ERK'94

Portorož, Slovenija, 26. – 28. september 1994

Zbornik tretje Elektrotehniške in računalniške konference ERK'94
Proceedings of the Third Electrotechnical and Computer Science Conference ERK'94

Zvezek B / Volume B

Računalništvo in informatika / Computer and Information Science
Umetna inteligenca / Artificial Intelligence
Robotika / Robotics
Razpoznavanje vzorcev / Pattern Recognition
Biomedicinska tehnika / Biomedical Engineering
Študentski članki / Student Papers

Uredila / Edited by
Baldomir Zajc, Franc Solina

Slovenska sekcija IEEE / Slovenia Section IEEE

Zbornik tretje
Elektrotehniške in računalniške konference
ERK'94

26.–28. september 1994

Portorož, Slovenija

Zvezek B

Računalništvo in informatika
Umetna inteligenca
Robotika
Razpoznavanje vzorcev
Biomedicinska tehnika
Študentski članki

Uredila

Baldomir Zajc in Franc Solina

Slovenska sekcija IEEE

Ljubljana • Slovenija

CIP - Kataložni zapis o publikaciji
Narodna in univerzitetna knjižnica, Ljubljana

621.3(063)

ELEKTROTEHNIŠKA in računalniška konferenca (3 ; 1994 ; Portorož)
Zbornik tretje Elektrotehniške in računalniške konference ERK '94, 26. - 28. september 1994, Portorož, Slovenija. Zv. B / uredila Baldomir Zajc, Franc Solina. - Ljubljana : Slovenska sekcija IEEE, 1994

Besedilo slov. ali angl. - Na vzpor. nasl. str.: Proceedings of the Third Electrotechnical and Computer Science Conference ERK '94. - Vsebina na nasl. str.: Računalništvo in informatika ; Umetna inteligenca ; Robotika ; Razpoznavanje vzorcev ; Biomedicinska tehnika ; Študentski članki

ISBN 961-6062-04-2

1. Solina, Franc 2. Zajc, Baldomir. - I. Electrotechnical and Computer Science Conference (3 ; 1994 ; Portorož) glej Elektrotehniška in računalniška konferenca (3 ; 1994 ; Portorož). - II. ERK '94 glej Elektrotehniška in računalniška konferenca (3 ; 1994 ; Portorož)
42571008

Pri organizaciji Elektrotehniške in računalniške konference ERK'94
so sodelovala naslednja društva:

Elektrotehniška zveza Slovenije,
Društvo avtomatikov Slovenije,
Slovensko društvo za merilno-procesno tehniko (ISEMEC 94),
CIGRE,
Društvo za medicinsko in biološko tehniko Slovenije,
Društvo robotikov Slovenije,
Slovensko društvo za umetno inteligenco,
Slovensko društvo za razpoznavanje vzorcev,
Slovensko društvo za simulacijo in modeliranje.

Organizacijo konference in izdajo zbornika je finančno podprlo Ministrstvo za znanost in tehnologijo Republike Slovenije

Sponzor: Hermes SoftLab, Ljubljana

Slovenska sekcija IEEE
Fakulteta za elektrotehniko in računalništvo
Tržaška 25, 61001 Ljubljana, Slovenija

Tisk: SOMARU d.o.o. Ljubljana

Kazalo

Object-Oriented Analysis and Design of Image Segmentation Package*

Aleš Jaklič, Aleš Leonardis, and Franc Solina
Computer Vision Laboratory
Faculty of Electrical and Computer Engineering
University of Ljubljana
Tržaška 25, 61001 Ljubljana, Slovenia
Tel. +386 (61) 123 1121, Fax +386 (61) 264 990
alesj@ninurta.fer.uni-lj.si

Abstract

This paper demonstrates how important is software engineering in development of experimental software in computer vision. An example of object oriented analysis and design of an image segmentation package is presented to show how programming languages and design techniques not only allow us to implement our ideas but also provide means to refine our sloppy concepts. The recover-and-select paradigm is expressed at a higher level of abstraction than its concrete applications to different image and task domains. We hope that this example will foster better software engineering practice among computer vision researches, which would in turn allow code reuse and make experimentation less tedious.

1 Introduction

Application of recover-and-select paradigm [2] to image segmentation problem in computer vision produced promising results in segmentation of

- range images: in terms of biquadratic surface models [2] and superquadric volumetric models [3],
- edge images: in terms of subset of conics curve models [2].

