulated Motion: Recognizing Moving Light Displays*. PhD thesis, University of Rochester, 1992.
- H. H. Hill and F. E. Pollick. Exaggerating temporal differences enhances recognition of individual from point light displays. *Psychological Science*, 11:223–228, 2000.
- G. Johansson. Visual motion perception. *Scientific American*, 232(6):75–80, 85–88, 1975.
- M. Köhle and D. Merkl. Things we observed when watching people walk: Classification of gait patterns with self-organizing maps. In *Proc. 7th Australian Conf. Neural Networks ACNN'96*, Canberra, April 10–12 1996.
- R. Polana and R.C. Nelson. Detecting activities. In *Proc. IEEE CVPR*, pages 2–7, 1993.
- P. S. Tsai, M. Shah, K. Keiter, and T. Kasparis. Cyclic motion detection for motion based recognition. *Pattern Recognition*, 27(12):1591–1603, December 1994.

On Laboratory Gait Analysis via Computer Vision

Jang-Hee Yoo and Mark S. Nixon

Department of Electronics and Computer Science
University of Southampton, Southampton SO17 1BJ, UK
[jhy00r|msn]@ecs.soton.ac.uk

Abstract

We describe a marker-less system for analysing and classifying human gait motion by combining a statistical approach and motion tracking with topological analysis guided by anatomical knowledge. The marker-less gait analysis system consists of three stages: detection and extraction of the moving human body and its contour from image sequences; extraction of human gait motion by the joint angles and body points; and kinematic analysis and feature extraction for classifying the gait pattern. The periodic motion of human gait is described by symmetry, and a 2D stick figure is used to represent the human gait model. The usefulness of proposed method is demonstrated in marker-less gait analysis with comparison to biomechanical data.

1 Introduction

The ability to recognise humans and their activities by computer vision is a very important task, with many potential applications (Gavrila 1999). Human gait (walking and running) is known to be one of the most universal and complex of all human activities, and each person appears to have his or her own characteristic gait pattern (Inman 1981, Murray 1964, Winter 1991). As a biometric, human gait may be defined as a means of identifying individuals by the way they walk (Nixon 1999). Using gait has many advantages over other biometrics such as fingerprints, most notably that it is non-invasive and available at low resolution. However, most gait capture systems use external markers which represent the observed movement in the form of 3D trajectories that translate into kinematic variables such as body movements and joint angles (Kleissen 1998). Marker based systems can acquire precise motion information, but require specialised hardware and subject contact. Therefore, marker-less systems are essential to enable greater application capability.

In this paper, we propose a new marker-less system for analysing and classifying human gait by computer vision techniques. The large amount of human gait data was collected from DV (digital video) cameras, and the human body and its contour is extracted from the image sequences. The human gait cycle is detected by analysing gait symmetry points and used for finding important

gait phases. To extract the gait motion, body angles of each segment are extracted from gait skeleton data by linear regression analysis, and gait motion between key-frames is described by tracking the moving points of locomotion. Also, trajectory-based kinematic features are calculated from gait sequences for analysing the gait motion. Then, an enhanced back-propagation algorithm is employed to classify the gait features. As such, the new system aims to derive measures of biomechanical significance, in part demonstrated by recognition capability.

2 Describing Human Gait Motion

2.1 Human Gait Database

Early gait databases reported in the literature were collected under very limited conditions and consisted of a relatively small number of subjects. To overcome this, larger databases have recently been developed by DARPA Human ID projects (Phillips 2002). The SOTON database (Shutler 2002) developed by the ISIS Research Group is one of the recent databases in the DARPA project. The SOTON database comprises two forms of indoor data (under controlled lighting with special background) and one outdoors (without lighting or background control). The first form of indoor data is a subject walking on a treadmill, and the second form is subject walking alone a specially designed continuous track.

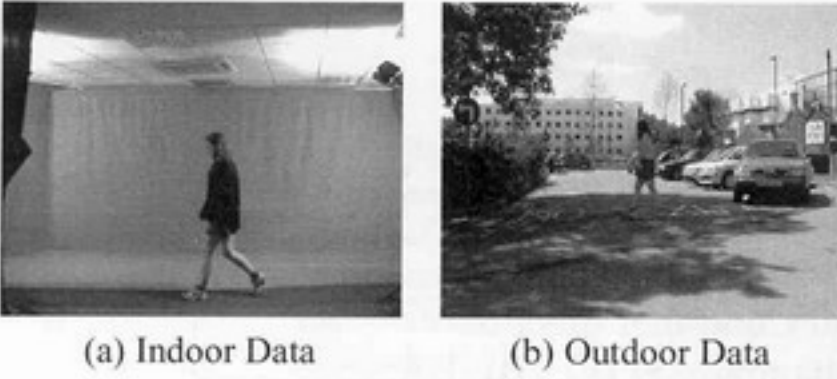


Figure 1: Images from SOTON Database

Sample images from the database are shown in Fig. 1. As can be seen in Fig. 1(a), chroma-key laboratory was constructed to allow controlled lighting conditions for the indoor treadmill and track data. An image sequence contains only a single subject walking at normal speed and was acquired at 25 frames / second with 720×576 colour pixels from a DV camera. Also, the subject was filmed walking in both directions, and the database contains more than 100 subjects. Accordingly, a marker-less analysis system can contribute to the debate concerning the use of a treadmill versus normal walking, for a large number of subjects.

In addition, human body extraction from the image sequences is easily achieved through background subtraction. After that, the histogram projection profiles are analysed to estimate the position of a human body, and the body region is verified by prior knowledge such as size and shape. Thresholding and morphology is then used to extract the contour of a detected human body. Here, a thresholding method based on similarity measures between the background and the object image is used (Yoo 2002). Finally, the body contour is extracted by subtraction followed by dilation and erosion. Thresholding is a simple image segmentation method, and erosion will remove the outer layer of pixels from an object.

2.2 Detecting Human Gait Cycle

During walking, the human body functionally divides into the passenger and locomotor units. The upper body is a relatively passive passenger unit that rides on a locomotor unit. The two lower limbs and pelvis are the anatomical segments that form the locomotor unit. Each gait cycle is separated into two distinct periods: the stance and the swing phases (Perry 1992). The stance phase is the entire period during which the foot is on the ground, and the swing phase begins as the foot is lifted from floor (toe-off). In general, the stance and swing phases have been divided into eight sub-phases: initial contact, loading response, mid-stance, ter-

minal stance, pre-swing, initial swing, mid-swing, and terminal swing.

On the other hand, the passenger unit must be balanced on a joint that is in space, therefore, it is moving in the plan of progression as an inverted pendulum rotates about the hip joint (Winter 1991). Namely, the centre of mass of the passenger unit will keep the maximum distance from front foot at the double supports, and it has minimum distance from the front foot at mid-stance (MT) or mid-swing (MW). Fig. 2(a) shows a horizontal centre of mass at chest region which is guided by anatomical data. To detect the gait cycle, the horizontal centre of mass in passenger unit is considered as a gait symmetry point. This gait symmetry point can be defined as

$$s_i = d_i / (\bar{d} + \sigma) \quad (1)$$

where $\bar{d}$ is mean value and σ is standard derivation of d_i during one gait cycle. Fig. 2(b) shows the curves of gait symmetry interpolated by trigonometric-polynomials (Yoo 2002) during a gait cycle. Important gait phases (three double supports and two single supports) can be detected by the obvious peak points (three maximum and two minimum). Some gait features such as gait frequency (f_0), amplitude (A), and walking velocity (v) can be also calculated by this analysis.

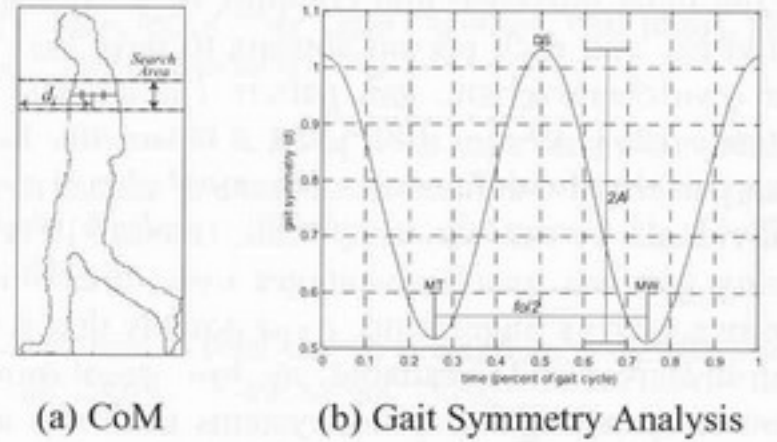


Figure 2: Detecting Gait Cycle

Human gait is a form of periodic motion, especially when walking laterally; hence we can predict human movement in a gait cycle. Murray (1964) considered human gait as “a total walking cycle” - the action of walking can be thought of as a periodic signal. Accordingly, the pattern of rotation angles around the joints has been modelled as a pendulum, the motion of which is characterised by simple harmonic motion. This assumes that the pattern of the motion is approximately sinusoidal in nature (Cunado 1997, Taga 1995). The sinusoidal functions are basically all periodic. Therefore, gait motion can be characterised as having a periodic component.

3 Extracting Human Gait Motion

3.1 Body Points Extraction

To extract body points in a contour image, a skeleton data with body segment properties is used. For a body height H , the vertical position of the neck, shoulder, waist, pelvis, knee and ankle was set by study of anatomical data (Dempster 1967, Winter 1990) to be $0.870H$, $0.818H$, $0.530H$, $0.480H$, $0.285H$, and $0.039H$, respectively. The gait skeleton can be simply calculated by two border points of each body part p with a range constraint as

$$x_{s,p} = (x_{b,p} + x_{e,p})/2 \quad (2)$$

where x_b and x_e represent the horizontal position of the begin and the end pixels on the horizontal line respectively. For example, the range constraint for trunk t can be given by mean value of neck's border points as

$$x_{s,t} = \begin{cases} x_{s,t} & \text{if } (\bar{x}_{b,neck} < x_{s,t} < \bar{x}_{e,neck}) \\ \emptyset & \text{otherwise} \end{cases} \quad (3)$$

In general, a primary gait skeleton is highly susceptible to difficulty in image estimation of the arm and foot movements. Therefore, a noise removal method using mean value and standard derivation in each body part is employed to select a skeleton point as

$$x'_{s,p} = \begin{cases} x_{s,p} & \text{if } (x_{s,p} \in \bar{x} \pm (\sigma + \alpha \cdot \sigma')) \\ \emptyset & \text{otherwise} \end{cases} \quad (4)$$

where σ' is standard derivation of $x'_{s,p}$ and α is parameter ($1 \leq \alpha \leq 2$) which depends on the body part. Figs. 3(a) and (b) show the gait skeleton and the noise removed skeleton data.

