

Dimitri Plemenos
Georgios Miaoulis (Eds.)

Intelligent Computer Graphics 2010



Springer

Dimitri Plemenos and Georgios Miaoulis (Eds.)

Intelligent Computer Graphics 2010

Studies in Computational Intelligence, Volume 321

Editor-in-Chief

Prof. Janusz Kacprzyk
Systems Research Institute
Polish Academy of Sciences
ul. Newelska 6
01-447 Warsaw
Poland
E-mail: kacprzyk@ibspan.waw.pl

Further volumes of this series can be found on our homepage: springer.com

- Vol. 296. Roger Lee (Ed.)
Software Engineering Research, Management and Applications, 2010
ISBN 978-3-642-13272-8
- Vol. 297. Tania Tronco (Ed.)
New Network Architectures, 2010
ISBN 978-3-642-13246-9
- Vol. 298. Adam Wierzbicki
Trust and Fairness in Open, Distributed Systems, 2010
ISBN 978-3-642-13450-0
- Vol. 299. Vassil Sgurev, Mincho Hadjiski, and Janusz Kacprzyk (Eds.)
Intelligent Systems: From Theory to Practice, 2010
ISBN 978-3-642-13427-2
- Vol. 300. Baoding Liu (Ed.)
Uncertainty Theory, 2010
ISBN 978-3-642-13958-1
- Vol. 301. Giuliano Armano, Marco de Gemmis, Giovanni Semeraro, and Eloisa Vargiu (Eds.)
Intelligent Information Access, 2010
ISBN 978-3-642-13999-4
- Vol. 302. Bijaya Ketan Panigrahi, Ajith Abraham, and Swagatam Das (Eds.)
Computational Intelligence in Power Engineering, 2010
ISBN 978-3-642-14012-9
- Vol. 303. Joachim Diederich, Cengiz Gunay, and James M. Hogan
Recruitment Learning, 2010
ISBN 978-3-642-14027-3
- Vol. 304. Anthony Finn and Lakhmi C. Jain (Eds.)
Innovations in Defence Support Systems, 2010
ISBN 978-3-642-14083-9
- Vol. 305. Stefania Montani and Lakhmi C. Jain (Eds.)
Successful Case-Based Reasoning Applications-I, 2010
ISBN 978-3-642-14077-8
- Vol. 306. Tru Hoang Cao
Conceptual Graphs and Fuzzy Logic, 2010
ISBN 978-3-642-14086-0
- Vol. 307. Anupam Shukla, Ritu Tiwari, and Rahul Kala
Towards Hybrid and Adaptive Computing, 2010
ISBN 978-3-642-14343-4
- Vol. 308. Roger Nkambou, Jacqueline Bourdeau, and Riichiro Mizoguchi (Eds.)
Advances in Intelligent Tutoring Systems, 2010
ISBN 978-3-642-14362-5
- Vol. 309. Isabelle Bichindaritz, Lakhmi C. Jain, Sachin Vaidya, and Ashlesha Jain (Eds.)
Computational Intelligence in Healthcare 4, 2010
ISBN 978-3-642-14463-9
- Vol. 310. Dipti Srinivasan and Lakhmi C. Jain (Eds.)
Innovations in Multi-Agent Systems and Applications - 1, 2010
ISBN 978-3-642-14434-9
- Vol. 311. Juan D. Velásquez and Lakhmi C. Jain (Eds.)
Advanced Techniques in Web Intelligence, 2010
ISBN 978-3-642-14460-8
- Vol. 312. Patricia Melin, Janusz Kacprzyk, and Witold Pedrycz (Eds.)
Soft Computing for Recognition based on Biometrics, 2010
ISBN 978-3-642-15110-1
- Vol. 313. Imre J. Rudas, János Fodor, and Janusz Kacprzyk (Eds.)
Computational Intelligence in Engineering, 2010
ISBN 978-3-642-15219-1
- Vol. 314. Lorenzo Magnani, Walter Carnielli, and Claudio Pizzi (Eds.)
Model-Based Reasoning in Science and Technology, 2010
ISBN 978-3-642-15222-1
- Vol. 315. Mohammad Essaaidi, Michele Malgeri, and Costin Badica (Eds.)
Intelligent Distributed Computing IV, 2010
ISBN 978-3-642-15210-8
- Vol. 316. Philipp Wolfrum
Information Routing, Correspondence Finding, and Object Recognition in the Brain, 2010
ISBN 978-3-642-15253-5
- Vol. 317-320. xxx
- Vol. 321. Dimitri Plemenos and Georgios Miaoulis (Eds.)
Intelligent Computer Graphics 2010
ISBN 978-3-642-15689-2

Dimitri Plemenos and Georgios Miaoulis (Eds.)

Intelligent Computer Graphics 2010



Springer

Dimitri Plemenos
University of Limoges
44, rue Antoine Bourdelle
87000 Limoges
France
E-mail: plemenos@numericable.com

Dr. Georgios Miaoulis
Technological Education Institute of Athens
Department of Computer Science
Ag. Spyridonos
Egaleo, 122 10 ATHENS
Greece
E-mail: gmiaoul@teiath.gr

ISBN 978-3-642-15689-2

e-ISBN 978-3-642-15690-8

DOI 10.1007/978-3-642-15690-8

Studies in Computational Intelligence

ISSN 1860-949X

Library of Congress Control Number: 2010934962

© 2010 Springer-Verlag Berlin Heidelberg

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilm or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Typeset & Cover Design: Scientific Publishing Services Pvt. Ltd., Chennai, India.

Printed on acid-free paper

9 8 7 6 5 4 3 2 1

springer.com

Preface

Nowadays, intelligent techniques are more and more used in Computer Graphics in order to optimise the processing time, to find more accurate solutions for a lot of Computer Graphics problems, than with traditional methods, or simply to find solutions in problems where traditional methods fail.

The purpose of this volume is to present current work of the Intelligent Computer Graphics community, a community growing up year after year. This volume is a kind of continuation of the previously published Springer volumes “Artificial Intelligence Techniques for Computer Graphics” (2008) and “Intelligent Computer Graphics 2009” (2009).

What is Intelligent Computer Graphics? It is a set of Computer Graphics problems whose solution is strongly improved by the use of intelligent techniques. These techniques are mainly based on Artificial Intelligence. So, in Declarative scene Modelling, problem resolution, constraint satisfaction and machine-learning techniques are used. In scene understanding, as well as in improved Monte-Carlo Radiosity, heuristic search techniques allow to improve solutions. In virtual world exploration, efficient camera movement is achieved by strategy games techniques. In behavioural animation, multi-agent techniques, as well as evolutionary algorithms are currently used.

However, it is obvious that techniques based on Artificial Intelligence cannot resolve all kinds of problems. In some cases, the use of specific Artificial Intelligence techniques may become too heavy and even inefficient, while, sometimes, simple human intelligence, easy to implement, can help find interesting solutions in cases where traditional Computer Graphics techniques, even combined with Artificial Intelligence ones, cannot propose any satisfactory solution. Such a case is the one of visual scene understanding, where it is sometimes easy to know what kind of view is expected by the user. Another case where the use of simple human intelligence is often requested is date visualisation, when a little bit of imagination can give interesting results.

During a long time, Artificial Intelligence techniques remained unknown and unused for Computer Graphics researchers, while they were already used in other graphic processing areas like image processing and pattern recognition. We think that the 3IA International Conference on Computer Graphics and Artificial Intelligence, organised since 1994, grandly contributed to convince many Computer Graphics researchers that intelligent techniques may allow substantial improvements in a lot of Computer Graphics areas. Nowadays, more and more researchers in Computer Graphics all over the world are interested in intelligent techniques. We think, the main contribution of techniques issued from Artificial Intelligence is to allow invention of completely new methods in Computer Graphics, often based

on automation of a lot of tasks assumed in the past by the user in an imprecise, often inefficient and (human) time consuming manner.

For Computer Graphics researchers it is important to know how the use of intelligent techniques evolves every year and how new intelligent techniques are used in new areas of Computer Graphics year after year.

When the 3IA International Conference on Computer Graphics and Artificial Intelligence was first created in 1994, its purpose was to put together Computer Graphics researchers wishing to use Artificial Intelligence techniques in their research areas, in order to create emulation among them. Nowadays, sixteen years after the first 3IA International Conference (3IA'94), the number of Computer Graphics researchers using Artificial Intelligence techniques became very important. Thus, an additional purpose of the 3IA Conference is to keep researchers informed on the existence of new intelligent methods, and even of corresponding software, for specific areas of Computer Graphics.