In order to demonstrate the general value of the paradigm we have to experimentally verify its application to various image domains (range image, gray image, edge image, etc.) and task domains (recognition, manipulation, navigation, geometric modeling, etc.). This requires experimentation in finding the appropriate

- models,
- distance functions,
- model parameter estimation algorithms,
- growing algorithms,
- function measuring the quality of description, i.e. objective function
- and selection algorithms.

Since the recover-and-select paradigm expresses dynamic spatio-temporal behavior visualization of segmentation process, beside the quantitative error measures, provides the experimenter the critical feedback needed to asses the quality of selected models, distance functions, parameter estimation algorithms, growing algorithms, objective function, and selection algorithms.

All current implementations of the paradigm were done in C programming language within UNIX and X windows environment. They do not allow for easy experimentation. This is primarily due to purely functional software design, encouraged by C semantics, and low level of abstractions. To alleviate the problem we decided to re-design the software within the object-oriented paradigm, selecting C++ as object oriented programming language [4]. This will allow us to integrate the newly designed software with encapsulation of existing software for numerical optimization written in C.

2 Object-Oriented Analysis of Recover-and-Select Paradigm

In this section we describe the main classes of the problem domain, at the highest level of abstraction, together with the examples of concrete classes for particular image domains. The examples naturally suggest the class hierarchy. Also the dynamic behavior of description objects is presented. It is expressed

*This research was supported by The Ministry for Science and Technology of The Republic of Slovenia, Project J2-6187.

in terms of abstract classes, suggesting the use of polymorphism for concrete class objects. The class diagrams are presented in Booch notation [1].

World is a finite set of points from a finite dimensional vector space, with defined distance function $d(x, y)$ between the points. The distance function of a point from the subset V of the world is defined as

$$d(x, V) = \min_{y \in V} d(x, y). \quad (1)$$

The ϵ neighborhood of subset V is defined as a set of points, such that the distance of a point from the subset is less or equal ϵ, and the points are not members of the subset V

$$N_\epsilon(V) = \{x : d(x, V) \leq \epsilon \wedge x \notin V\}. \quad (2)$$

Model is an abstract entity with a defined distance $d(M, x)$ between the model and the point from the world. The distance between the model and the subset V of the world is in turn defined as

$$d(M, V) = \sum_{x \in V} d(M, x). \quad (3)$$

Model is usually parametrized with a finite number of parameters. We denote the parametric model as $M(\mathbf{p})$, where $\mathbf{p}$ is a finite dimensional vector of parameters with dimension $\dim(\mathbf{p})$. To use the model in the recover-and-select paradigm one must be able to find the parameters $\mathbf{p}_{min}$ that will minimize the distance between the model and the given subset V of the world

$$d(M(\mathbf{p}_{min}), V) = \min_{\mathbf{p}} d(M(\mathbf{p}), V). \quad (4)$$

Models do not exist in isolation from the world. As their name implies, they model the subsets of the world.

Description is a pair (V, M) of subset from the world and the model that describe this subset. The quality of subset representation is measured by description error ξ, which equals the distance between the model and the subset

$$\xi = d(M, V). \quad (5)$$

Descriptions are spatio-temporal dynamic objects. They evolve over time and space by

1. forming the neighborhood of the description,

$$N(M, V) = \{x : x \in N_\epsilon \wedge d(M, x) \leq \gamma\} \quad (6)$$

   if the set is empty the description enters a terminal state, from where it cannot evolve any more

2. finding the model for the union of the neighborhood and the world subset of the description, that minimizes the distance between the union and the model

$$d(M(\mathbf{p}_{min}), V \cup N(M, V)) = \quad (7)$$
$$\min_{\mathbf{p}} d(M(\mathbf{p}), V \cup N(M, V)). \quad (8)$$

3. if the distance of the model from the union is lower than a certain threshold, the description is updated by the union and the new model.