Now we can calculate the body angles from the skeleton data by using linear regression analysis. The angles θ_p of body part p can be approximated by using the slope of the lines in linear regression equation as

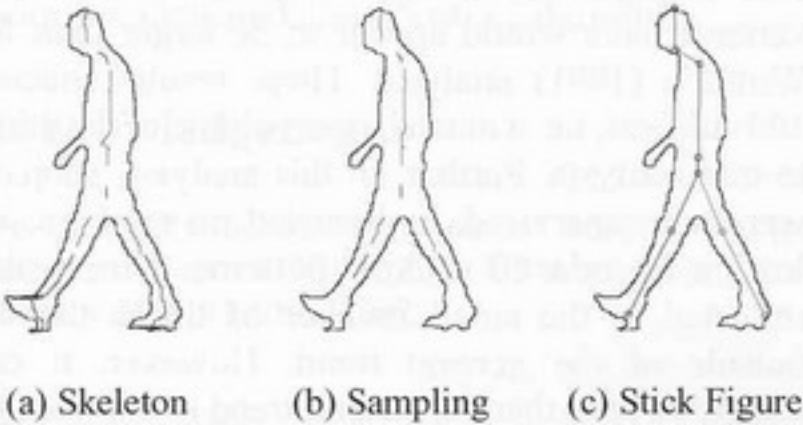


Figure 3: Extracting Body Points at Key-frame

$$\theta_p = \tan^{-1} \left(\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{\sum_{i=1}^n (y_i - \bar{y})(x_i - \bar{x})} \right) \quad (5)$$

where n is the number of the skeleton points. Each body points (positions) can be calculated by

$$x_p, y_p = [x_i + L_p \cos(\phi + \theta_p) \quad y_i + L_p \sin(\phi - \theta_p)] \quad (6)$$

where ϕ is the phase shift, x_i and y_i are the coordinates of a previously established position, and L_p is the length of body segments guided by anatomical knowledge (Dempster 1967). As shown in Fig. 3(c), a 2D stick figure with the nine body points is derived from the skeleton data of each body segment by linear regression.

3.2 Tracking Moving Points

In section 2.2, the important gait phases are detected by gait cycle analysis. The body points are clearly extracted around three double support phases (initial contact, terminal stance, and terminal swing), but the points around single support phases are not as good as the double supports. Therefore, a motion tracking method between double supports is used to extract body points at low limbs. To track knees and ankles, the left-most skeleton points around knee region and the right-most skeleton points around ankle region are considered. Namely, the skeleton points by a size of each body segment are sorted as

$$x_{knee} = \{x | x_i < x_{i+1}\} \text{ and } x_{ankle} = \{x | x_i > x_{i+1}\}. \quad (7)$$

The proper size of body segments is also guided by anatomical knowledge. The knee and ankle points for tracking are given by mean value of sorted data as

$$x_s = \{x_i | i = m, \dots, n\} \text{ and } \bar{x}_s = \text{mean}(x_s) \quad (8)$$

where m and n are sequence index of sorted data, here set to 3 and 5 respectively. The forward displacement Δx_t at frame t is calculated as

$$\Delta x_t = \bar{x}_t - \bar{x}_{t-1}. \quad (9)$$

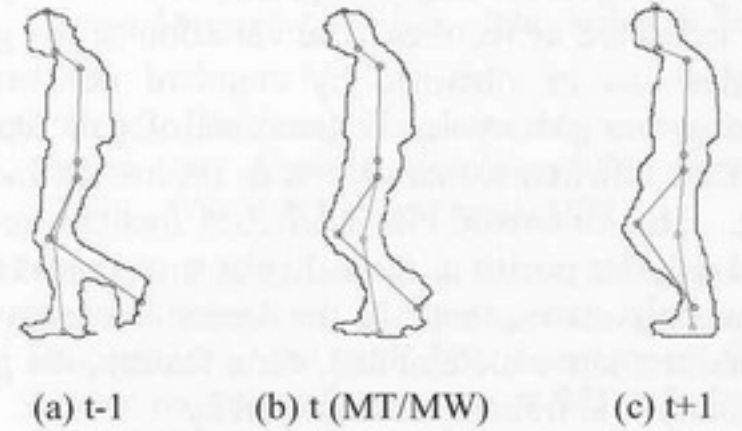


Figure 4: Body Points at Crossover of the Legs

Finally, the body points at frame t are computed by Eqns. (5) and (6) using the new angle values and segment sizes of thigh and shin.

In addition, functional or physical constraints exist in human gait motion. During the gait cycle, the other foot is in contact with the floor (and does not move forwards). Also, the knee angle should be equal to or lower than hip angles. The crossover of the two legs is performed on two single support points during one gait cycle. Fig. 4 shows an example of the body point extraction by using the gait constraints. In the figure, the crossover of knee is detected during the three frames of a single support, and ankle crossover is started after a single support. These constraints of human walking are used to improve the robustness of the method.

3.3 Feature Extraction

There are nine coordinates (body points) extracted by linear regression of skeleton data and motion tracking with topological analysis. The 2D stick figure obtained by connecting the nine coordinates. Fig. 5(a) shows the extracted stick figures during one gait cycle. We can re-calculate the six angles which are associated with joint positions: θ_{neck} , θ_{back} , θ_{thigh1} , θ_{thigh2} , θ_{knee1} , and θ_{knee2} . By definition (Winter 1991), the joint angles are relative, thus the relative angles in each joint were computed from the measured values of the angles. The definition of relative joint angles is shown in Fig. 5(b).

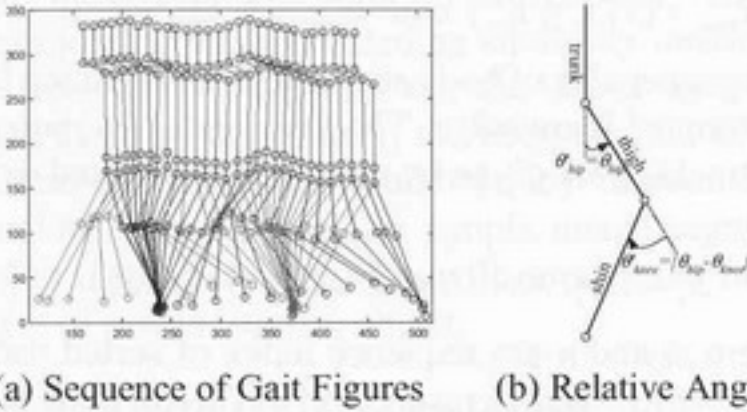


Figure 5: Gait Figures and Joint Angles

On the other hand, the mean and variation of the gait angles for single sequence of each subject are extracted as features. The variation of the gait angles can be obtained by standard derivation during one gait cycle. Trajectories of gait figure contain kinematic characteristics on human moving. The kinematic characteristics include linear and angular position, their displacements and the time derivatives, notably the linear and angular velocities and accelerations. As a feature, the gait velocity v_t at frame t is calculated by

$$v_t = (x_t - x_{t-1}) / \Delta t. \quad (10)$$

Also, the angular velocities ω_t at frame t , given an inter-frame time Δt (1/25 sec), are

$$\omega_t = (\theta_t - \theta_{t-1}) / \Delta t. \quad (11)$$

The trajectories are basically vector-valued function at each frame t of an image sequence, and they are periodic as in gait motion.

4 Results

In the experiments, three indoor image sequences of each of the 100 subjects are used. A set of gait data extracted from image sequences provides potentially valuable time-dependent pattern as a gait time series. Figs. 6(a) and (b) show the mean and standard derivation, by biomechanical data from anatomical markers in limb segment (Winter 1991), for joint angles of the hip and knee motion during one gait cycle, and Figs. 6(c) and (d) show the results of measuring relative joint angles during a gait cycle. In the figures, the lines are the curves that result from using 4th-order trigonometric-polynomial interpolants.

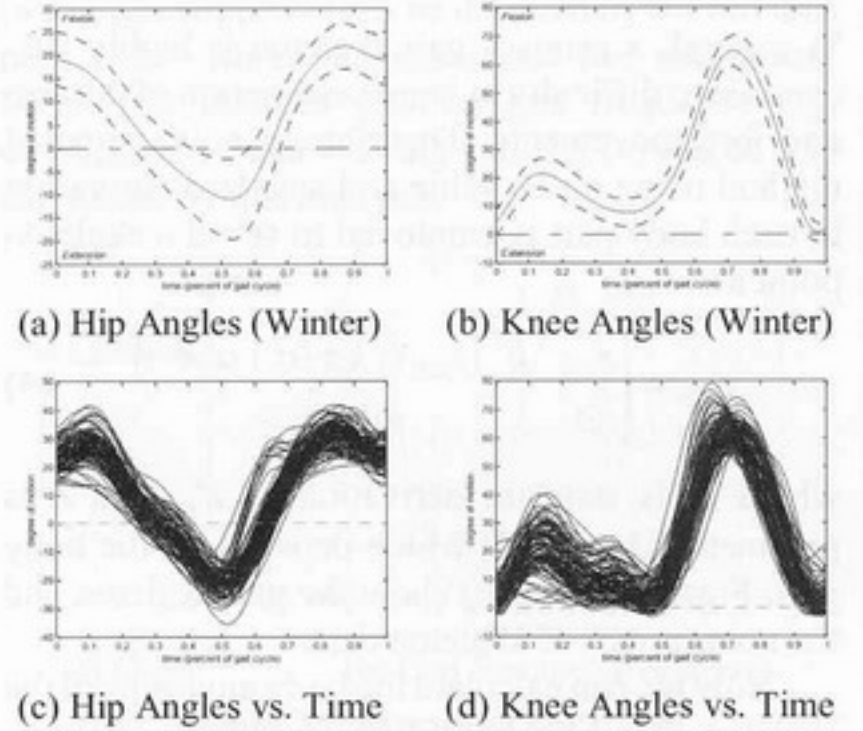


Figure 6: Gait Motion during One Gait Cycle

The results show that the new approach works successfully, comparing well with biomechanical data acquired by a marker-based system. The variance here would appear to be larger than for Winter's (1991) analysis. These results concern 100 subjects, i.e. a much larger volume of data than in that analysis. Further, in this analysis, subjects were not supervised, and carried no markers, allowing for relaxed walking patterns. This is also reflected in the small number of traces that lie outside of the general trend. However, it can clearly be seen that the general trend is followed by most of the traces, suggesting that earlier analyses

should be revised by this new approach.

To identify humans by their gait, an enhanced back-propagation neural network algorithm is employed as a classifier. We have mainly used the 20 features extracted based on kinematic analysis during one gait cycle. In training phase, the classification rate was 82% for three sequences (93% for two sequences) of 100 subjects, but we could not reach meaningful performance for recognition. However, the recognition rate for a small number (10) of subjects was 100%. Naturally we seek to extend the technique in terms of biometric application capability as well as analysing a large number of subjects in future. Notwithstanding this, the marker-less approach can clearly handle a small number of subjects successfully. By this, these measurements show that people are unique by their walking pattern, according with earlier biomechanical suggestions, and buttressing other similar results.

5 Conclusions

We have described an automated marker-less gait analysis system using computer vision techniques. The gait cycle was detected by symmetry property of the horizontal centre of mass. The gait figure was extracted from body contour by determining the body points using the linear regression of body skeleton and motion tracking with topological analysis. The joint angles extracted from a sequence of the gait figures were interpolated by trigonometric-polynomial functions. Also, trajectory-based kinematic features were extracted, and the enhanced back-propagation algorithm was employed for classifying the gait. As such, anatomical knowledge has guided development of a marker-less gait analysis system that not only accords with biomechanical analysis in the results it can produce, but also confirms uniqueness - as earlier suggested in biomechanical studies. In future, we require a more features of gait motion for describing and recognising the gait activities, and a more sophisticated classification algorithm.

Acknowledgements

We gratefully acknowledge partial support by the European Research Office of the US Army under Contract No. N68171-01-C-9002.