This volume contains selected extended papers from the last 3IA Conference (3IA'2010), which has been held in Athens (Greece) in May 2010. This year papers are particularly exciting and concern areas like rendering, viewpoint quality, data visualisation, vision, computational aesthetics, scene understanding, intelligent lighting, declarative modelling, GIS, scene reconstruction and many other important themes.

We think that in Computer Graphics still exist a lot of areas where it is possible to apply intelligent techniques. So, we hope that this volume will be interesting for the reader and that it will convince him (her) to use, or to invent, intelligent techniques in Computer Graphics and, maybe, to join the Intelligent Computer Graphics community.

Dimitri Plemenos
Georgios Miaoulis

Contents

An Introduction to the <i>Bio-logic</i> of Artificial Creatures	1
<i>Y. Duthen, H. Luga, N. Lassabe, S. Cussat-Blanc, T. Breton, J. Pascalie</i>	
A Cutaway Visualization Technique for Understanding Scenes with Embedded Objects	25
<i>Nancy Dandachy, Dimitri Plemenos, Safwan Chendeb, Bachar El Hassan</i>	
Surface Reconstruction with Smart Growing Cells	47
<i>Hendrik Annuth, Christian-A. Bohn</i>	
Collaborative Foreground Background Object Isolation and Tracking	67
<i>John Dragonas, Anastasios Doulamis, George Miaoulis, Dimitri Plemenos</i>	
Using Fuzzy Logic to Describe Vague Solid Models	85
<i>Angel A. Zaldivar, Juan C. Torres, Alejandro Leon</i>	
A GIS Platform for Automatic Navigation into Georeferenced Scenes Using GIS Scene Explorer (GIS-SE)	105
<i>Christos P. Yiakoumetsis, Georgios Bardis, Georgios Miaoulis, Dimitri Plemenos, Djamchid Ghazanfarpour</i>	
A Level of Interaction Framework for Exploratory Learning with Characters in Virtual Environments	123
<i>David Panzoli, Christopher Peters, Ian Dunwell, Stéphane Sanchez, Panagiotis Petridis, Aristidis Protopsaltis, Vincent Scesa, Sara de Freitas</i>	

A Comparison of Two Machine Learning Approaches for Photometric Solids Compression	145
<i>Samuel Delepoule, François Rousselle, Christophe Renaud, Philippe Preux</i>	
A Collaborative Framework for Virtual Globe Based 3D City Modeling	165
<i>Christos P. Yiakoumetsis, Georgios Bardis, Georgios Miaoulis, Dimitri Plemenos, Djamchid Ghazanfarpour</i>	
A Computerized Method to Generate Complex Symmetric and Geometric Tiling Patterns	185
<i>Ali Izadi, Mahdi Rezaei (Ghahroudi), Azam Bastanfar</i>	
Personalized View Selection of 3D Molecular Proteins	211
<i>Nikolaos Doulamis, Evangelos Chronis, George Miaoulis, Dimitri Plemenos</i>	
Interactive Volume-Based Indirect Illumination of Dynamic Scenes	229
<i>Athanasis Gaitatzes, Pavlos Mavridis, Georgios Papaioannou</i>	
Author Index	247

An Introduction to the *Bio-logic* of Artificial Creatures

Y. Duthen, H. Luga, N. Lassabe, S. Cussat-Blanc, T. Breton, and J. Pascalie

Abstract. In this chapter we emphasize the most crucial developments in the design of artificial creatures starting with the seminal work of Karl Sims presented at the SIGGRAPH'94 conference and ending with the latest research in virtual beings based on cell development.

Nowadays much research relies on the same principles in order to generate more and more complex artificial beings but also to advance evolutionary robotics with prototypes capable of self-modeling, self-reproduction and self-assembly. Inspired by biology, this research moves closer to nature paradigms, considering the creature as the result of a developmental process of a single element: a cell. This logic, presented here, as the *bio-logic* of artificial creature, needs not only new simulators such as chemical or hydrodynamic environments but also new mechanisms such as *an artificial regulatory network* to control the development process. Although these new beings are endowed with a very simple metabolism, they are still capable of transforming their local environment in order to grow and to piece together “logical organs”.

1 Introduction

The automatic design of such complex systems as artifacts in virtual worlds or real machines exhibiting the same ‘strong’ properties of living organisms represents a considerable challenge. This ambitious aim was expressed in 1966 by the pioneer of computer science, J. Von Neumann in his works on the first self replicating *cellular automata* [68]. In 1998 Daniel Mange developed the concept of “*Life in Silico*” and “*bio-inspired machines*”, and his definitions of living organisms are the keystone of much recent research. His approach relies on three main biological concepts that can encompass many mechanisms of computer science:

- *The phylogenetic* model: this aspect concerns the ability to evolve, to transform genetic material to produce different species,
- *The ontogenetic* model: this concerns the ability to grow from a single cell by division and differentiation,

Y. Duthen · H. Luga · N. Lassabe · S. Cussat-Blanc · T. Breton · J. Pascalie

VORTEX Research Team, IRIT UMR 5505

University of Toulouse, France

e-mail: duthen@irit.fr

- *The epigenetic* model: this last model includes the ability to learn like the nervous system or the immune system,

and he introduced the sentence: “... if a machine has these three properties, it could be difficult to distinguish it from a living organism” [67].

In this chapter we will concentrate specifically on the two first concepts: phylogeny and ontogeny, the last concept can be seen as a very complex emergent function, a consequence of the interactions between a physiology and an environment. One main aim of research into evolutionary creatures is to define a general goal for a system and let the system auto-organize, develop, auto-reconfigure or design others systems... in order to solve problems [41, 42, 43].

The new sciences associated with this approach are entitled “*morphogenetic engineering, embryomorphic engineering, machines that can design others machines, etc.*”

In this context we have labeled the approach *bio-logic* and this chapter is a general introduction to the main underlying mechanisms for designing such systems.

In section 2, we categorize through history the kind of creature we will have to deal with, in section 3 we emphasize environmental considerations: the importance of the world where the creatures will grow. Section 4 gives an overview of the main research into creatures assembled from pieces. Section 5 is a transition section that introduces the ontogenetic approach and notions as metabolism. An overview of developmental models is then shown in section 6.

2 What Kind of Creatures?

The term Artificial Creatures in the graphics community generally refers to characters in a virtual world with humanlike aspects or avatars. Since the pioneering work of Nadia Magnenat Thalmann and Daniel Thalmann [1] illustrated in figure 1, many techniques have been developed to produce high quality rendering and animation controlled by hand-made descriptions, by motion or behavioral captures or by user-predefined scenarios and scripts. In the present field of games or serious games applications where immersion and interaction with the user become crucial, one of the main challenges for autonomous entities and NPCs (Non-Player Characters) concerns the credibility [2] of behavior. The goal is to provide high-level behavior and the ability for entities to adapt and anticipate situations dynamically in unknown environments. Related mechanisms usually deal with models of machine learning (neural networks, classifiers systems, etc).

In this chapter we focus on quite different artifacts. We consider that a virtual creature is, in a general way, the result of a program that automatically synthesizes a shape but also a behavior (reactive behavior at this level) and sometimes a metabolism in a simulated environment. This means the process of building (constructing or developing) these entities has to include provision for the temporal



Fig. 1
“Rendez vous à
Montreal” D. Thal-
mann 1987

interaction between the output of the generator and the environment in order to improve creature-solution.

To categorize the evolution of these creatures briefly we can divide major research into two classes:

- Creatures assembled from pieces such as blocks, sticks, artificial muscles and controllers usually based on neural networks,
- Creatures developed from seeds such as a grammar, a generator, or an artificial cell.

Creatures assembled from pieces

The first class is illustrated in figure 2 by some important work in the field.

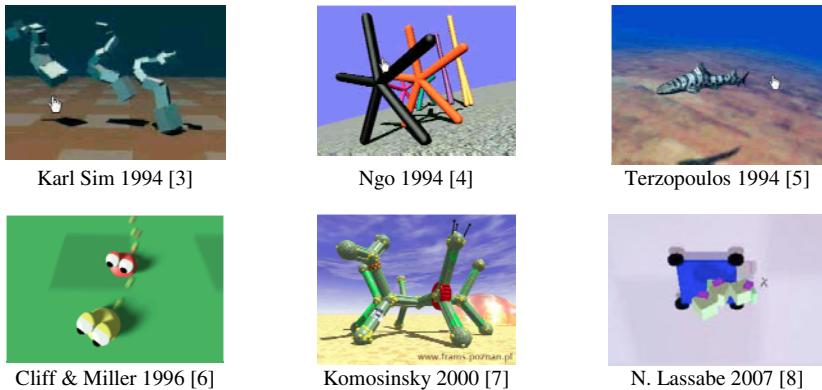


Fig. 2

In all of these cases the common process of exploring the research space (for shapes and behaviors) of such creatures is based on an evolutionary algorithm able to recombine individuals characteristics coded into a population of genotypes. The interpretation (called morphogenesis) of each genotype produces a phenotype loaded in an environment. This phenotype is a creature whose behavior is evaluated in a simulator which generally takes only Newtonian forces into account. The management of mass, force and collisions by the dynamic engine (such as ODE, Bullet, etc) leads to movement or locomotion. Different criteria embedded in the fitness function serve to select the well-adapted individual for recombination in the evolution loop.