4. otherwise the description tries another type of model at step 2, if all the models have been already tried the description enters the terminal state and cannot evolve any more

Recover-and-select paradigm initiates a set of descriptions in the world, and then allow them to evolve in time and space. Since the evolution of each description is computationaly expensive the description that does not model the world well enough in cooperation with other descriptions is not allowed to evolve any further and is discarded from the overall description of the world. The selection of individual descriptions is based on the solution of the Quadratic Boolean Problem, where function $F(\mathbf{d})$ that measures the quality of the overall description of the world is maximised

$$F(\mathbf{d}) = \mathbf{d}^T \mathbf{Q} \mathbf{d} = \mathbf{d}^T \begin{bmatrix} c_{11} & \cdots & c_{1N} \\ \vdots & & \vdots \\ c_{N1} & \cdots & c_{NN} \end{bmatrix} \mathbf{d} \;. \quad (9)$$

Where $\mathbf{d}^T = [d_1, d_2, \ldots, d_N]$ is a vector of presence variables, d_i having the value 1 for the presence and 0 for the absence of the description D_i in the overall description. The diagonal terms of the matrix $\mathbf{Q}$ express the cost-benefit value of a particular description D_i,

$$c_{ii} = \mathrm{K}_1|V_i| - \mathrm{K}_2\xi_i - \mathrm{K}_3 \dim(\mathbf{p}_i) \;. \quad (10)$$

where K_1, K_2, K_3 are weights which can be determined on a purely information-theoretical basis. The off-diagonal terms handle the interaction between the overlapping descriptions

$$c_{ij} = \frac{-\mathrm{K}_1|V_i \cap V_j| + \mathrm{K}_2\xi_{ij}}{2} \;. \quad (11)$$

where the mutual description error equals

$$\xi_{ij} = \max(d(M_i, V_i \cap V_j), d(M_j, V_i \cap V_j)). \quad (12)$$

Maximizing the objective function $F(\mathbf{m})$ belongs to the class of problems known as combinatorial optimization problems. Since the number of possible solutions increases exponentially with the size of the problem, it is usually not tractable to explore them