References

- D. Cunado, M. S. Nixon, and J. N. Carter. Using Gait as a Biometric, via Phase-Weighted Magnitude Spectra. in *Proceedings of the 1st International Conference on Audio-and Video-based Biometric Person Authentication*, pp.95-102, Crans-Montana, Switzerland, March 1997.
- W. T. Dempster and G. R. L. Gaughran. Properties of Body Segments Based on Size and Weight. *American Journal of Anatomy*, **120**, pp.33-54, 1967.
- D. M. Gavrilu. The Visual Analysis of Human Movement: A Survey. *Computer Vision and Image Understanding*, **73**(1), pp.82-98, January 1999.
- V. T. Inman, H. J. Ralston, and F. Todd. *Human Walking*, Williams & Wilkins, 1981.
- R. F. M. Kleissen, J. H. Buurke, J. Harlaar, and G. Zilvold. Electromyography in the Biomechanical Analysis of Human Movement and its Clinical Application. *Gait and Posture*, **8**(2), pp.143-158, 1998.
- M. P. Murray, A. B. Drought, and R. C. Kory. Walking Patterns of Normal Men. *Journal of Bone and Joint Surgery*, **46A**(2), pp.335-360, 1964.
- M. S. Nixon, J. N. Carter, D. Cunado, P. S. Huang, and S. V. Stevenage. Automatic Gait Recognition. in *BIOMETRICS - Personal Identification in Networked Society*, A. K. Jain, et al. Eds., pp.231-249, Kluwer Academic, 1999.
- J. Perry. *Gait Analysis: Normal and Pathological Function*, Slack, 1992.
- P. J. Phillips. Human Identification Technical Challenges. in *Proceedings of the IEEE 2002 International Conference on Image Processing*, pp.49-52, Rochester, USA, September 2002.
- J. D. Shutler, M. G. Grant, M. S. Nixon, and J. N. Cater. On a Large Sequence-based Human Gait Database. in *Proceedings of Recent Advances in Soft Computing*, pp.66-71, Nottingham, UK, December 2002.
- G. Taga. A Model of the Neuro-Musculo-Skeletal System for Human Locomotion. *Biological Cybernetics*, **73**(2), pp.97-111, 1995.
- D. A. Winter. *The Biomechanics and Motor Control of Human Movement*, 2nd Eds., John Wiley & Sons, 1990.
- D. A. Winter. *The Biomechanics and Motor Control of Human Gait: Normal, Elderly and Pathological*, 2nd Eds., Waterloo Biomechanics, 1991.
- J. H. Yoo, M. S. Nixon and C. J. Harris. Model-Driven Statistical Analysis of Human Gait Motion. in *Proceedings of the IEEE 2002 International Conference on Image Processing*, pp.285-288, Rochester, USA, September 2002.

High-cost cortical image processing on low-cost clusters

Hans du Buf *

*Vision Laboratory, University of Algarve
Campus de Gambelas-FCT, Faro, Portugal
dubuf@ualg.pt

Abstract

The anisotropic, local image representation of simple and complex cells in the primary visual cortex provides input for higher-level processes. Although many functions are not yet well understood, we can assume that there are line and edge detectors, end-stopped operators, bar and grating cells, even specific texture cells. Using phase information of complex Gabor filters that model simple cells, it is possible to extract motion and depth information. This set of feature-extraction engines can be seen as a frontend that prepares input for yet higher-level, cognitive processes. Gabor filters tuned to many different frequencies and orientations, and the subsequent feature extraction engines, require a huge computing power. Cheap microprocessors for PCs such as AMD's XP, which can deliver more than 500 MFLOPS, cannot be compared with high-end, 64-bit RISC processors, but it is easy to build a cluster of PCs. It is less easy to parallelize code. The main aim of this paper is to show that every laboratory, however small, can afford a small cluster and can apply parallel processing, provided that easy parallelization tools and libraries are used.

1 Introduction

Neurophysiological studies reveal important aspects of the processing going on in our visual cortex. By analyzing the behavior of specific cells in different areas to various input patterns, it is possible to deduct what their function is. In many cases this function is highly nonlinear, i.e. one cell may respond only to one pattern and not to other patterns. Such a selectivity reveals information about the symbolic image representation in the different layers, although the exact processing, i.e. the connections between cells in each layer and those between layers, may still be unknown. In other words, we can try to formulate cell models and, as experimental architects, try to create a biologically-motivated processing hierarchy that can be used to model higher-level, cognitive effects, or that could be used in robots.

Modeling the visual cortex or parts thereof requires huge computing power. Most cell models are based on the responses of anisotropic Gabor filters that mimic so-called simple cells. Often we apply a pre-frontend that consists of many filters, easily with 12 orientations and 8 frequencies. Filtering is done in the frequency domain, which requires one forward FFT and 96 inverse FFTs. One complex

FFT of size 256×256 takes about 30 ms on a PC with an AMD XP processor running at 1.6 GHz. Hence, 100 FFTs take about 3 seconds. But this is only the pre-frontend applied to one, small image; the real processing still needs to be done, images can be larger, and stereo/depth processing implies an additional factor of at least two. An intelligent robot that needs to act in realtime will have problems, unless we apply parallel processing.

Parallel processing implies a parallelization of tasks, for example DO loops (here I will not go into distributed processing of different tasks). Fortunate researchers at Universities with a computer center may have access to expensive SMP or shared-memory systems, even clusters of SMP systems (NUMA). Fortran and C compilers for such systems allow parallelization by means of directives, called pragmas in C. A booming alternative is to use clusters of normal PCs, called Beowulf clusters, that can combine hundreds or even thousands of processors (see www.beowulf.org and www.top500.org). Parallelization must be done using communication libraries, for example MPI (Message Passing Interface). Using MPI is much more complex than using a few compiler directives, and communications on clusters are much less efficient than on SMP sys-

tems. However, (1) small clusters can be used by researchers with a small budget, (2) communications can be reduced by using clever strategies, (3) parallelization or programming overheads can be eliminated by developing higher-level libraries and tools, and (4) supercomputer performance can be achieved for realtime cortical processing. Below I will first illustrate some examples of cortical processing, with an emphasis on parallel processing. Then I will turn to the parallelization tools that we develop: (a) directives, (b) libraries and (c) faster interconnects.

2 Cortical processing

Petkov and Kruizinga (1997) published models for bar and grating cells. These cells are very selective: a grating cell does not respond to isolated bars, whereas a bar cell does not respond to the individual bars in a periodic grating. As we will see later, grating cells can play a role in texture segregation. Our improved bar and grating cell models (Santos and du Buf, 2002a) are based on a retinal preprocessing by means of separate ON and OFF Gaussian filters, which can be seen as DOG filters, but together with a shunting inhibition to normalize the signals. The output of this stage are two signals, ON and OFF, which are both positive or zero: the ON reacts to bright bars, the OFF to dark bars. Orientation selectivity is created by Gabor filtering of the two signals, but we keep separate ON and OFF channels that are positive or zero. We now assume grouping units that consist of five equidistant cells that sample the ON and OFF signals in the direction of the Gabor filters (frequency selectivity). A grating cell responds when it can group ON,OFF,ON,OFF,ON or OFF,ON,OFF,ON,OFF responses, whereas bar cells respond when seeing either OFF,ON,ON,ON,OFF or ON,OFF,OFF,OFF,ON responses. Grating cell outputs can be OR-ed together in order to detect linear textures with arbitrary frequency and orientation. Figure 1 shows a *Navicula oblonga* diatom image with grating and bar cell outputs. The striation of the diatom valve normally varies from the center to the two end poles. Hence, the striation can be detected, frequency and orientation features can be computed (Gabor filters!), and these can be used to identify diatoms. Grating and bar cell processing can be easily parallelized, for example by distributing the frequencies (grating periodicities and bar widths) over available processors, after which each processor will first

apply the Gaussian filters (one width) and then all Gabor filters (the set of orientations). In a last step, one processor can collect the feature maps in order to combine them into one or two maps.

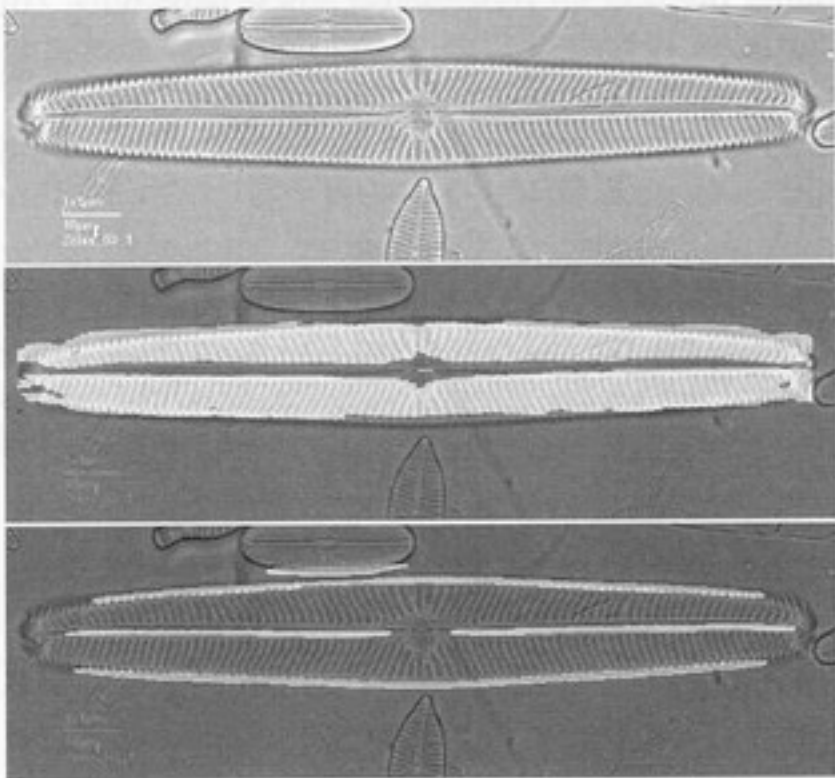


Figure 1: *Navicula oblonga* diatom (top), striated regions detected by grating cells superimposed on a darkened version of the input image (middle), and bar cell outputs (bottom).

The striation in Fig. 1 is a linear texture. In general, textures are more complex: they can have multiple orientation components, even with different periodicities. Using 12 orientations, we can group the outputs of grating cells which are e.g. orthogonal. This can be done for 6 different orientation pairs. Hence, after some normalization, “texture symmetry cells” can detect rectangular textures irrespective of their rotation. By combining other orientations, textures with other symmetry orders can be detected as well: hexagonal and octogonal structures (Santos and du Buf, 2002b). Figure 2 shows that grating cells can be used in texture segregation. Shown are Brodatz textures of different symmetry orders and the outputs of “rectangular” cells tuned to two frequencies. This is *not* a segmentation, merely a bottom-up or data-driven feature extraction, but the encoded regions could be used in some segmentation stage. The result of Fig. 2 was obtained by considering only components having the same periodicity, hence a parallelization as described above does not involve additional communications between processors. However, additional groupings between different periodicities, for example in the case of a fine horizontal structure in combination with a coarser

vertical one, requires an efficient solution. If different frequencies/periodicities are processed by different processors, more communications cannot be avoided, for example to let one processor collect all grating maps of all frequencies and orientations. An alternative is to re-distribute the symmetry-grouping results, such that one processor does all rectangular structures, a second does hexagonal ones, and a third octogonal structures. But this poses another problem, because in the case of octogonal structures both “octogonal cells” and “rectangular cells” will respond, but the octogonal will have priority because of the higher texture complexity. Only by implementing some methods and by comparing CPU times the most efficient solution can be determined.