Creatures developed from seeds

The next step in the design of more complex creatures is to consider that the result of the morphogenesis (the phenotype) is a seed (such as an artificial cell), which will grow thanks to duplication or division mechanisms triggered by conditions supplied by the environment. The creature develops therefore its structure but also

its function simultaneously, and so we can expect to obtain from such a process some properties of a living organism such as robustness, reproduction and adaptation, auto-organization or self-repairing capabilities.

This second class is illustrated in figures 3, 4, 5



Tree (shoot and root) developed by S. Bornhofen from a L-System and a transfer-resistance mechanism managing light and nutriments [9]

Fig. 3

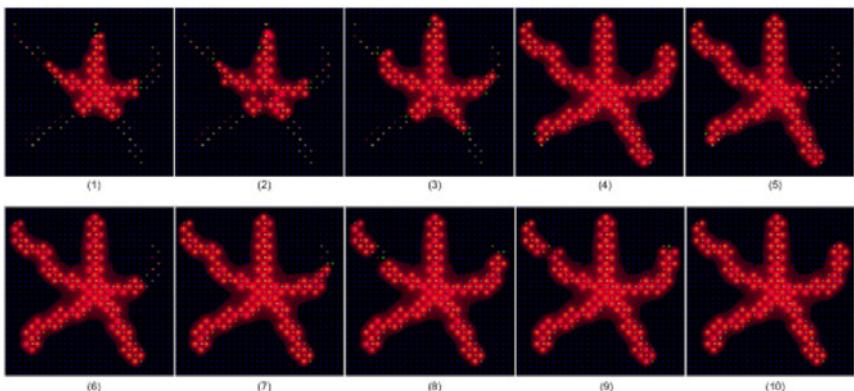


Organisms produced by Joachimczak with a physical 3D-environment [10]



Organism produced by Eggenberger [11], able to move thanks to a morphological modification induced by a gene regulatory network.

Fig. 4



Starting from one cell with an artificial chemistry, S. Cussat-Blanc's organisms [12] grow thanks to a metabolism, and exhibit self-repairing abilities [66], VORTEX team.

Fig. 5

The three last examples integrate a special mechanism: an Artificial Regulatory Network (ARN), following the vital *bio-logic* lead emphasized here. Bio-inspired ARN [13] are used to control the activity of parts of the genotype coding different functions of the cells. External signals enhance or inhibit regions and we will see in section 6 that this regulation takes on a very particular importance during the cell division phase.

It is therefore worthy of note that a cell is now defined by a genotype aggregating its basic behaviors in an environment with a simulated chemistry, and an ARN, which regulate this behavior. To evaluate this kind of more complex creature during the evolution loop, the environment must consequently supply a simulator of an artificial chemistry. This means the environment must furnish more and more physical simulators in order for the fitness function to give a good evaluation of each individual. A *hydrodynamic* simulator, an *artificial chemistry* simulator and a *dynamic* engine have to be considered at different levels during the growing process of the creature and during the period of its artificial life. This is developed in the next section.

3 The Importance of Simulated Environments

Once the evolutionary computation framework is defined we need to allow the creature prototype to act in virtual worlds. These worlds, for which creatures are designed, have their own laws. To offer the creature the means of finding a behavior, they need to assume the laws of the worlds they are living in. Why do we need to make these laws explicit? Because, as demonstrated by Pfeifer and Bongard's book, "*the body shapes the way we think*" [14]. In other words, living beings behave according to their morphology and, as introduced by Darwin in [15], morphology develops in a long process called *evolution*. In this process, environmental pressures are the hands that shape living beings generation by generation. To reproduce these processes *in silico*, we need to simulate that kind of environmental pressure and thus we need to simulate a set of physical forces.

The Newtonian Simulator: a way to improve motion

The first kind of realistic environment simulation was achieved by incorporating the laws of physics in the interaction loop between creatures and their environment. A physical simulator embeds several components used to describe motion on the one hand and to calculate its consequences along the course of time on the other. Karl Sims was one of the pioneers of artificial development coupled with artificial physics [3]. In his work, the environment embeds a physics engine that allows frictional forces, gravity and a kind of viscosity.

The accuracy of the physical simulation was put to the fore in 2000 by Taylor and Massey [16] which tried to re-implement Sim's model using the MathEngine library to solve physical constraints. They demonstrate issues around the step size

of physical integration and conclude by underlining the need for a compromise between accuracy and computation time for a given individual. Following works have used physics in the same way as Karl Sims, using collision detection as medium response and frictional forces to improve motion. Some of these models have been transferred to the real world by the means of robotics and they have shown the limits of virtual physics faced with real physics. Indeed the robots did not have as precise a motion as those found in the simulator [17,18].

Recently, some physics engines have been used to maintain the morphology of artificial creatures. Morphologies built around the *tensegrity* model or spring-mass model need a physics engine to maintain three-dimensional morphology over time [19].

Physics engines are also used for a second class of creatures, those developed from cells. In the field of artificial embryogeny, the physics engine computes lower level collisions between deformable and/or mobile cells.

Artificial chemistry: a way to improve metabolism

Dittrich and his colleagues formally introduced the concept of artificial chemistry in 2001 [20]. They defined artificial chemistry as “*a man-made system, which is similar to a real chemical system*”. They formalize it with a triple (S, R, A) in which S is the set of possible molecules, R is the set of interaction rules between molecules of a subset of S , A is a reaction algorithm that defines the application method of rules in R to a collection P of all possible molecules of the environment (with the possibility of repetition if one or more molecules are available). This collection is also named reactor, soup or population.

This approach has mainly been used to simulate membrane development and the maintaining of one cell or a small group of cells. Ono and Ikegami have adopted this formalism to build a simple *protocell*¹ model that simulates the chemical dynamics in a 2-D matrix [21]. This model can simulate a cell division. By modifying repulsive forces between molecules, various kinds of divisions have been obtained, as illustrated in figure 4. For a certain range of values, system instability causes the death of some cells due to membrane destruction.

Rasmussen et al. propose in [22] a more simple approach that simulates one protocell able to use its environment to feed itself. A part of the cell metabolism is simulated by means of thermodynamic data provided by *in vivo* experiments.

These kinds of models have exploited artificial chemistry as a metabolism tool, but it can also be a powerful tool to explore morphologies for a given organism.

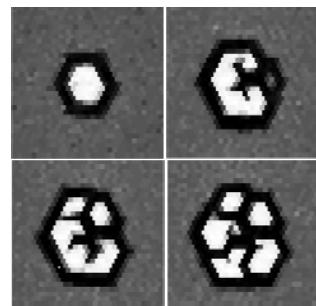


Fig. 4 Simulation of a protocell division. Ono & Ikegami.

¹ A simple cell without nucleus.

Knowledge from the field of developmental biology teaches us that *positional* information is embedded in the cell environment. With this hypothesis, chemical simulators became tools to explore morphologies.

The Reaction-Diffusion model: a way to explore morphologies

Alan Turing produced the first work on a cell developmental model in 1952. He suggested that a reaction-diffusion model could be a good approach to model cell development from a chemical point of view [23]. In his model, a set of differential equations gives the evolution rules of *morphogens* concentration in an environment. In good conditions and when stabilized, this system creates interesting patterns, that are close to real natural chemical patterns.

This model inspired many of past and current models. However, it is mainly used to describe lower level phenomena like chemical substance spread in the environment. Other bio-inspired mechanisms have been introduced to add cell interaction with this kind of environment, mainly used for the spread of morphogens produced by organisms.