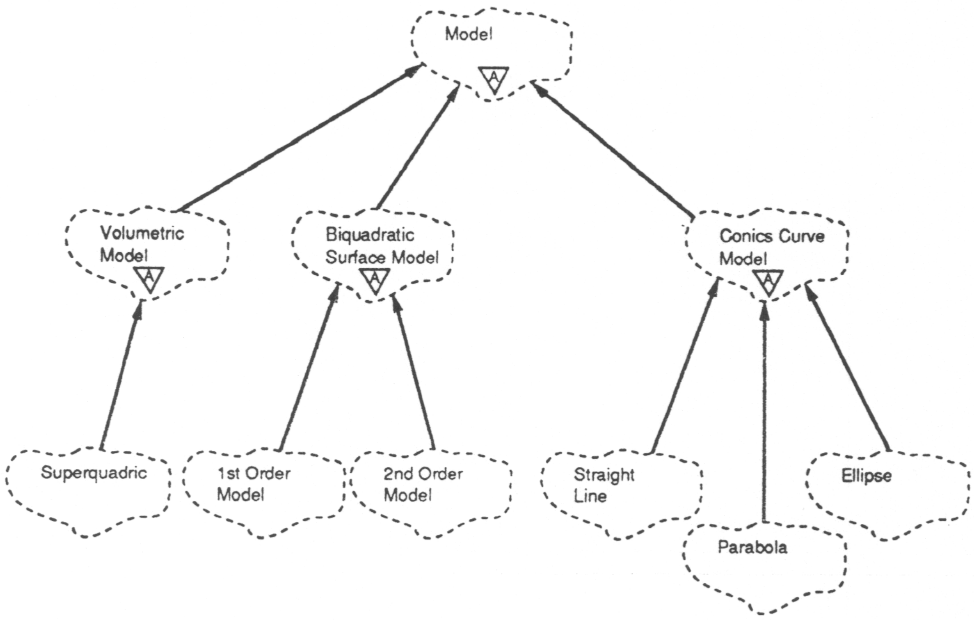

Figure 1: Class diagram of **models**

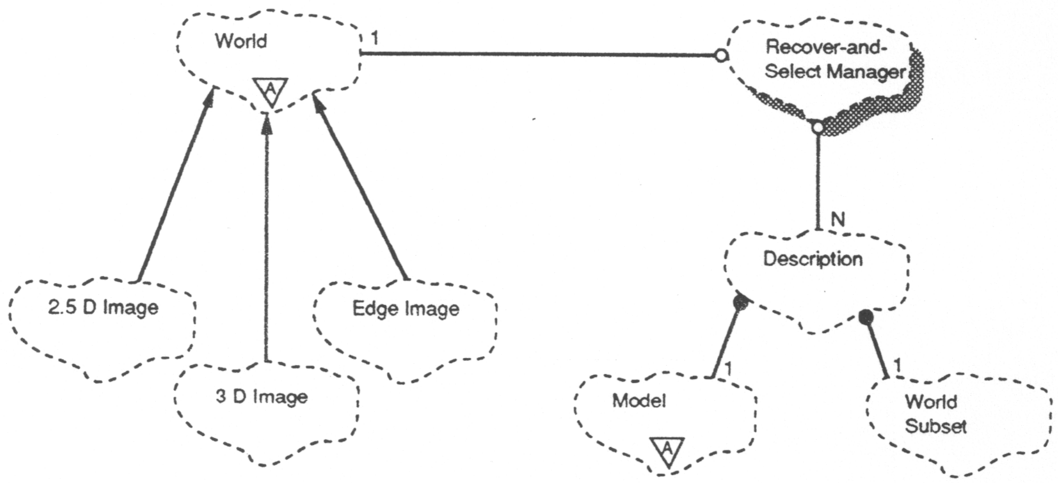

Figure 2: Class diagram of **world**, **description**, and **recover-and-select class utility**

exhaustively. A few approaches to the problem are described in [2]. For this analysis only the formation of the matrix $\mathbf{Q}$ is important since it involves the interaction of descriptions to calculate the off-diagonal terms.

3 Behavior of Recover-and-Select Manager

The manager first iterates the world set, and constructs the initial local descriptions. If the error of description is above the threshold then the description is destructed. The manager provides the initial type of model and the world subset to the description constructor. After the initial descriptions are constructed, the manager invokes an evolution method for each description, determining the extent of the evolution. If a description is not in a terminal state it will try to evolve. If it can not, it will enter a terminal state. After this evolution or recover stage, the manager invokes selection stage. The matrix $\mathbf{Q}$ is formed from selector member functions of description class returning $|V_i|$, ξ_i, and $\dim(\mathbf{p})$. The error ξ_{ij} and set cardinality $|V_i \cap V_j|$ of the overlapping descriptions are also formed through the selector member functions of description class, taking the other description as an argument. After solving the optimization problem the descriptions that are not selected are destructed. If there are any remaining description that are not in the terminal state, they are further evolved. This iteration is repeated until there are no descriptions in the nonterminal state. The final description of the world consists of all existent descriptions in terminal state.

4 Visualization

The main purpose of visualization is to monitor the evolution of descriptions. Though the recover-and-select paradigm is not restricted by dimensionality of the vector space, from which the points of world are drawn, we will restrict the discussion to the 2-D and 3-D images. In the first case there exists a natural mapping between the plane and the 2-D image, so the descriptions can be visualized by mapping corresponding subsets to the plane, and use color to distinguish between them. However, things get complicated if we deal with 3-D images, since a single projection from a chosen viewpoint is not sufficient to visualize the description completely. In this case we have to provide some means for experimenter to see the world from different viewpoints, again using color to identify different subsets. Besides, we might need a visualization of the model itself beside the subset that it models as is the case in segmenting the 3-D image with volumetric models [3].

5 Conclusions and Further Work

Object-oriented analysis and design of the image segmentation package based on recover-and-select paradigm not only produced a base for an implementation in an object oriented programming language but also produced a more concise and generalized description of recover-and-select paradigm. This might lead to applications of the paradigm to similar problems in machine learning.

Currently, we are working on an implementation of the package, with concrete application to range image segmentation with superquadric volumetric models. Thus approaching the first iteration of the micro development process (see [1]).

References

[1] G. Booch. *Object-oriented Analysis and Design.* Benjamin/Cummings, second edition, 1993.

[2] A. Leonardis. *Image Analysis Using Parametric Models.* PhD thesis, Faculty of Electrical Engineering and Computer Science, University of Ljubljana, 1993.

[3] A. Leonardis, F. Solina, and A. Macerl. A direct recovery of superquadric models in range images using recover-and-select paradigm. In *Proceedings of Third European Conference on Computer Vision, Vol. I*, pages 309–318, 1994.

[4] B. Stroustrup. *The C++ Programming Language.* Addison-Wesley, second edition, 1993.