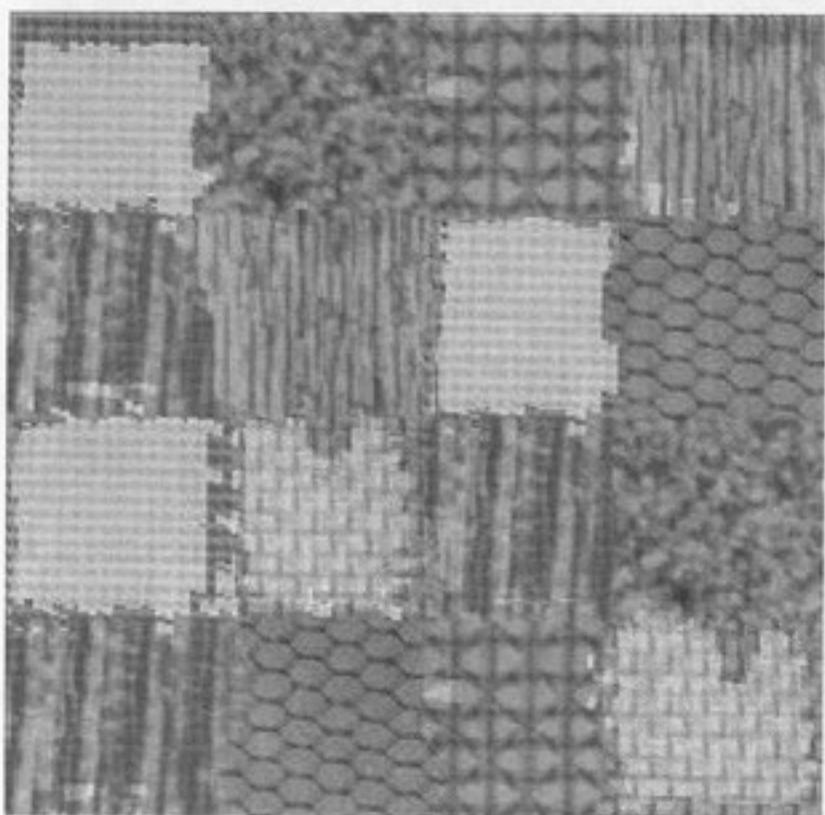


Figure 2: Grouping outputs of orthogonal grating cells allows to detect rectangular textures. Shown are the results for two periodicities (lighter areas).

More basic is a multiscale line/edge representation. Lines and edges can assume various profiles, i.e. a bar cell may not detect a bar with a Gaussian profile. Normally, in multiscale analysis only a few Gabor filter scales are considered, which makes it difficult to link detected lines and edges over scales. Computer performance is now such that an almost continuous filter scaling can be applied. In brightness models we can apply hundreds of scales (du Buf and Fischer, 1995). Such models were tested in 1D, but need to be extended to 2D, which makes the linking over scales more difficult. The main problem is that Gabor filter responses are influenced by interference effects, i.e. two close edges at a cross-

ing may cause the local phase to indicate e.g. a line (du Buf, 1994). Figure 3 shows a preliminary result, single scale with 8 orientations, obtained by combining extrema, zero crossings and the modulus of local Gabor filter responses. The “simple” test image, with a strong vertical grating, a weaker annulus and two faint lines, wrecks most detection schemes, like those of Canny and Marr-Hildreth (van Deemter and du Buf, 2000), which don’t like crossings. Here, our result looks perfect, but our method still lacks an additional processing layer that can complete gaps, for example (dendritic) summation fields that improve continuity, irrespective of line/edge polarity and for various curvatures, including inflection points. We are also thinking about including curved Gabor filters (“banana wavelets,” see Krüger, 1998), which cannot be easily implemented in the frequency domain, but combinations of neighboring, straight Gabor filter responses can provide approximations (Krüger, 1998; see also van Deemter and du Buf, 2000). A particular challenge, apart from parallelizing all algorithms, is to unify all feature maps into a representation that can be used for object recognition: lines, edges, bars, gratings, textures, even the results of end-stopped operators (Heitger et al., 1992).

A related aspect concerns the modeling of psychophysical data, measured by means of visual experiments and human observers. We are working on nonlinear brightness and detection models. In the case of detection, the contrast is so low that the observer can only detect a change of the background on which the pattern is superimposed. Figure 4 shows a contrast sensitivity function (CSF) for sinusoids, i.e. the contrast necessary for detection as a function of the spatial frequency, both in log units. The solid symbols are measured data points. The model is applied to images of size 256×256 , with 8 frequencies and 8 orientations. Gabor filter responses are nonlinearly summed over orientations and local neighborhoods, and then over frequencies. Each frequency channel has a gain factor to be determined by a numerical optimization technique. We use a constrained version of Powell’s method (Numerical Recipes) because the channel gains must be positive. Figure 4 also shows the contributions of different frequency channels after model calibration. Model calibration takes only a few seconds. After calibration for certain summation exponents, the model should be able to predict correctly CSFs of other patterns.

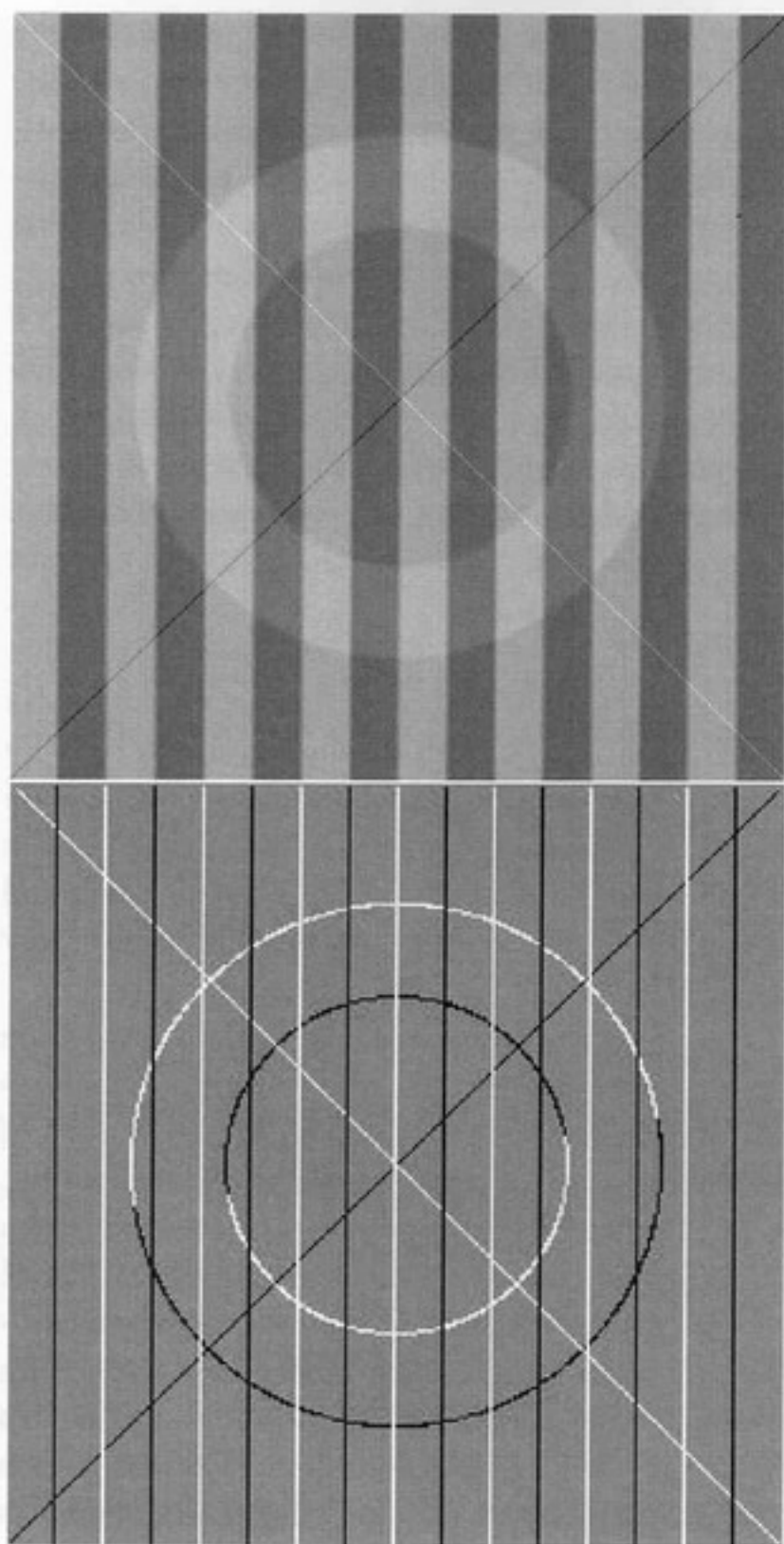


Figure 3: Test image (top) with the coded output of single-scale line/edge detection. Note that edge polarity depends on the filter orientation.

Figure 5 shows predictions in the case of incremental and decremental disks, i.e. luminance increments and decrements as a function of disk area, again in log coordinates (see also du Buf, 1992).

Apart from determining the right summation exponents and other free model parameters, one difficulty is that predictions of contrasts at the *input* of the model must be determined such that at the *output* of the model the amplitude is either +1 or -1. This must be done by using a bisection method in which the model is applied until the detection criterion is fulfilled. This is computationally very demanding. The model was parallelized over the fre-

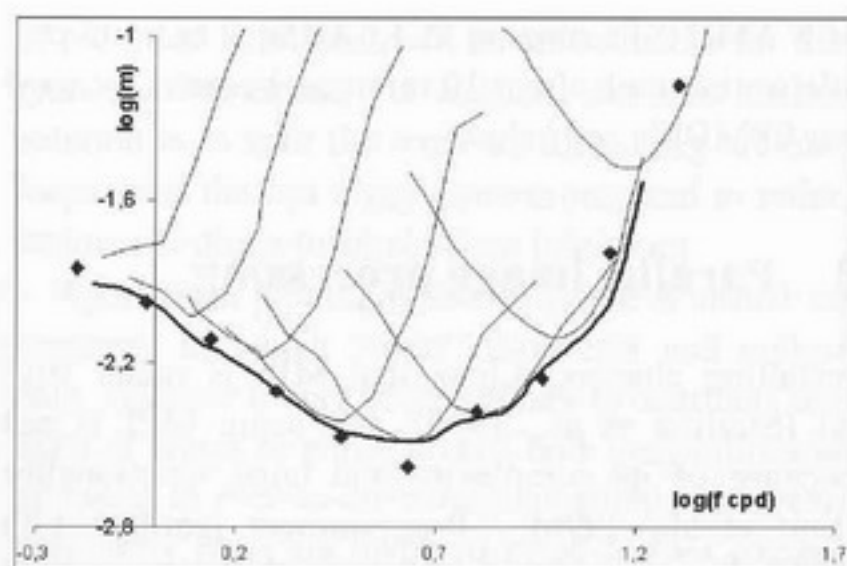


Figure 4: Detection data for sinusoids (solid symbols) with model calibration result, also showing the contributions of the frequency channels.