Hydrodynamic simulations: a new tool for positional information

This simulator manages hydrodynamic substrate interactions. Its main aim is to propose a method inspired by the gastrulation stage of some living beings (vertebrates) to position morphogens. This early stage of organism development allows

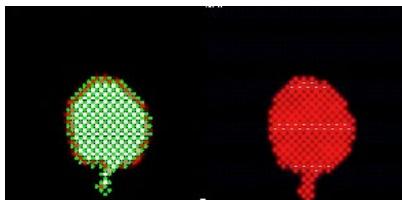


Fig. 5 Use of hydrodynamic constraints to position morphogens across the environment. VORTEX Team.

the morphogen positioning of the embryo in its near environment. It then allows the development of its organs. These theories put to the fore the physical interaction leading to embryo patterning. The egg is considered as a viscous medium composed of cells and limbs where movement is induced by inertia and hydrodynamic physics.

To reproduce this kind of dynamic in bio-inspired models based on morphology development, hydrodynamic simulation

appears to be a promising tool to replace the classic spreading algorithm seen in most models. In our previous works we thus proposed the introduction of this kind of simulator in order to spread morphogens along the environment as illustrated in figure 5. The experiments carried out have shown that this model is difficult to parameterize but offers a high dynamic in the search for a specific morphology [24].

After this rapid overview of the importance of the simulated environment, the next chapter will focus on the first class of artificial creatures, those assembled from pieces.

4 Phylogeny of Creatures Assembled from Pieces

Due to the huge amount of resources required for running simulations involving autonomous entities, the abilities of simulated virtual creatures have always been related to the available computing power. The development of opportunities for creating virtual creatures has consequently followed the emergence of increasingly powerful computers.

Historically, the first autonomous artificial creature simulators used simple, low dimensional universes as environments. Rather than providing realistic 3D creatures, their aim was to study the emergence of self organization [25, 26] in a controlled environment where parameters which control behavior could be tuned to validate a series of assumptions against real data sets. These first experiments show that quite simple agents could act to solve quite complex tasks.

The artificial creatures in Ray's "*Tierra*" simulator [69, 70] become alive in 1991 as computer programs which compete to survive in an environment composed of a simulated virtual computer. Memory of this restricted environment is initialized with a simple self-replicating code². After some evolution (mutations, replications) has occurred the creatures start to exhibit behaviors found in nature: parasitism, viral behavior and enhanced replication speeds could be observed.

To enhance the results and link the simulator with more natural looking creatures, Ventrella proposed a model and a tool: Genepool [71], where worm-like creatures behave in a 2D simulated Petri box. This environment took into account more complex interactions because creatures were endowed with characteristics which included the notions of species and energy management. Its creatures could move in the environment and thus could only reproduce locally with the surrounding creatures. They could sense the energy (the virtual food), which was necessary for their survival. One drawback was that the increase in the rendering of the creatures restricted the degree of freedom in their internal characteristics, and thus did not allow exhibiting as much complexity in the creature's relationship as in *Tierra*.

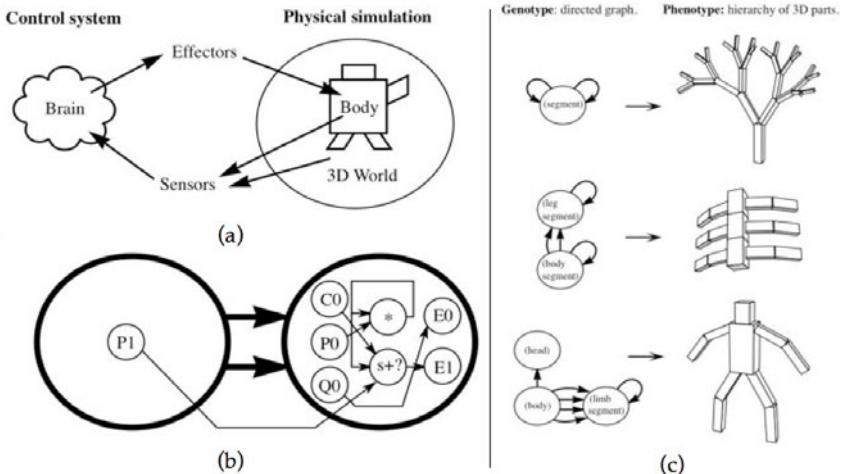
These simulators show that a set of quite simple rules could simulate complex interactions between virtual creatures. They opened the path to the simulation of complex virtual creatures simulations "in silico".

Karl Sims in 1994 presented a very impressive work on the use of artificial evolution to automatically generate both the behavior and the shape of artificial creatures immersed in a Newtonian 3D environment. With the help of a huge computing facility, Sims used artificial evolution to optimize his creatures to make them perform pre-defined tasks such as moving, swimming or jumping. He showed that such artificial evolution systems are able to come up with solutions which are visually very similar to those produced by natural evolution.

His work laid the foundations for several works dealing with such evolved 3D creatures. He proposed:

² At the start of this project, Ray believed that self-replicating programs could emerge automatically from random initialization of the initial memory.

- To use a 3D Newtonian dynamic simulator which allows computing the moves of the objects and creatures according to forces and torques.
- To use a modular definition of creatures by the use of a grammar which defines the basic blocks and their relationship.
- To define behavior³ by the use of a neural network embedded inside each block.



From Sims 94: Illustration of: (a) link between behavior and morphology of the creature. (b) Neural network inside a node. (c) Morphology development.

Fig. 6

Both parts are coded in a compact structure, which represents the “genetic code” of the creature. The evolution of this genotypic structure is then achieved thanks to an evolutionary algorithm, which first performs morphogenesis⁴ of the latter structure into a phenotype (phenotype = creature). This phenotype has to be evaluated according to the problem to solve.

Morphology definition

The morphological nature of most artificial creatures is defined using a formal grammar applied recursively to the rewriting of an initiator. This approach allows modularity, which is highly important for an efficient definition of complex creatures starting from compact coding.

³ In this application a behavior is the computation of the relative moves between linked blocks.

⁴ It is worth noting that such translation could include a growth stage in the case where the model includes some aspect of artificial embryogeny.

The behavior controller of such creatures is distributed over each node of the morphology graph. It is composed of a kind of neural network where each node is able to perform computation on signals incoming from its inputs. The latter are linked to the sensors of each block and outputs are linked to effectors able to set torques on the links between blocks, thus making them move. It is worth noting that several authors [28] use such controllers for the automatic adaptation of behavior to a pre-defined morphology.

The original complete phylogenetic approach assumes that morphogenesis of the genotype leads to complete creatures, which are not subject to changes according to their surrounding environment. In another work, Sims uses co-evolution to provide more adaptive creatures. In each experiment, two creatures have to compete for the domination⁵ of a cube. The fitness of each creature is computed by its performance against all of the others. This fits with the appearance of several strategies Miconi has also noted in [29].

Some years after this pioneering work, several researchers attempted expanding Sim's experiments in different ways using the same foundations.

Lipson, Hornby and Pollack in 2000 used this kind of evolutionary approach to break the gap between the virtual and real world by generating real robots. They evolve a population of virtual robots in a simulator using an evolutionary system, and then translate the best entity into a real robot. Their GOLEM project proposes a morphology composed of linked bars and actuators with an exogenous control system based on neural networks. These sticks can then move according to the use of motors or pistons. Virtual entities are selected according to their gait. In the quest to build more complex robots [30], they study with the Tinkerbot project another morphology definition, which is based on L-Systems. Behavior of such creatures is embedded in their morphology definition as motor commands, which represent the joints between several sticks.

Komosinski moves towards the simulation of a more complete 3D ecosystem in [31, 32] with his Framsticks project. His original aim is to “try to fill the gap between advanced artificial life models with their consequences and advanced simulation tools with their realism”. In this ecosystem, creatures can fight to gain energy and can reproduce either according to the use of an evolutionary system or to a chance encounter with a mate in the environment. He proposes to use sticks as building blocks and expands the set of sensors⁶ and actuators, which can be used to trigger the neural networks inputs. Moreover this work introduces a modeling language which can be used to define or to modify the creatures manually.

Several authors increase the complexity of the tasks the creatures have to perform. Chaumont [39] proposes to evolve catapults, Miconi [36] tries to enhance Sim's results through the co-evolution of fighting creatures, Shim's has worked on the co-evolution of the body and the brain for flying creatures [37]. In 2007, Mouret [33] followed the work of Lipson, Pollack and Hornby by trying to generate real flying robots automatically.

⁵ ‘Fitness’ is the distance between a creature and the cube at the end of the simulation.

⁶ Sensors are able to provide more complex information than those of Sim, for example, global orientation (gyroscopic), sensing of other creatures.

Following another path for the generation of artificial creatures, Bongard and Pfeifer proposed in 2003 [35] to use artificial ontogeny to evolve creatures starting from a single cell in a similar way to the one described in section 6. Differentiation occurs when a cell is duplicated thanks to a gene regulatory network. Each cell could embed one or more sensors, actuators and perform computation by the use of artificial neurons. The generated creatures are immersed in a 3D simulated Newtonian environment. They evolve to move towards a target.

