Available online at www.sciencedirect.com

SCIENCE DIRECT®

Image and Vision Computing 24 (2006) 870–881

image and vision COMPUTING

www.elsevier.com/locate/imavis

Automatic parameter regulation of perceptual systems

Daniela Hall *

INRIA Rhône-Alpes, Prima Group, 655, avenue de 1'Europe, 38330 Montbonnot, France

Received 16 June 2005; received in revised form 2 December 2005; accepted 16 February 2006

Abstract

Changes in environmental conditions frequently degrade the performance of perceptual systems. This article proposes a system architecture with a control component that auto-regulates parameters to provide a reduction in the sensitivity to environmental changes. We demonstrate the benefit of this architecture using the example of a long-term tracking system.

The control component consists of modules for auto-critical evaluation, for auto-regulation of parameters and for error recovery. Both modules require a measure of the goodness of system output with respect to a scene reference model. We describe the generation of the scene reference model and propose measures for the model quality and for the goodness of system output in form of measurement trajectories. Our self-adaptive tracking system achieves better recall than a manually tuned tracking system on a public benchmark data set.

Keywords: System architecture; Tracking; Autonomic computing; Learning

1. Introduction

Perceptual systems observe the world, interpret the observations in form of input signals such as images, audio data, or laser range data and communicate the result as numeric, vectorial or symbolic events. Such perceptual systems may be used for a diverse set of applications, including tracking systems in video surveillance, regulation systems for traffic control or image segmentation systems.

Stability, reliability and robustness are required for perceptual systems to be exploited widely in commercial applications. To meet these constraints, developers commonly design simple systems whose parameters can be adapted manually. Such systems perform well as long as the environment stays constant. Unfortunately, in most real applications the environmental conditions perceived by the sensors frequently change, which often breaks the system and requires reinitialisation and new hand tuning of the parameters. In this article, we propose a solution that makes hand tuning of parameters obsolete by a method for automatic parameter regulation and error recovery.

Shekhar [1] proposed a component based vision system with self-tuning capabilities. The result of the vision system is monitored by a user who provides high-level feedback on the quality of the result. An automatic rule based control engine translates the feedback to particular parameter values and algorithm choices. The rules for performance evaluation and for failure repair are generated manually from the expert knowledge of the designers. The system displays a similar functionality as the one, which we desire to obtain with our self-adaptive system. The big challenge of our system is to reduce the amount of human supervision and the manual definition of the rules for the control engine to a minimum.

Murino addresses the problem of automatic parameter regulation in [2]. He proposes a multi-layered component architecture. Each layer has its own set of parameters that are tuned such that the evidence (coming from the lower level) and the expectation (coming from the higher level) are consistent. The improvements are not convincing and the approach lacks the use of an external knowledge base.

In [3], Min proposes an approach for comparing the performance of different segmentation algorithms by searching the optimal parameters for each algorithm. He proposes an interesting multi-loci hill-climbing scheme on a coarsely sampled parameter space. The segmentation system performance is evaluated with respect to a given ground truth. This approach is designed for the comparison of algorithms and requires to test a large number of different parameter settings. For this reason, the utility of this approach for on-line parameter regulation is less appropriate.

* Tel.: +33 476 62 80 40; fax: +33 476 61 52 10.
E-mail address: daniela.hall@free.fr.

0262-8856/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2006.02.011

A different approach is proposed by Robertson and Brady [4]. They consider an image analysis system as a closed-loop control system that integrates knowledge in order to be self-evaluating. Measuring and comparing the system output to the desired output and applying a corrective force to the system leads to increased performance. They demonstrate their approach on the segmentation of aerial images using a bank of different filter operators. The system selects automatically the best filter for the current image conditions.

In our approach, we adapt recent innovations from autonomic software architectures [5] and implement them by an independent control component. In particular, our approach endows the system with capabilities of auto-critical evaluation (self-diagnosis), auto-regulation (self-optimisation) and self-healing (error recovery).