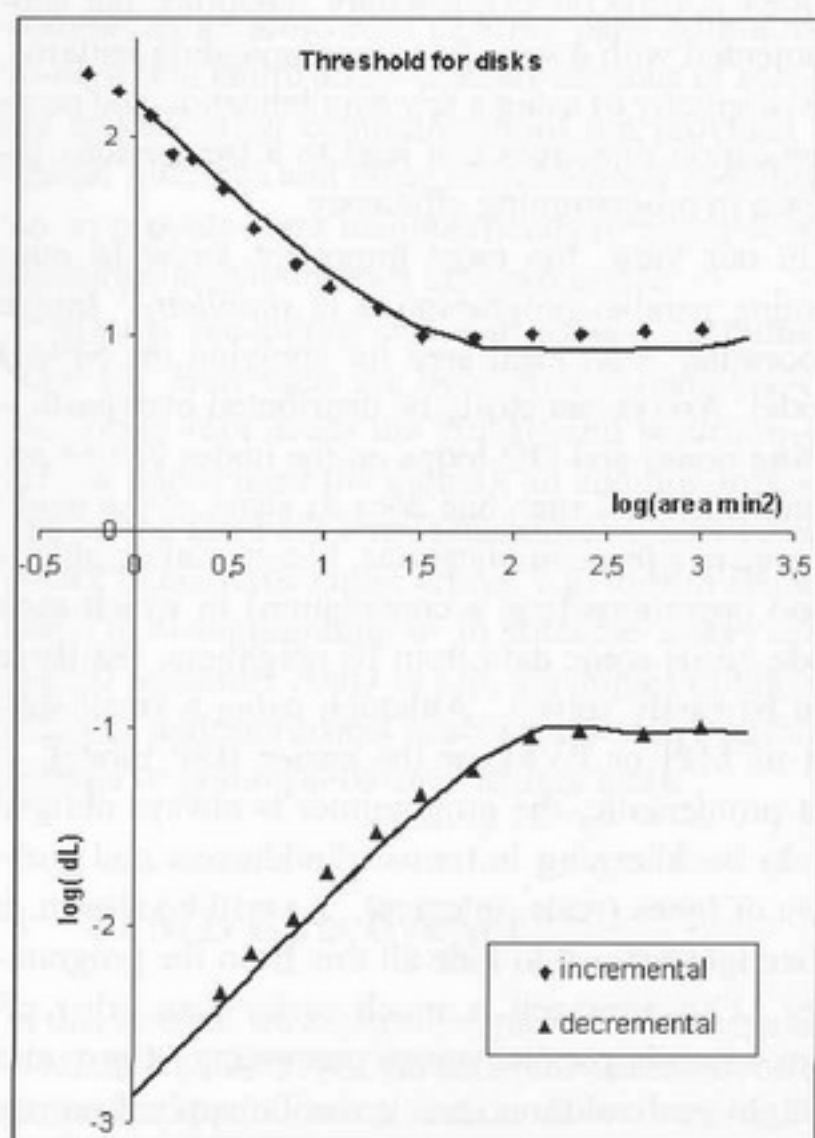


Figure 5: Model predictions for incremental (upper part) and decremental disks, along with measured data (solid symbols).

quencies, i.e. the final summation over frequency channels involves a simple sum reduction. The computation of the two curves in Fig. 5, i.e. two times 20 data points, which takes about 90 minutes on an older SGI server with a MIPS R10000 RISC processor at 180 MHz, takes about 2 minutes on a cluster

of 8 AMD XPs running at 1.6 GHz. The code parallelization took about 10 minutes, because we used our SPMDlib; see below.

3 Parallel image processing

Installing clusters, Linux and MPI is rather trivial (Sterling et al., 1999), but using MPI is not because of its complexity and huge functionality (Snir et al., 1996). Programmers familiar with SGI's directive C\$DOACROSS or the new standard OpenMP (C\$OMP, see www.openmp.org) on expensive SMP and NUMA architectures find MPI programming rather archaic. Especially in image processing, where many parallel loops can be split (which is C\$DOACROSS's only function), but supplemented with a very few communication patterns, the simplicity of using a few parallelization and communication directives can lead to a tremendous increase in programming efficiency.

In our view, the most important keyword concerning parallel programming is *simplicity*. Image processing is an ideal area for applying the SPMD model. Arrays can easily be distributed over participating nodes and DO loops on the nodes can be adjusted such that each one does its share of the work. There are a few complications, like spatial neighborhood operations (e.g. a convolution) in which each node needs some data from its neighbors, but these can be easily solved. Although using a small subset of MPI or PVM, or the easier BSP model, is not problematic, the programmer is always obliged to do bookkeeping in terms of addresses and numbers of bytes (reals, integers). As will be shown, it is straightforward to hide all this from the programmer. Our approach is much easier than other efforts towards parallel image processing, it provides a high-level solution, and it can be applied on any system with MPI support. In addition, we aim at developing a tool which is (a) low-cost in terms of price and usage, (b) very easy to use, and (c) which preserves code portability. The latter points imply that the same program can be executed on one CPU (no communications at all), on SMP systems (shared memory with very efficient "communications" using C\$DOACROSS or C\$OMP) or on clusters (distributed memory with very expensive communications using our C\$SPMD), depending on the context.

We will mention a few other approaches to parallelizing image processing. Lückenhaus and Eck-

stein (1998) apply an agent-based approach with the thread concept on Sun SMP systems. Using task graphs, three processing models can be realized: data parallel (e.g. pixels), task parallel (independent tasks like serial code blocks) and pipelining (e.g. applying the same processing steps to an image sequence). Squyres et al. (1998) describe the PIPT or Parallel Image Processing Toolkit. Instead of explicitly parallelizing each routine, a small set of basic operations is provided such that users can easily define a window operator, for example, which will be "embedded" in a parallelized routine. Gennart and Hersch (1999) describe a Computer-Aided Parallelization (CAP) tool. CAP is a precompiler that generates C++ source code. It enables programmers to specify at a high level of abstraction threads by using the split-merge concept. Seinstra et al. (2001) use the C++ function template mechanism to create generic algorithms on the basis of many basic and pre-defined functions like add, sub, mult and pow that the user should include in function calls.

One common aspect of these efforts is to hide parallelization details from the programmer: *automatic* parallelization still is one of the Holy Grails of computer science. However, *efficient* parallelization, especially on distributed-memory systems, requires that the programmer must have complete control over all communications in order to minimize communication overheads: each *millisecond* spent with communications implies a loss of about 6 MFLOPs on a cluster of 8 AMD Athlon XPs (1.6 GHz). This is the reason that we extend parallelization directives with a few communication directives: we aim at both programming and execution efficiency.

Below we will first discuss our SPMD initiative, and then explain the directive set, the parser, and the first library implementation. We will conclude this paper by describing new library implementations and some concepts for Fortran 90/95 and C, including communication optimizations.

4 The SPMD initiative

The main goal is to develop a parser or preprocessor that translates parallelization and communication directives (SPMDdir) into library calls (SPMDlib). The limited functionality and simplicity of SPMDlib leads to a very small set of directives: the one-to-one mapping and modularity of both library and parser has two important advantages: (1) in contrast to the

other efforts mentioned in Section 3, the development will not cost many person-years, and (2) future extensions can be easily integrated. Additional advantages are: (3) although the programmer should know about communication costs and parallelization granularity, he/she must not necessarily know MPI; (4) once that we have a working parser version, we can optimize the library and even the parser itself. Like BERT (www.plogic.com), the parser could be made intelligent in order to instruct the programmer about e.g. the number of nodes to use.

The simplicity of our approach implies a restricted functionality. However, most if not all image processing operations can be parallelized. In the ideal case one node (called root) will read an image file into an array. It will then distribute equal array parts to other nodes, and all nodes, including root, will process their array parts. After all processing has been completed, the other nodes will send their parts to root, which will assemble them and then write an output file. More sophisticated processing, such as the examples with 2D FFTs given in Section 2, can be organized such that parallel processing can be optimized.

The underlying idea of SPMDlib is to parallelize loops over arrays, i.e. 2D image arrays, but 1D and 3D arrays, or even n D arrays, can be processed in the same way. SPMD implies that all participating nodes are executing the same program, but each node has an identifier and knows the total number of nodes ($0 \leq \text{myid} \leq \text{nprocs} - 1$). Also knowing the loop size, each node can easily compute its part, i.e. `mystart` and `myend`. The only restriction of SPMDlib is that (in Fortran) parallel loops over arrays must start with 1 and have a stride of 1 (in the case of nested loops over multidimensional arrays only the *outer* loop over the *last* dimension must be parallelized). The implicit array partitioning allows to hide all bookkeeping from the programmer, and in most applications all nodes will equally share the work load. When the outer loop/array size is not a multiple of `nprocs`, for example a large prime number, there are several solutions: (a) let the last node do less work, (b) transpose the array, maybe the other dimension is more suitable, (c) extend the array with a few blank lines, or (d) use for this loop fewer nodes, i.e. put to “sleep” temporarily some nodes. Below we include a solution for dynamic loop scheduling (load balancing), but this will involve more communications and, as for most

of the other solutions, will be less suitable for fine-grain parallelization. The simplest and most efficient solution is to split the work equally using the outer loops over the last array dimensions, and to reduce communications to an absolute minimum.

Apart from pipelining, we only use a virtual star topology in which “root” distributes and collects data. Because it may be necessary to distribute only parts of arrays or entire arrays, both possibilities are included in *root-to-all* communications. However, only array parts are involved in *all-to-root* communications (an *all-to-root* of entire arrays is called an array reduction; see below). In addition, when all nodes have processed their array parts, it may be necessary that all nodes need the entire array: *all-to-all* communications can be more efficient¹ than combining an *all-to-root* of array parts with a *root-to-all* of the entire array. Special versions of *root-to-all* and *all-to-all* communications are provided for spatial filterings and other neighborhood operations, i.e. to provide extra neighborhoods to nodes or only exchange neighborhoods between nodes.

Normal reductions concern scalars: maximum, sum, etc. Here there are two options: *reduction-on-root* (only root needs the global) and *reduction-on-all* (all nodes need the global). In addition to scalar reductions there are array reductions: it may be necessary to combine entire arrays, e.g. to sum elementwise in histogramming or to combine arrays using logical operators AND or OR. Examples of OR-ing together multiple arrays were discussed in Section 2: outputs of grating cells and “texture cells.”

5 SPMD directive set

In this Section we explain the main directive clauses in relation to the SPMDlib calls; the latter will be explained in the following section. Directives for Fortran start in the first column and have the form `C$SPMD clause[, clause] ...` in which the clauses are executed from left to right (depending on the context). Clauses can be continued on multiple lines, but each line starts with `C$SPMD` and clauses are separated by comma-blank. `C$SPMD` directives should not be “mixed” with `C$DOACROSS` or `C$OMP` directives; although these are allowed, they will be ignored. The clauses and their meaning are (advanced features are marked by an *; these will not be discussed in this paper):

¹Efficiency also depends on the network usage: multicast routing is ideal for root-to-all communications!