In our team, Nicolas Lassabe proposed to use another kind of controller inspired by classifier systems [8]. Its aim was to provide a controller that can be more easily analyzed than neural nets while maintaining the proved performance of the latter. For that purpose he proposes a rule-based system where rules are triggered according to the sensing of each block. The triggering of a rule builds a piece of a mathematical function that will control an actuator. An efficiency factor is then provided for the rule according to the overall behavior of the creature in the environment. This efficiency factor will be used as a discriminator factor in case a set of rules could be triggered at the same time step. In the case of a constant sensing the generated signal will be inverted thus providing a kind of cyclic actuator. This effect is one important fundamental of all sensory motor systems in both natural and artificial systems.

Lassabe attempts several experiments which exhibit the ability of its novel controller to provide a wide set of solutions. He also proposes to study the emergence of symmetric and modular structures in the same class problem and proves that symmetry and modularity are of great help when the environment complexity increases.

The cited models are summarized in the following table (table 1), which emphasizes the differences between the models and the main experiments each author has performed.

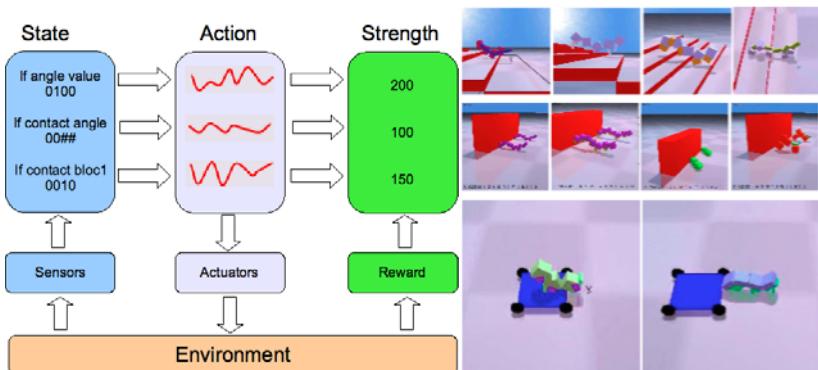


Fig. 7 From Lassabe 2007, Links between the rules, the mathematical and the environment. Creatures, which push a block in a cooperative way, climb stairs and practice skateboard.

Table 1 Characteristics of block-assembled mode

Model	Ref.	Key points	Morphology	Controller	Main experiments
Sims	[3] [27]	Pioneering work Simulation quality Co evolution	Oriented Graph and blocks	Neural network	Walking, Swimming, Jumping Fighting
Lipson, Hornby & Pollack (GOLEM)	[34]	Real building of evolved creatures as robots	Tubes and pistons	Centralized Neural network	Moving creatures
Bongard	[35] [72]	Artificial embryogeny Key work on cell specialization Interaction with an object	Set of linked cells	Neural network	Moving toward a target
Miconi	[36]	New kind of neural networks	Oriented Graph and blocks	Neural network	Fighting creatures
Shim	[37]	Expands the environment for the generation of flying creatures		Neural network	Flying creatures
Lipson, Hornby & Pollack (Tinkerbots)	[30] [40]	Real building as robots Mophology	L-System producing a structure composed of sticks and actuators	Direct encoding in the structure	Moving with a block
Komosinski	[31] [38]	New environment which includes several creatures.	Oriented graph and sticks	Neural network	Ecosystem Ski jumpers
Taylor	[16]	State of the art on Newtonian dynamic engines.	Oriented graph and Cylinders	Neural network	Real walking creatures Simulated swimming creatures.
Lassabe	[8]	More complex environments. Interaction with objects (blocks, skating)	Oriented Graph and blocks	Classifier system triggering control curves.	Climbing stairs Moving between blocks Pushing blocks Skating
Chaumont	[39]	Indirect fitness computation	Oriented graph and Cylinders	Neural network	Evolving catapults

5 From Phylogeny to Ontogeny

Phylogeny is the study of the developmental processes during species evolution, generation after generation whereas ontogeny focuses on the development of a single organism from egg to maturity. This section is dedicated to the bio-inspired evolution of some artificial creatures in an environment interaction standpoint.

Artificial metabolism in morphogenesis

In artificial phylogeny, most of the literature works rest on environmental adaptation. Nevertheless, an interesting approach has been conducted by Bornhofen [9, 44] these last years in an

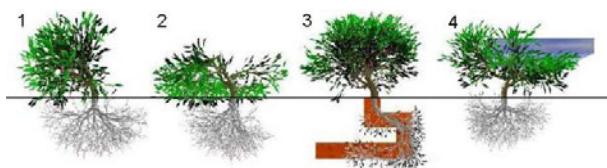


Fig. 8 Artificial plants evolved in various conditions (Bornhofen)

artificial plant growth simulation, by introducing metabolism. Biological metabolism is the whole process where chemical products are converted into energy and organic tissues. Commonly, artificial plant growth is modeled by L-systems [45], presenting a good description of evolution by recursive rules and providing a complete step-by-step developmental process description. By bringing in resource distribution, Bornhofen incorporates an artificial metabolism in his simulations. The 3D environment is divided in soil and sky and a simplified artificial ecosystem is built with minerals and sun light as resources. A genetic algorithm then determines plants evolution and a multi-agent platform is used to induce competition. Results prove the emergence of life-like traits “*without [any] intentional parametric bias*” [9] and show different virtual shapes very close to real plants (cf. figure 8), corresponding to natural evolution in various environments.

Machine metabolism

A new challenge arises from artificial morphogenesis: beyond environment adaptation, its reorganization and transformation. First, if numerous objects have been evolved by means of artificial ontogeny, “*few [...] have been transferred to reality*” [46] due to constructability problems. Rieffel and Pollack have then focused on “*how to build*” rather than “*what to build*” [46] and they evolved assembly plans, integrating virtual scaffolding, opening a way to bring artificial objects to reality. More recently, Hjelle and Lipson have developed a simulation with machine metabolism [47], evolving a robot able to manipulate truss structures (cf. figure 9). These structures could be seen as edges connected to joints in a 3D space. In this way a construction tree is built to simplify reconfiguration sequence with few manipulating operators. From a given truss, the goal is to optimize a given function, such as maximum height, length of a bridge, etc. A genetic algorithm is then run to find a short and correct sequence, leading to a full truss structure reconfiguration description.

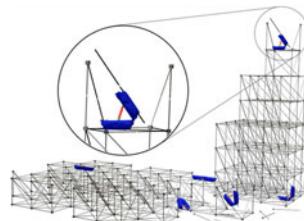


Fig. 9 Artificial truss
Hjelle & Lipson

Self-reconfigurable modular robots

According to Østergaard and al. [48] “*self-reconfigurable robots are robots that can change their shape in order to better suit their given task in their immediate environment.*” These last twenty years, several modular self-reconfigurable robots have been designed. Nearly each of them is composed of identical modules linked in a lattice-type (or chain-type) system [49]. Shape and motion are then co-evolved by an evolutionary algorithm. This leads to robustness, self-reparation capabilities and allows a good adaptation to any situation. A lot of functionalities can emerge from this approach. For example, in 2004, Mytilinaios et al. [50] implemented molecube robots self-reproduction. More recently, Zykov and al. [51, 66] have stepped across a new challenge by introducing various modules into their robots, such as grippers, wheels, battery, etc. to combine multipurpose functionalities. In our team, new works are currently developed by T. Breton based on artificial metabolism, where robots are able to change their shape and functionality by integrating elements in order to build structures.



Fig. 10 The modular robot *Molecube*

From virtual to real bio-inspired robots

In order to build real robots, artificial creatures are evolved in a 3D physical engine that allows tests to validate the conception. For both phylogeny and ontogeny approaches, an evolutionary algorithm is used. For example, results obtained by Hjelle and Lipson encourage them to develop a design of a real truss structure and a robot able to act as well [52], opening a way to a machine metabolic process in the near future. On the other hand, most of bio-inspired robots are modular and many applications come up with these creatures, such as exploration tasks, satellite recovery, conventional engineering and so on. Another stage of development is to reduce significantly the module size to fit nano-robotics or molecular nanotech, a fast-expanding field with numerous future applications, especially in medical science.