- Auto-critical evaluation: the system is able to judge its own performance.
- Auto-regulation: the capability of continually seeking opportunities to improve the performance of software components (such as parameter tuning).
- Self-healing (error recovery): the capability of automatically detecting, diagnosing and repairing localised software problems.

The article is organised as follows. Section 2 describes the architecture of a self-adaptive system. In Section 3, we define the steps for the design of a task specific self-adaptive system. Section 4 applies this approach to a robust tracking system. We demonstrate the utility of the approach in Section 5 and finish with conclusion and an outlook.

2. Architecture of a self-adaptive perceptual system

In this section, we propose an architecture for self-adaptive perceptual systems (shown in Fig. 1). As proposed by Robertson and Brady [4], we consider our system as a closed-loop control system. Monitoring the data flow makes it possible to estimate the current performance of the system and eventually apply a corrective force by control feedback to the basis component of the system.

The abilities of auto-critical evaluation, auto-regulation and error recovery are implemented as modules within the control component. The control component is separated from the perceptual system, which allows the application of this architecture to a wide range of systems. The output of the perceptual system is monitored by the module for auto-critical evaluation. Modules for auto-regulation and error recovery can be triggered by requests of the post-processing module. The post-processing module implements a function $f(s_t, s_{t-\Delta t})$ that either transmits the events or triggers the modules for auto-regulation or error recovery. The modules have access to a common knowledge base that contains the reference model and a database on repair strategies

$$f(s_t, s_{t-\Delta t}) = \begin{cases} \text{transmit event}, & \text{if } s_t > S \text{ and } s_{t-\Delta t} > S \\ \text{trigger error recovery}, & \text{if } s_t > S \text{ and } s_{t-\Delta t} \leq S \\ \text{trigger regulation}, & \text{if } s_t \leq S \text{ and } s_{t-\Delta t} \leq S \end{cases} \quad (1)$$

where s_t denotes the score provided by the auto-critical evaluation at time instant t and S denotes threshold defined a priori. This is a simple post-processing function. Depending on the application, a more complex function may be implemented.

The structure and content of this knowledge base is task dependent. This means that each system and each camera set-up may require a knowledge base with different structure and different content. The knowledge base can be hand-crafted or computed off-line by a learning process. It can be provided by the user or be acquired automatically. More advanced systems may be capable of subsequently updating and refining their knowledge base using incremental learning techniques.

3. Approach for designing a self-adaptive perceptual system

To implement the components of a self-adaptive system, several difficult problems need to be solved. This section

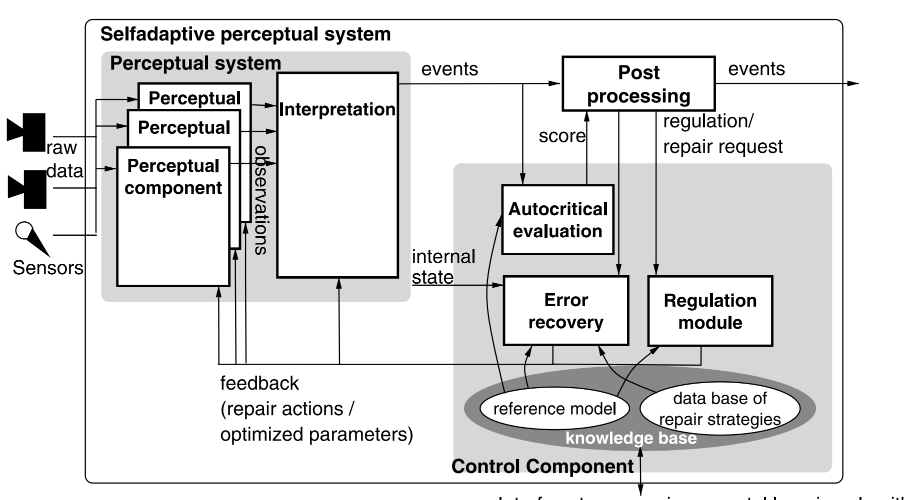

Fig. 1. Architecture of a self-adaptive perceptual system.