INIT	initialize SPMDlib
END	finalize SPMDlib
BARRIER	node synchronization
RONLY	root only does
TIME, TTIME	time and total time
PARBLOCK	parallel code blocks (*)
DOPAR	parallel DO loop
DODYN[...]	dynamic DO, chunk size
PIPE[...]	node pipelining (*)
R2A[list]	root-to-all
A2R[list]	all-to-root
A2A[list]	all-to-all
RoR[list]	reduction-on-root
RoA[list]	reduction-on-all

in which *list* is a list of scalars and array names (only names!); the latter will be indicated by <an> below. The following abbreviations and combinations are used in some clauses:

TCPUS= <i>n</i>	total number nodes (*)
NCPUS= <i>n</i>	nodes to be used (*)
E:<an>	entire array
P:<an>	parts of array
P(X= <i>n</i>):<an>	exchange <i>n</i> elements/ lines/planes 1/2/3D arrays
P(XC= <i>n</i>):<an>	cyclic exchange
SUM:variable	type of scalar or array reduction: MAX, MIN, AND...
IN=variable	input scalar or array in reduction
OUT=variable	output scalar or array in reduction

INIT and END should be used in the main program or in one subroutine. All other clauses, except for DODYN and PIPE, can be used in other subroutines, provided that all nodes call these.

5.1 Brief explanation

SPMDlib consists of a few subroutines for initialization (`spmd_init`), synchronization (`spmd_barrier`) and for splitting loops and arrays (`spmd_split`). Most routines are for communications. There also is an include file (`spmd.h`) with predefined and prefixed variables and a common block (in Fortran) to pass parameters to communication routines. Below follows a brief description of most directives in relation to the library.

INIT invokes `spmd_init(ierr)` that provides predefined scalars `spmd_myid`, `spmd_nprocs`, `spmd_mystart`, `spmd_mypart` and `spmd_myend` (all integers), as well as prefixed scalars for timing purposes (normal reals). Initializes `spmd_nprocs` and `spmd_myid` (0 to

`spmd_nprocs-1`). Root will be number 0. The programmer is free to use the predefined scalars mentioned above, although code portability will be lost.

END invokes `spmd_end(ierr)`. Must be included before stop/end.

BARRIER invokes `spmd_barrier(ierr)` and must be used where an explicit node synchronization is required, for example when root only (see **RONLY** below) needs a significant CPU time. Note that all communications and reductions cause an implicit node synchronization because of the synchronous `mpi_send` and `mpi_recv` used in the library. If **BARRIER** is used together with a closing **RONLY**, the program will not be deadlocked because it will be executed after the **RONLY** code (the parser will detect a **BARRIER** inside two **RONLY**s and will print a warning).

RONLY is a switch to define regions in which Root-ONLY is active, for example to prepare or output data (read/write a file, print); the other nodes will skip this region and continue the execution of the subsequent code until they encounter a barrier or a communication clause. Must be inserted before and after the code lines (or only once for timing, see **TIME!** below), and invoke only an if-endif:

```
C$SPMD RONLY
      call readfile(...,array,dim)
      call normalize(...,array,dim)
C$SPMD RONLY, R2A[P:array]
```

becomes

```
if (spmd_myid.eq.0) then
  call readfile(...,array,dim)
  call normalize(...,array,dim)
end if
call spmd_split(dim)
call spmd_r1DPtoall(array,dim)
```

This example also shows a *root-to-all* clause **R2A** and the associated routine `spmd_r1DPtoall` in the case of a real, 1D array declared with dimension *dim*, the *P* meaning that root must distribute array parts. The parser will also insert the `spmd_split` call if (a) this was not done before or (b) the dimension was different. All communication and reduction routines synchronize all nodes, hence no barrier is necessary. If the CPU time is large and the region not immediately followed by e.g. an **R2A** directive, the second **RONLY** can be combined with **BARRIER**.

TIME and **TTIME** are for timing purposes. **TTIME** is for larger code blocks that contain **TIME** directives. Both must be combined with **RONLY**. The timer starts or stops on the **RONLY** line that contains (T)TIME, i.e. the work to be done by root only can be excluded or included. If an **RONLY** does not involve other work for root, then **RONLY**,

TIME! or RONLY, TTIME! must be used. Both must be used pairwise, i.e. they imply start-stop, and both return predefined normal reals `spmd_time` and `spmd_ttime` that the programmer can use. A future version may include an additional clause for telling the parser what code to include in order to print a message. For example:

```
C$SPMD RONLY, TIME!
      code block to be timed
C$SPMD RONLY, TIME!,
C$SPMD PRINT[Code block took:,spmd_time]
```

would be translated into

```
if (spmd_myid.eq.0) call spmd_rtime
code block to be timed
if (spmd_myid.eq.0) then
call spmd_rtime
print*, 'Code block took:', spmd_time
end if
```

which could include some arithmetic to convert seconds to e.g. milliseconds.

DOPAR must be inserted directly before a parallel DO loop, not counting empty or comment lines (e.g. C\$OMP). Must always be the outer loop over 2D/3D arrays because it invokes `spmd_split(dim)`, in which `dim` is the outer loop/array dimension, to compute `spmd_mystart`, `spmd_mypart` and `spmd_myend` on all nodes. The parser will change the loop `do i=1,dim` to `do i=spmd_mystart,spmd_myend` and the three parameters (start,part,end) will be passed (via common block) to communication routines involving array parts. Note that root (node number 0) equally contributes and will also process array parts, i.e. the first parts. Since **DOPAR** implies a static and equal loop scheduling, the user can put a **BARRIER** after the loop if the loads are not balanced (not necessary if the loop is immediately followed by a communication or reduction). The routine `spmd_split(dim)` will check that `dim` is a multiple of `spmd_nprocs`, and will print a warning message if this is not the case (i.e. at runtime; the parser can also be instructed—using **TCPUS**—to check this).

DODYN[chunk,array] is for dynamic scheduling of an outer DO loop, in which each node will start with a “chunk” and, after sending its chunk of data to root, will receive a request for doing another chunk, until the loop has been completed. Because of the irregular communication pattern, root will only distribute chunks (start numbers of the loop)

and collect results (corresponding array chunks); all non-root nodes must have all necessary data. The chunk size depends on the load imbalance, and should be smaller than the (outer) loop dimension divided by the number of non-root nodes. A smaller chunk size implies more communication calls, but each with less data. The parser will put all necessary code lines and library calls before and after the loop. A typical application is Direct Volume Rendering in which the root node has an OpenGL window for displaying a rotating object (in this case we use the frontend of the cluster as root, *not* using `mpirun -nolocal`). Note: **DODYN** has not yet been implemented in the library, but on the basis of our experience we know that its implementation is rather straightforward using a few routines and predefined integers in the `spmd.h` include file.

5.2 Communications and reductions

All communications are done by **R2A** (root-to-all), **A2R** (all-to-root) and **A2A** (all-to-all), indicating scalars, entire arrays (E:) or parts of arrays (P:), but without any types and array dimensions. Example:

```
C$SPMD R2A(val1,val2;P:array1)
```

Already implemented are integer, real, double and complex types for 1D and 2D arrays. The following restrictions and extensions apply:

1. Array names in **R2A**, **A2R** and **A2A** must be preceded by E: (Entire) or P: (Parts), like P:arrayname. This is not necessary in **RoR** and **RoA** reductions.
2. By default arrays are split into equal parts, the number of which corresponds to the number of nodes specified on the command line (e.g. `mpirun -np 8`).
3. The variable list is a list of scalars and arrays separated by commas and semicolons like: `scal1,scal2;P:arr1;E:arr2,arr3`
4. Array communications: **R2A** can be P or E. **A2R** must be P. **A2A** with E implies that all nodes having only one part will receive all parts (the entire array).
5. Spatial neighborhood operations: in the case of **R2A** and **A2A**, P can be combined with (X=n), X meaning eXchange, or (XC=n), XC meaning eXchange Cyclic. Example: `R2A[P(XC=3):arr1;P:arr2]`, where instead of “3” we can use an integer scalar. **R2A** will distribute the normal array parts of `arr1` plus the neighboring elements/lines for a first iteration; **A2A** will only

exchange neighboring elements/lines necessary for multiple iterations of window operations.

Scalar reductions can be done using only two clauses: (1) **RoR** (Reduction-on-Root), where only root gets a global value; the other nodes will continue with their local values. (2) **RoA** (Reduction-on-All), where all nodes will continue with the global value. Reduction types: we will implement at least SUM, MAX, MIN, AND and OR. Allowed data types for SUM, MAX and MIN are integer, real and double; logical operations should use integers with values 0 and 1.

The same clauses and types are used for entire **array reductions**, which work elementwise. The only complication occurs when all nodes are using the same working arrays (or scalars) and the result should be stored in another array (or scalar). This can be done by using for example `RoR(SUM:IN=arr1,OUT=arr2)`, in which `arr1` and `arr2` have the same type and dimensions (could be scalars of the same type). In the case of e.g. a SUM reduction, both `RoR` and `RoA`, the user should initialize `arr2` to zero on root (using `RONLY`).

6 The dumb parser and existing SPMDlib

The parser, written in C, has the following main tasks: (1) to detect and check all directives in a source program, (2) to detect data types and array dimensions of scalars and arrays included in directives, (3) to keep track of parallel loop and array dimensions for splitting them, (4) to change parallel DO loops and (5) to insert the correct SPMDlib calls. Only in the case of the advanced clauses `DODYN` and `PIPE` it will significantly change the source code, but the parsed code will stay legible because of extra comment lines that indicate the actions taken. Furthermore, all SPMDlib routines and special scalars/arrays are prefixed (`spmd_`) in order to avoid conflicts. Here we describe the existing library, i.e. the routines already implemented in Fortran (C is under construction). Advanced features for C and Fortran 90/95, as well as the new library to be implemented, will be described in a following section.

We note that the library was first developed for being used by the programmer. Because of the user-friendliness, this implies that the routines must be named such that it is easy to select the right ones for different data types and actions, and that there

are many different routines. For example, `R2A` communications involve 4 data types, 1D and 2D arrays, entire arrays and array parts, plus special routines for extending array parts with additional elements/lines for neighborhood operations. This means that there are 24 `R2A` routines. However, since MPI is not concerned about data types, i.e. only the number of bytes for each data type is important, routines for integers call the routines for reals, and those for complex call the `real*8` ones. All this is not necessary in developing the new library (see next section), because the new routines are not meant to be used by the programmer—only the directives must be used. In addition, the existing library was built taking into account the most important features that we needed. A future version should be more versatile, such that most basic operations are covered, such as all possible reduction types.

The main program and all subroutines that use SPMDlib calls need special declarations and a common block in the `spmd.h` include file². Most important are the integers `spmd_myid`, `spmd_nprocs`, `spmd_mystart`, `spmd_mypart` and `spmd_myend`.

The routine `spmd_init(ierr)` initializes MPI as well as a few variables, notably `spmd_myid` and `spmd_nprocs`. The routine `spmd_end` quits MPI. The routine `spmd_split(dim)` is used for static loop splitting and array partitioning; it computes `spmd_mystart`, `spmd_mypart` and `spmd_myend` for changing loops and for use in the communication routines for array parts.

All communication routines have the following naming convention:

`spmd_[ty][nD][P/E][func](vars)`
in which:

`[ty]` = `i/r/d/c` mean integer, real, `real*8` (double) and `complex*8` data type,

`[nD]` = `1D/2D` mean arrays with one or two dimensions (3D to be done),

`[P/E]` mean only Parts of an array or an Entire array,

`[func]`: `toall`, `toroot`, `sumonroot`, `syncarray`, `filtX` and `toallX`, and

`(vars)`: the name of one array and its dimensions.

Notes: (1) `Psyncarray` implements `A2A` with the `E` option; (2) `PtoallX` implements `R2A` with `P(X=n)`, `XC` not yet being implemented³; (3) `PfiltX` imple-

²See <http://w3.ualg.pt/~dubuf/spmd.html>

³Is cyclic neighborhood processing important? Top image

ments A2A with P(X=n), see the same footnote; (4) Esumonroot is an example of an array reduction; all other types can be written in one hour or so.