6 Artificial Embryogeny

In 2003, Stanley and Miikkulainen proposed a taxonomy for artificial embryogeny in [53]. They describe the artificial embryogeny as a “*methodological approach for reaching the level of complexity seen in natural organisms*”. This definition seems to be a good basis of artificial ontogeny. Since this paper, many works have been done to develop artificial organism starting based on digital cells as smaller units. The rest of this section presents the main existing approaches and models about artificial embryogeny.

The two main axis of this research field are cell differentiation and cell development (organism growth simulation). However both are linked in an organism growth, their mechanisms are very different. The end of the section presents these two approaches.

Cell differentiation

Cell differentiation mechanism simulation is one of the keys to artificial embryogenesis. In natural developmental process, cell differentiation designates the specialization of a cell. Starting from a unique undifferentiated cell named the zygote, it allows the creation of various cell types such as neurons, muscle cells, stem cells, etc.

To simulate this feature, Kauffman introduces Random Boolean Networks (RBN) [54]. A RBN is a network in which each node has a boolean state: activated or inactivated. These nodes are interconnected thanks to boolean functions represented by edges in the network. State $t+1$ depends on these boolean functions applied to nodes' values at time t . The interpretation of this regulatory network is simple: each node controls a gene and the node state represents the gene activity (activated or inhibited). The genome transcription produces the cell's final function.

In 1994, Dellaert proposed a developmental model based on this network [55]. In his model, he uses a boolean network as an artificial regulatory network in order to control cell differentiation. His model is based on a 2-D matrix that allows simple cell divisions: the initial organism is made of one cell only that covers up the entire grid; during the first division, the grid is split in two horizontally or vertically and the new cell can be differentiated. This differentiation is graphically observable by a modification of the cell colour. A genetic algorithm allows the regulatory network evolution in order to obtain a user-defined shape. Dellaert obtained several shapes. The behaviour of division and differentiation is adapted to the pattern that must be realised.

Many current developmental models use Artificial Regulatory Networks (ARN) to simulate cell differentiation. These systems are more or less inspired by gene regulation systems of living systems. In living systems, cell functions are described in the organism genome and are controlled by a regulatory network. Cells use external signals from their environment to activate or inhibit the transcription of some genome parts into mRNA (messenger RiboNucleic Acid), the future DNA (DeoxyriboNucleic Acid) model of the daughter cell. Cells can pick up external signals thanks to protein sensors positioned on the cell membrane. The gene expression in daughter cell will give its capacities.

Artificial regulation network models inspired by biology appeared about ten years ago. Torsten Reil was one of the first to present a biologically plausible model [56]. He defined his creature genome as a vector of numbers. The vector size is not previously fixed. Each gene starts with the particular sequence 0101, named "*promoter*". This kind of structure is present in living systems in which the TATA block plays the part of a promoter in real DNA. Reil uses a graph visualization to observe gene activations and inhibitions over time with randomly generated networks. Several patterns are obtained such as gene activation sequencing, chaotic expressions or cycle emergence for example. Reil also points out that after

random genome deteriorations, the system is able to rebuild the same pattern after an oscillation period.

Developmental models



Fig. 11 A turtle produced with de Garis' cellular automaton

states (dead or alive, 0 or 1) and where two simple rules using a Moore neighbourhood define the birth and the death of a cell. According to the initial environment configuration (repartition of living and dead cells at the beginning of the simulation), several shapes emerge and are able to move, reproduce, merge, etc.

Hugo de Garis added a genetic algorithm to evolve the transition rule set [58]. He was able to develop 2D shapes. With this aim, he did some modifications to Conway's model. The environment is initialised by only one living cell. Cells can reproduce themselves (in other words switch on a new cell in its neighbourhood) if they have at least one free neighbour. The creature is developed during a fixed number of iterations. According to this definition, only 14 states are available for a cell in a 2-D environment. The genetic algorithm fitness function evaluates the obtained shape according to the number of cells in the shape defined by the user and the number of cells out of this shape. He produces several configurations like simple shapes (triangles, squares, etc.) or more complex (letters, turtles, snowmen, etc.) as illustrated in figure 11.

The combination of all previously presented techniques creates many interesting developmental models. In a first stage, developmental models were based on a cellular automaton embedded with a gene regulatory network that modifies the cell colour like in A. Chavoya and Y. Duthen work [59]. This model is mainly used to solve the French flag problem where a French flag (3 colours: blue, white, red) must be developed starting from a single cell [60].

The regulatory network can also produce a function for the organism. Thanks to morphology modification, this kind of simulator can generate a growing creature with a "T" shape (see Eggenberger [11]). His model is also able to move in the

One of the first models of cellular development is the cellular automaton. The discovery of cellular automata is ascribed to John Von Neumann in the 1950s. A cellular automaton can be defined as a matrix of cells. They can be either dead or alive (0 or 1). A set of rules governs the transition between two simulation steps. Each rule gives the state (alive, dead, etc.) of each cell at time $t+1$ of the simulation according to the states at time t of the current cell and its neighbours.

Based on a cellular automaton, John H. Conway has defined the famous "*game of life*" described by M. Gardner in by [57] where cells can only have two

environment by modifying its morphology. It can also simulate the invagination mechanisms of vertebrate during the *gastrulation* stage⁷.

The cellular automaton was replaced by Knabe by a more complex system able to simulate the cell growth in a matrix of pixels [61]. Cell shapes are here more complex than the simple square produced with a cellular automaton. The environment strongly influences the cells morphologies and cells are in competition in order to grow and divide.

With a more realistic mass-spring model to simulate intercellular adhesion forces, R. Doursat has increased the precision in the positioning for each individual cell and thus for the entire organism [62]. Starting from a cell cluster in a 2-D environment, the author produces an organism in which several differentiated areas appear thanks to cell differentiations. Many shapes have been produced like the salamander presented in figure 12. In this latter, cell colours show their differentiation. Several differentiated areas are observable in this figure: legs, abdomen (divided up in six pieces) and the head (also divided up in ten pieces).

The addition of a physics engine to the developmental model can simulate Newtonian laws applied to each cell. With a 3-D extension of the simulation, many models develop various shapes [63, 10]. However, these organisms do not have any function. Their only goal is to produce a user-defined morphology. A gene regulatory network is often used to modify the cell colour.

In our team, we have integrated a simplified artificial chemistry coupled with different kinds of simulators that allows artificial organisms to grow in their environments using substrate to develop an adapted metabolism. This feature, often forgotten in developmental model, is one of the most important in Nature in order to provide energy to the cells. This chemistry also gives functions to the produced organisms: move a substrate, harvest the environment, etc. The function is also uncommon in a developmental model but necessary to develop high-level creatures able to interact with their environments. This kind of model was used to develop an “*artificial joint*”, composed of one “*kneecap*”, two “*bones*” and a set of “*muscular fibres*” [65] (figure 13). This joint is able to move thanks to a morphology modification of muscular fibre cells that lead to a rotation around the kneecap.

To summarise this overview of developmental models previously presented, the table 1 presents their main characteristics: simulation of the environment (simulators), genetic regulation and organism size and capacities. The chemical simulation is described with four levels: “realistic” if the simulation is highly detailed, “simplified” if the chemical simulation is less realistic, “diffusion” if a reaction-diffusion algorithm is used or “None” if the chemistry is not at all taken into account. The physics simulation column describes the simulation of mechanical interaction between cells in the environment. Genetic regulation expresses the

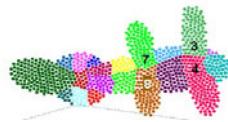


Fig. 12 Doursat's salamander developed thanks to a mass-spring model

⁷ This stage of the growth process allows the morphogens positioning and early-differentiated cell in the embryo in order to produce the four members and the head and tail of vertebrates.

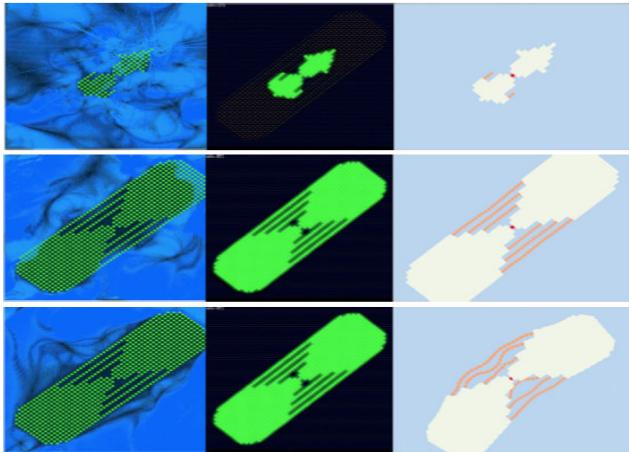


Fig. 13 From left to right, we can observe vertically the development of the same organism (bone & muscle) in three different simulators. The first one is a hydrodynamic simulator; the second one is in an artificial chemistry and on the right is the dynamic engine “Bullet”.

Table 2 Characteristics of developmental models

Author	Ref	Simulators		Genetic regulation	Organism Size	Capacities		
		Chemistry	Physics			Shape	Function	Metabolism
Rasmussen & al.	[22]	Realistic	None	None	~6 cells	No	No	Yes
Ono & al.	[21]	Realistic	None	None	~6 cells	No	No	Yes
Dellaert & al.	[55]	Diffusion	None	Yes	~30 cells	Yes	No	No
Reil	[56]	Diffusion	None	Yes	1 cell	No	No	No
Conway	[57]	None	None	None	~75 cells	Yes	Yes	No
De Garis	[58]	None	None	None	~50 cells	Yes	No	No
Chavoya & al.	[59]	Diffusion	None	Yes	~250 cells	Yes	No	No
Eggenberger Hotz	[11]	Diffusion	Yes	Yes	~75 cells	Yes	Yes	No
Knabe & al.	[61]	Diffusion	None	Yes	~100 cells	Yes	No	No
Doursat	[62]	None	Yes	Yes	~250 cells	Yes	No	No
Stewart & al.	[63]	Diffusion	Yes	Yes	~250 cells	Yes	No	No
Joachimczak & al.	[10]	Diffusion	Yes	Yes	<200 cells	Yes	No	No
Cussat-Blanc & al.	[65]	Simplified	Yes	Yes	~500 cells	Yes	Yes	Yes

implementation of a form of a gene regulatory network in the model. The organism size is the number of cells of organisms produced by the model. Finally, the capacities represent the properties of produced organisms: the shape corresponds to the capacity to produce a user-defined shape; the function to produce functional creature and the metabolism to produce organisms able to produce energy from environmental resources and to use this energy to perform their actions.

We can observe that the metabolism, so important in the natural developmental process, is often unimplemented. Moreover, this metabolism is strongly linked to the chemical simulation. The size of simulated organisms is very variable. It depends on the number of simulated details: the more details are simulated, the

smaller the organism is. Finally, the main aim of the common organism is to fill up a user-defined shape like the French flag problem. Few models presented in table 2 are able to accomplish a global function (in other words, modify the environment in a particular way). This last feature tends to be possible thanks to the regulatory networks: the colour shift produced by such a mechanism can give different actions to cells to act in the environment.

7 Conclusion

In this chapter we have tried to show that there is a very strong vital lead on the research on artificial creatures that can be labeled the Bio-Logic of the artificial creatures. This presentation is a general introduction to a lot of concepts and mechanisms in a large research field. Considering the three main properties of living organisms, the phylogenetic model is associated to creatures assembled from blocks, the ontogenetic model is associated to creatures developed from seeds and the epigenetic model is not particularly treated here. This latter is of course intrinsically linked to these two previous models, but the development of a nervous system (more precisely the physiology of the nervous system), which could have the capacity to memorize and to learn is another very complex subject.

In both phylogeny and ontogeny models, the principal underlying mechanism to produce automatically virtual creatures, but also models for modular robotics consists of evolutionary algorithms, which synthesize phenotypes able to develop themselves in an environment running different physic simulators. The more precise the simulators are, the more accurate the fitness function will be.

Once the artificial creatures have been elaborated in these artificial worlds, it can be a model to develop real systems with properties closed to living organisms. We have seen in section 5 that the notion of metabolism can be also applied to machines through self-reorganization models and transformations of the environment. This could deals with very innovative applications.

Of course there are a lot of difficulties to transfer such virtual creatures to reality due to constructability problems, but the approach exhibits a very strong potential for the design of a new generation of complex systems and machines in the future.

References

- [1] Magnenat-Thalmann, N., Thalmann, D.: The Direction of Synthetic Actors in the Film Rendez-vous à Montréal. *IEEE Computer Graphics and Applications* 7(12), 9–19 (1987)
- [2] Laird, J.E., Lent, M.V.: Human-level AI's killer application: Interactive computer games. In: Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence, pp. 1171–1178. AAAI Press / The MIT Press (2000)
- [3] Sims, K.: Evolving Virtual Creatures. In: SIGGRAPH 1994, pp. 15–22 (1994)

- [4] Fukunaga, A., Marks, J., Ngo, J.T.: Automatic control of physically realistic animated figures using evolutionary programming. In: Proc. of the Third Annual Conference on Evolutionary Programming, pp. 76–83 (1994)
- [5] Terzopoulos, D., Tu, X., Grzeszczuk, R.: Artificial fishes: Autonomous locomotion, perception, behavior, and learning in a simulated physical world. *Artificial Life* 1(4), 327–351 (1994)
- [6] Cliff, D., Miller, G.F.: Co-evolution of pursuit and evasion II: Simulation methods and results. In: Maes, P., Mataric, M.J., Meyer, J.-A., Pollack, J.B., Wilson, S.W. (eds.) *From animals to animats*, pp. 506–515, p. 20. MIT Press, Cambridge (1996)
- [7] Komosinski, M.: The world of framsticks: Simulation, evolution, interaction. In: Heudin, J.-C. (ed.) *VW 2000. LNCS (LNAI)*, vol. 1834, p. 214. Springer, Heidelberg (2000) (pp. 15, 16, 17, 42, 45, 64, 89)
- [8] Lassabe, N., Luga, H., Duthen, Y.: A new step for evolving creatures. In: IEEE-ALife 2007, pp. 243–251 (2007)
- [9] Bornhofen, S., Lattaud, C.: Life History of Virtual Plants: Trading Off Between Growth and Reproduction. In: Runarsson, T.P., Beyer, H.-G., Burke, E.K., Merelo-Guervós, J.J., Whitley, L.D., Yao, X. (eds.) *PPSN 2006. LNCS*, vol. 4193, pp. 808–817. Springer, Heidelberg (2006)
- [10] Joachimczak, M., Wrobel, B.: Evolution of the morphology and patterning of artificial embryos: scaling the tricolour problem to the third dimension. In: 10th European Conference on Artificial Life (ECAL 2009). Springer, Heidelberg (2009)
- [11] Eggenberger Hotz, P.: Combining developmental processes and their physics in an artificial evolutionary systems to evolve shapes. *On Growth, Forms and Computers*, 302 (2003)
- [12] Cussat-Blanc, S., Luga, H., Duthen, Y.: From Single Cell to Simple Creature Morphology and Metabolism. *Artificial Life*. ACM, Winchester (2008)
- [13] Banzhaf, W.: Artificial regulatory networks and genetic programming. *Genetic Programming Theory and Practice*, 43–62 (2003)
- [14] Pfeifer, R., Bongard, J.: How the body shapes the way we think. MIT Press, Cambridge (2006) ISBN:0262162393
- [15] Darwin, C.: The origin of species (1859)
- [16] Taylor, T., Massey, C.: Recent developments in the evolution of morphologies and controllers for physically simulated creatures. *Artificial Life* 7(1), 77–87 (2000)
- [17] Bongard, J., Lipson, H.: Automated robot function recovery after unanticipated failure or environmental change using a minimum of hardware trials. In: Procs. of the NASA/DoD Conference on Evolvable Hardware, Seattle, WA (2004)
- [18] Lipson, H.: Evolutionary robotics and open-ended design automation. In: Cohen, B. (ed.) *Biomimetics*. CRC Press, Boca Raton (2005)
- [19] Paul, C., Lipson, H., Valero Cuevas, F.J.: Evolutionary form finding of tensegrity structures. In: Proceedings of the 2005 Genetic and Evolutionary Computation Conference, Washington D.C., USA (June 2005)
- [20] Dittrich, P., Ziegler, J., Banzhaf, W.: Artificial chemistries - a review. *Artificial Life* 7(3), 225–275 (2001)
- [21] Ono, N., Ikegami, T.: Model of Self-Replicating Cell Capable of Self-Maintenance. In: Floreano, D., Mondada, F. (eds.) *ECAL 1999. LNCS*, vol. 1674, p. 399. Springer, Heidelberg (1999)
- [22] Rasmussen, S., Chen, L., Nilsson, M., Abe, S.: Bridging nonliving and living matter. *Artificial Life* 9(3), 269–316 (2003)