The routines for scalar reductions do only the communications, i.e. a local maximum or sum needs to be done with a split loop after which globals are determined from the locals with one (or more) reduction call. Naming convention:

RR means Reduction-on-Root (only root gets global maximum (for example)),

RA means Reduction-on-All (global maximum goes to all). Already implemented are:

m means max and min, and

s means sum and sum-of-squared, for integer, real and real*8 data types. Examples:

```
spmd_iRAm(max,min)
```

```
spmd_dRRs(sum1,sum2)
```

These routines are not yet linked to directives, but were developed for image processing: (1) image normalization using the maximum and minimum values, and (2) for computing the mean and variance in one (split) image loop (well, RRs can do two independent sum reductions).

Note: sending one or two scalars takes the same time. Hence, if we want only a maximum (minimum), we put a dummy min (max) with an arbitrary value. The parser can also do this.

7 Status, other languages, etc.

After parallelizing a few programs with the existing routines we know that the SPMDdir/lib concept works, that there are no or few flaws in the design, and that a simple parser could be beneficial for many programmers. The parser has now highest priority and the standard Fortran version will be ready by summer 2003. By then we will also have selected an implementation solution for the new libraries in Fortran and C (the directives, called pragmas in C, will be the same as in Fortran).

The three main differences in the C version will be: (a) no common block can be used, hence all pre-defined spmd variables need to be global, (b) dynamic memory allocation must be included, with all its implications for splitting loops and communications, and (c) loops and their nesting differ from Fortran. It is likely that only one pointer structure will be supported, such as the one used by MPI.

For Fortran 90/95 we will also consider dynamic memory allocation. The question is whether a parser can always detect the position where an array size dimension is valid during runtime. If this causes a problem, we could include an additional directive to assist the parser (the same problem may occur in C). Fortran 90/95 also has compact array syntax. The parser should be capable to translate array syntax to be parallelized into (nested) DO loops, but this will impose restrictions on the use of the syntax and the arrays.

The main difference between the existing library and the new ones for the different languages will be the number of routines. Instead of creating many different routines for different data types and 1D and 2D (even 3D) arrays, a few generic routines can be shared. Even P- and E-type communications, the P-type with or without the X (eXchange) option, can be done with three routines for R2A, A2R and A2A traffic. Likewise, scalar and array reductions can be implemented by means of a few generic routines that must be called with a parameter that specifies the type of reduction.

At the moment separate communications for scalars and arrays are done by calling separately the necessary library routines. This involves much MPI overheads, especially for small messages where the "elapsed" time does not depend on the number of bytes (340 μ s for up to about 500 bytes using normal 100 Mb/s ethernet). For small messages the latency is important, whereas for large ones the bandwidth counts (we measured 85.8 Mb/s). Hence, one would think that "packing" scalars and arrays into one "payload" will reduce the overheads. This is probably true, because packing and unpacking involves copying from one position in memory to another position, which can be done at 83 Mbyte/s (Pentium III @ 450 MHz), a factor of 8 times faster than using ethernet. However, the final payload will still require a latency of 340 μ s and the effective bandwidth will be 85.8 Mb/s. Although the directive set allows to combine different scalar and array communications at one point in a program, it should be emphasized, again and again, that communications must be restricted to an absolute minimum. As a consequence, only the different scalars as specified by one directive might be combined by packing them together.

The use of a few generic routines enables the development of specific, optimized libraries for differ-

lines have no correlation with bottom lines. We never use this!

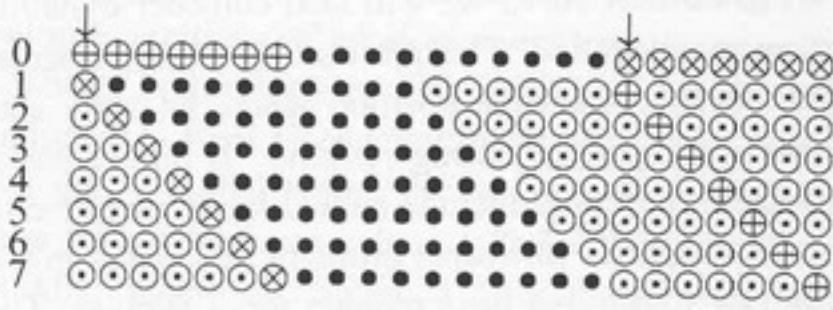


Figure 6: Node activity between a “toall” and a “toroot” call, marked by the vertical arrows, in the case of eight CPUs, assuming that there is no work before and after the calls. Symbols: $\oplus$ = send data; $\otimes$ = receive data; $\bullet$ = work; $\odot$ = idle (waiting).

ent cluster configurations and network topologies. For example, if a system (Linux kernel) uses multicast ethernet routing, the difference between R2A communications with E and P options might be very small, because root will output all packets only once. If this is the case, there is no need for specific R2A communications, and more effort could be spent in optimizing the A2R and A2A communications.

8 Communication optimizations

For the first versions the communication efficiency is less important; we aim at something which is more or less complete and which works. The root-to-all and all-to-root communications are all based on the same concept: after testing `spmd_myid` root will execute a loop over all non-root nodes to send or receive data and the non-root nodes will only receive or send data (virtual star topology). This solution is simple but it implies a lot of idle time, i.e. time that nodes are waiting to be served.

Let us compare some timings, assuming a cluster with 8 CPUs and expressing timings in units T which corresponds to the time necessary to send 1/8th of an array (in the case of big arrays we measured 85.8 Mb/s of the theoretical 100 Mb/s). Let us also assume that we start a timer before a “toall” call, and stop the timer after a “toroot” call. Between the two calls each node must process data, which takes NT seconds, say. This is illustrated in Figure 6 with $N = 10$.

(1) Serial execution takes $8 \times N \times T$ seconds. (2) Parallel execution takes $N \times T + 2 \times 7 \times T = (14 + N)T$ seconds: this includes sending twice 7/8th of the array. The speedup will be $8N/(14 +$

$N) = 8/(1 + 14/N)$. Obviously, the bigger N , or the workload, the better the speedup. (3) Root will be completely busy sending and receiving data, whereas other nodes will be receiving or sending 1/8th of the array. These are not idle times. Figure 6 shows that the total idle time ($\odot$ symbols) will be $2 \times 2 \times 3 \times 7 = 84T$, which is an average of $10.5T$ per node. If we take out the root node, because it will be completely busy with communications, the average even becomes $12T$. In words: all nodes, except root, will be idle 12 times the time it takes to transmit 1/8th of the array. The above calculation is an approximation assuming that there is no parallelism in the communications (synchronous `mpi_send` and `mpi_recv`). In practice, however, there may be substantial parallelism in the communications, e.g. during the reconstruction of array parts from ethernet packets.

In order to reduce idle times, we can adopt two strategies: (A) Instead of attributing root an equal share of the workload, its workload could be reduced, for example halved (in Fig. 6 there would only be 5 $\bullet$ s in row number 0). This solution involves another array partitioning and loop splitting, it will still be static, and it can be easily implemented. However, in the ideal case, when root once distributes data and once collects data, and the bulk of processing does not involve massive communications, root will not equally contribute to the workload. (B) As for the DODYN directive, it would be better to reserve root for only doing communications and simple, fast array processing where required. In this case the workload will be evenly distributed over the $(N - 1)$ other CPUs. This solution is ideal for clusters that are operated through a frontend, and it also can be implemented very easily.

Before we installed our new cluster consisting of eight Athlon single-CPU XP nodes, we experimented with a small cluster with dual-Pentium III nodes. The communications between the CPUs in one node (IN in Fig. 7) are faster than those between CPUs in different nodes (we measured a factor of 2.66, i.e. instead of 10.7 Mbyte/s (85.8 Mb/s) we measured 28 Mbyte/s). But 28 Mbyte/s is much slower than copying an array in memory (82 Mbyte/s). The reason for this difference is hidden in the TCP/IP stacks: an `mpi_send` goes down the stack, until the detection of the loopback, i.e. packets are not going out on the wire, and then up again to the matching `mpi_recv`.

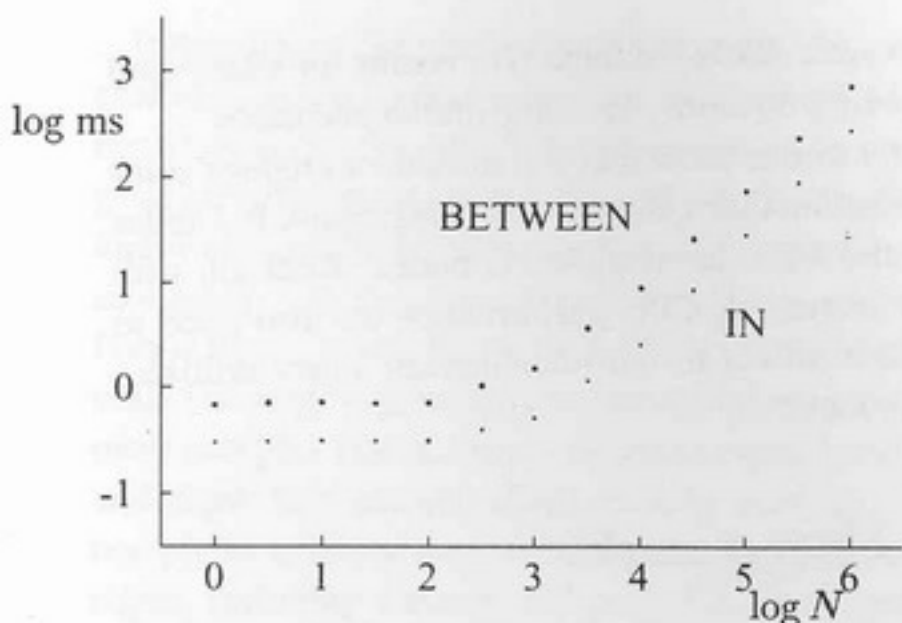


Figure 7: Ping-pong times in log(ms) as a function of log(N) in REALs on a dual-CPU cluster with fast ethernet. IN refers to communications between two CPUs inside one node. Single-trip times are half the ping-pong times.

There are two solutions for speeding up the communications: (a) replacing the interconnect by a more powerful one and (b) short-cutting the TCP/IP stacks. We studied (a) by replacing the ethernet by SCSI (Fig. 8). This was indeed faster, but not much because the SCSI protocol does not have a bus master (this was solved by circulating a “token”). Hence, IP-over-SCSI⁴ needs an alternative, preferably parallel interconnect with bus mastering. The second (b) problem, i.e. MPI-over-XXX has not yet been studied, although copying over SCSI by using the “p5” device layer was very efficient (the upper curve in Fig. 8). MPI-over-SCSI or MPI-over-XXX will involve the use of direct routing tables, such that the CPUs in dual-CPU nodes know their “neighbors.” In this case the entire TCP/IP stacks can be avoided by copying in memory (82 Mbyte/s), but not when doing communications between different nodes (10.7 Mbyte/s, unless another interconnect (XXX) is being used: p5-over-SCSI can attain 23 Mbyte/s for large payloads; Fig. 8).

Apart from dual-CPU nodes being more efficient in terms of space and maintenance, they can be more efficient in communications because the underlying routines can be rewritten to exploit parallelism. Here we study the updating of an array of 10^6 REALs on 4 CPUs in 2 nodes: an A2A communication in which each CPU has 1/4th of the array but each CPU needs the entire array. Hence, we will need at least the

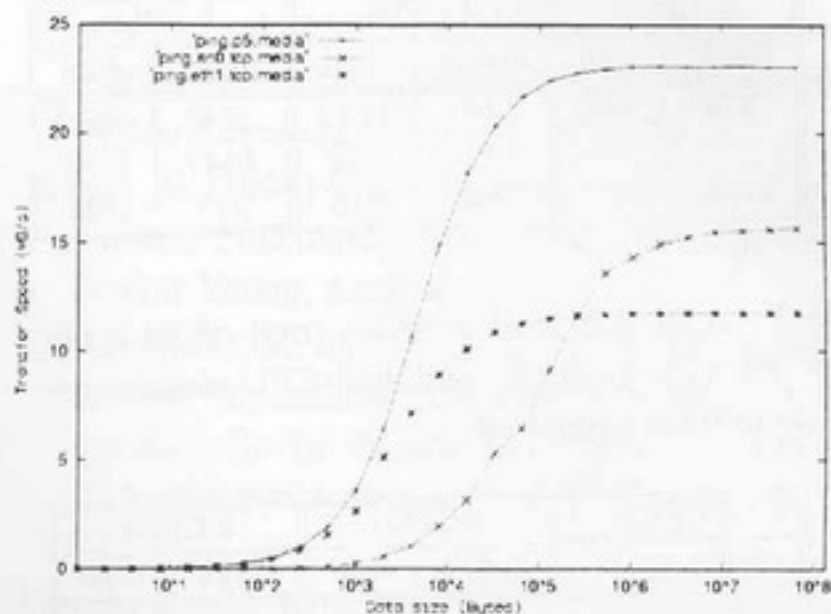


Figure 8: Measured bandwidths (Mbyte/s) as a function of the payload size. To the right, the lowest curve was obtained by using normal (fast) ethernet; the middle curve by IP-over-SCSI; the top curve by using the “p5” device over SCSI. See footnote 4 for seeing this graph in its full glory.

Size	IN	BETWEEN
1/1	142	373
3/4	106	280
1/2	72	187
1/4	36	94
1/8	18	47

Table 1: Single-trip times in ms as a function of the array fraction for 10^6 REALs. IN means inside a dual-CPU node, BETWEEN is between nodes.

transmissions of quarters (1/4) in an A2R routine and the entire array (1/1) in a R2A routine.

Table 1 gives single-trip times in ms for the payload sizes in REALs that we will need in order to compute communication times for $N = 10^6$ REALs. We can see that all numbers scale well with the size, i.e. sending the entire array takes four times the time of sending a quarter. This also follows from Fig. 7: the linearity of the curves down to 10^3 REALs, below which the communication overheads become more significant. This means that the speedup factors given below are valid until arrays of about 4000 bytes.

Scenario S1: CPUs 1, 2 and 3 send their 1/4 to 0 (A2R), after which root sends 1/1 to all others (R2A). All communications in this case are sequential. This implies for the A2R one IN of 36 ms plus two BETWEENs of 94 ms, a total of 224 ms. The

⁴See <http://ipoverscsi.sourceforge.net>

	2 CPUs		4 CPUs		8 CPUs	
	single	dual	single	dual	single	dual
S1	374	144	1401	1112	2940	2680
S2			1027	738	1447	1187
S3			748	516	937	706

Table 2: A2A distribution times (ms) of an array of 10^6 REALs on single- and dual-CPU clusters according to three scenarios.

	2 CPUs		4 CPUs		8 CPUs	
	single	dual	single	dual	single	dual
S1	0.7	0.3	2.1	1.7	4.9	4.5
S2			1.4	1.0	2.1	1.7

Table 3: RoA reduction times (ms) on single- and dual-CPU clusters for two scenarios.

R2A takes one IN of 142 plus two BETWEENs of 373, a total of 888 ms. Hence, S1 costs 1,112 ms.

Scenario S2: In the A2R CPU 1 sends 1/4 to 0 and, at the same time, 3 sends 1/4 to 2, after which 0 and 2 have each a different 1/2 of the array. This costs only one IN of 36 ms. Then 2 sends 1/2 to 0, one BETWEEN of 187. The A2R costs 223, the same as in the first scenario. We now apply the same parallelism in the R2A: first 0 sends 1/1 to 2 (one BETWEEN of 373), and then 0 and 2 send 1/1 simultaneously to 1 and 3 (one IN of 142). The R2A costs 515 ms, hence S2 costs only 738 ms.

Scenario S3: We can make one optimized A2A routine in which root does not collect and redistribute. In the first pass (a) 0 sends 1/4 to 1 and simultaneously 3 sends 1/4 to 2, which costs one IN of 36 ms. After this 1 and 2 have each 1/2. In the second pass (b) 1 sends 1/2 to 2 and 2 sends 1/2 to 1, which costs two BETWEENs of 187 or 374. Now 1 and 2 have both 1/1, but 0 and 3 need only 3/4. The third pass (c) costs only one IN of 106. We see that S3 only costs 516 ms, i.e. we have gained more than a factor of 2 relative to the first scenario.

Table 2 summarizes the results (4 CPUs) and compares these with times on single-CPU systems as well as a smaller (2 CPU) and larger (8 CPU) system. As expected, the times on dual-CPU systems are always less than those on single-CPU systems because of the smaller IN times.

Scalar RoA reductions can also be optimized. From Fig. 7 we see that the single-trip IN time is 0.15 ms and the BETWEEN time 0.35 ms for small

messages. Table 3 summarizes results for single- and dual-CPU systems, by using similar scenarios.

Our results show that it is possible to reduce communication times, not only by using dual-CPU nodes but also by using single-CPU nodes. After all, with ever increasing CPU performance we also need to squeeze all out of our interconnect: every millisecond counts!

9 Final remarks

There are many scientific areas in which cluster computing found a widespread use. In image processing, including cortical processing, cluster computing is now being introduced. I could not have titled this paper “High-cost CIP on low-cost clusters” because Cortical Image Processing is an area that is still not well established. In contrast, in Computational Fluid Dynamics or CFD this is already the case (Hauser et al., 2000), and cluster computing can enter the Gordon Bell competition concerning the price-performance ratio in megaflops per dollar. Hauser et al. were a GB finalist with US\$1.86 per MFLOPS in single precision, using the Kentucky Linux Athlon Testbed 2 (KLAT2), listing in the cost table 16 pizzas and 4 cases of soda for the students that helped putting all components together.

As an indication, our cluster of 8 AMD XPs at 1.6 GHz consists of normal PCs with, apart from the motherboards, CPUs and memories (512 MB per node), the cheapest components. Each node costs \$750, totaling \$6000, and I measured 5.8 GFLOPS, which gives about 1 dollar per MFLOPS. However, I did not include the other infrastructure, including a rack, two 1500 VA UPS systems, the ethernet switch and the frontend, a dual Pentium-III system. This infrastructure, especially the use of gigabit ethernet boards and switches, can easily double the price of a cluster. However, any luxury can be omitted when a lab’s budget is too limited. As shown in this paper, communication bandwidths can be increased by using alternative interconnects (IP-over-SCSI, still under construction) that will appear on the mass market with affordable prices, and communication times can be reduced by optimizing the communication scenarios in higher-level routines. Instead of paying a lot of money for state-of-the-art components, and using common libraries like MPI which are not very user friendly, it is possible to optimize performance and user friendliness by developing our own tools.

Returning to Cortical Image Processing, the main challenge at the moment is to optimize the processing schemes for the different cell models and to integrate all cells into a scheme that provides a complete image representation that can be used for specific visual tasks. On the basis of simple cells or complex Gabor filters, it will be possible to construct a multi-scale line/edge representation, bar and grating maps, more complex texture maps, and even extract motion and depth information, either directly from the Gabor phase information, from e.g. moving lines and edges, including vertices, or by combining all information in order to get a very robust system like our own visual system. I am convinced that there are numerous applications that could benefit from our efforts.

Acknowledgements

Research supported by the FCT project MOVIDE: POCTI/36445/PSI/2000 and FEDER, also by the FCT Programa Operacional Sociedade de Informação (POSI) in the framework QCA III. Images by courtesy of: Luís Santos (Figs 1 and 2), João Rodrigues (Fig. 3), Ulli Bobinger (Figs 4 and 5), Luis Sismeiro and Pedro Semeano (Fig. 8).

References

- du Buf, J.M.H. (1992) Modeling spatial vision at the threshold level. *Spatial Vision*, Vol. 6, pp. 25-60.
- du Buf, J.M.H. (1993) Responses of simple cells: events, interferences, and ambiguities. *Biol. Cybernetics*, Vol. 68, pp. 321-333.
- du Buf, J.M.H. and Fischer, S. (1995) Modeling brightness perception and syntactical image coding. *Optical Engineering*, Vol. 34, pp. 1900-1911.
- van Deemter, J.H. and du Buf, J.M.H. (2000) Simultaneous detection of lines and edges using compound Gabor filters. *Int. J. Pattern Recogn. and Artif. Intelligence*, Vol. 14, pp. 757-777.
- Gennart, B. and Hersch, R. (1999) Computer-aided synthesis of parallel image processing applications. In: *Parallel and distributed methods for image processing III*, Proc. SPIE 3817, pp. 48-61.
- Hauser, T. et al. (2000) High-cost CFD on a low-cost cluster. Available at <http://aggregate.org/KLAT2/>
- Heitger, F. et al. (1992) Simulation of neural contour mechanisms: from simple to end-stopped cells. *Vision Research*, Vol. 32, pp. 963-981.
- Krüger, N. (1998) Visual learning with a priori constraints. PhD thesis, Univ. of Bielefeld, Germany. Shaker Verlag, Aachen.
- Lueckenhau, M. and Eckstein, W. (1998) A thread concept for automatic task parallelization in image analysis. In: *Parallel and distributed methods for image processing II*, Proc. SPIE 3452, pp. 34-44.
- Petkov, N. and Kruizinga, P. (1997) Computational models of visual neurons specialised in the detection of periodic and aperiodic oriented visual stimuli: bar and grating cells. *Biol. Cybernetics*, Vol. 76, pp. 83-96.
- Santos, L.M. and du Buf, H. (2002a) Identification by Gabor features. Chapter 10 in: *Automatic Diatom Identification*, H. du Buf and M.M. Bayer (eds), World Scientific, Series in Machine Perception and Artificial Intelligence Vol. 51, pp. 187-220.
- Santos, L.M. and du Buf, J.M.H. (2002b) Computational cortical cell models for continuity and texture. *Biologically Motivated Computer Vision Workshop 2002*, Tuebingen, November 22-24 (in press).
- Seinstra, F., Koelma, D. and Geusebroek, J. (2001) A software architecture for user transparent parallel image processing on mimd computers. *Proc. 7th Int. Euro-Par Conf.*, Springer LNCS 2150, pp. 653-662.
- Snir, M., Otto, S., Huss-Lederman, S., Walker, D. and Dongarra, J. (1996) *MPI: The complete reference*. MIT Press.
- Squyres, J., Lumsdaine, A. and Stevenson, R. (1998) A toolkit for parallel image processing. In: *Parallel and distributed methods for image processing II*, Proc. SPIE 3452, pp. 69-80.
- Sterling, T., Salmon, J., Becker, D. and Savarese, D. (1999) *How to build a Beowulf*. MIT Press.