- [23] Turing, A.: The Chemical Basis of Morphogenesis. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences* 237(641), 37–72 (1952)
- [24] Cussat-Blanc, S., Pascalie, J., Luga, H., Duthen, Y.: Morphogen Positioning thanks to a hydrodynamic engine *Artificial Life XII* (2010)
- [25] Arthur, W.B.: On Designing Economic Agents that Behave Like Human Agents. *Journal of Evolutionary Economics* 3, 1–22 (1993)
- [26] Deneubourg, J.L., Goss, S., Franks, N., Sendova-Franks, A., Detrain, C., Chrétien, L.: The dynamics of collective sorting robot-like ants and ant-like robots. In: Meyer, J.-A., Wilson, S. (eds.) *From Animals to Animats*, pp. 356–365. MIT Press, Cambridge (1991)
- [27] Sims, K.: Evolving 3D Morphology and Behavior by Competition. In: *Artificial Life IV Proceedings*, pp. 28–39. MIT Press, Cambridge (1994)
- [28] Meyer, J.A., Husbands, P., Harvey, I.: Evolutionary robotics: A survey of applications and problems. In: Husbands, P., Meyer, J.-A. (eds.) *EvoROB/EvoRobot 1998*. LNCS, vol. 1468, pp. 1–21. Springer, Heidelberg (1998)
- [29] Miconi, T., Channon, A.: An Improved System for Artificial Creatures Evolution. In: Rocha, L.M., et al. (eds.) *Proceedings of the 10th conference on the simulation and synthesis of living systems (ALIFE X)*, Bloomington, Indiana, USA, MIT Press, Cambridge (2006)
- [30] Pollack, J.B., Lipson, H., Hornby, G., Funes, P.: Three generations of automatically designed robots. *Artificial Life* 7(3), 215–223 (Summer 2001)
- [31] Komosinski, M.: The Framsticks system versatile simulator of 3D agents and their evolution. *Kybernetes: The International Journal of Systems & Cybernetics* 32, 156–173 (2003)
- [32] Komosinski, M., Polak, J.: Evolving free-form stick ski jumpers and their neural control systems. In: *Proceedings of the National Conference on Evolutionary Computation and Global Optimization*, Poland, pp. 103–110 (2009)
- [33] de Margerie, E., Mouret, J.B., Doncieux, S., Meyer, J.A., Ravasi, T., Martinelli, P., Grand, C.: Flapping-wing flight in bird-sized UAVs for the ROBUR project: from an evolutionary optimization to a real flapping-wing mechanism. In: *7th European Micro Air Vehicle Conference (MAV 2007)*, Toulouse (2007)
- [34] Lipson, H., Pollack, J.B.: Automatic design and manufacture of artificial lifeforms. *Nature* 406, 974–978 (2000)
- [35] Bongard, J., Pfeifer, R.: Evolving complete agents using artificial ontogeny. In: *Morpho-functional machines: The new species (designing embodied intelligence)*, pp. 237–258 (2003)
- [36] Miconi, T.: Evosphere: evolutionary dynamics in a population of fighting virtual creatures. In: *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2008)*, Hong Kong (2008)
- [37] Shim, Y.-S., Kim, C.-H.: Generating flying creatures using body-brain co-evolution. In: *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, San Diego, California, July 26–27 (2003)
- [38] Komosinski, M., Ulatowski, S.: Framsticks: Towards a simulation of a nature-like world, creatures and evolution. In: Floreano, D., Mondada, F. (eds.) *ECAL 1999*. LNCS, vol. 1674, pp. 261–265. Springer, Heidelberg (1999)
- [39] Chaumont, N., Egli, R., Adami, C.: Evolving Virtual Creatures and Catapults. *Artificial Life* 13(2), 139–157 (Spring 2007)

- [40] Hornby, G.S., Lipson, H., Pollack, J.B.: Evolution of generative design systems for modular physical robots. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2001), vol. 4, pp. 4146–4151 (2001)
- [41] Bongard, J., Zykov, V., Lipson, H.: Resillient Machines Through Continuous Self Modeling. *Science* 314(5802), 1118–1121 (2006)
- [42] Zykov, V., Mytilinaios, E., Adams, B., Lipson, H.: Self-reproducing Machines. *Nature* 435(7038), 163–164 (2005)
- [43] Yim, M., Zhang, Y., Roufas, K., Duff, D., Eldershaw, C.: Connecting and disconnecting for chain self-reconfiguration with PolyBot. *IEEE/ASME Trans. Mechatron.*, 7, 4, 442–451 (2002)
- [44] Bornhofen, S., Lattaud, C.: Competition and evolution in virtual plant communities: a new modeling approach. *Natural Computing* 8(2), 349–385 (2009)
- [45] Prusinkiewicz, P., Lindenmayer, A.: The algorithmic beauty of plants. Lavoisier (1990)
- [46] Rieffel, J., Pollack, J.: Automated assembly as situated development: using artificial ontogenies to evolve buildable 3-d objects. In: Proceedings of the 2005 Conference on Genetic and Evolutionary Computation, pp. 99–106 (2005)
- [47] Lobo, D., Hjelle, D.A., Lipson, H.: Reconfiguration algorithms for robotically manipulatable structures. In: Proceedings of ASME/IFToMM International Conference on Reconfigurable Mechanisms And Robots, ReMAR 2009 (June 2009)
- [48] Østergaard, E.H., Kassow, K., Beck, R., Lund, H.H.: Design of the ATRON lattice-based self-reconfigurable robot. *Autonomous Robots* 21(2), 165–183 (2006)
- [49] Murata, S., Kurokawa, H.: Self-Reconfigurable Robot: Shape-Changing Cellular Robots Can Exceed Conventional Robot Flexibility. *IEEE Robotics & Automation Magazine* (March 2007)
- [50] Mytilinaios, E., Marcus, D., Desnoyer, M., Lipson, H.: Designed and Evolved Blueprints For Physical Self-Replicating Machines. In: Ninth International Conference on Artificial Life (ALIFE IX), pp. 15–20 (2004)
- [51] Zykov, V., Williams, P., Lassabe, N., Lipson, H.: Molecubes Extended: Diversifying Capabilities of Open-Source Modular Robotics. In: International Conference on Intelligent Robots and Systems, Self-reconfigurable Robots Workshop, IROS (2008)
- [52] Hjelle, D.A., Lipson, H.: A robotically reconfigurable truss. In: Proceedings of ASME/IFToMM International Conference on Reconfigurable Mechanisms and Robots, REMAR 2009 (June 2009)
- [53] Stanley, K., Miikkulainen, R.: A taxonomy for artificial embryogeny. *Artificial Life* 9(2), 93–130 (2003)
- [54] Kauffman, S.A.: Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology* 22, 437–467 (1969)
- [55] Dellaert, F., Beer, R.: Toward an evolvable model of development for autonomous agent synthesis. In: *Artificial Life IV*. MIT press, Cambridge (1994)
- [56] Reil, T.: Dynamics of gene expression in an artificial genome implications for biological and artificial ontogeny. *LNCS*, pp. 457–466 (1999)
- [57] Gardner, M.: The fantastic combinations of John Conway's new solitaire game life. *Scientific American* 223, 120–123 (1970)
- [58] de Garis, H.: Artificial embryology and cellular differentiation. In: Peter, E., Bentley, J. (eds.) *Evolutionary Design by Computers*, pp. 281–295 (1999)
- [59] Chavoya, A., Duthen, Y.: A cell pattern generation model based on an extended artificial regulatory network. *BioSystems* 94(1-2), 95–101 (2008)

- [60] Wolpert, L.: The French flag problem: A contribution to the discussion on pattern development and regulation. *Towards a theoretical biology* 1, 125–133 (1968)
- [61] Knabe, J., Schilstra, M., Nehaniv, C.: Evolution and morphogenesis of differentiated multicellular organisms: autonomously generated diffusion gradients for positional information. *Artificial Life XI* (2008)
- [62] Doursat, R.: Organically grown architectures: Creating decentralized, autonomous systems by embryomorphic engineering. *Organic Computing*, 167–200 (2008)
- [63] Stewart, F., Taylor, T., Konidaris, G.: Metamorph: Experimenting with genetic regulatory networks for artificial development. In: Capcarrère, M.S., Freitas, A.A., Bentley, P.J., Johnson, C.G., Timmis, J. (eds.) *ECAL 2005. LNCS (LNAI)*, vol. 3630, pp. 108–117. Springer, Heidelberg (2005)
- [64] Cussat-Blanc, S., Pascalie, J., Luga, H., Duthen, Y.: Three simulators for growing artificial creatures. In: *Proceedings of the IEEE Congress on Evolutionary Computation, IEEE CEC 2010* (2010)
- [65] Cussat-Blanc, S., Luga, H., Duthen, Y.: Cell2Organ: Self-Repairing Artificial Creatures thanks to a Healthy Metabolism. In: *IEEE Congress on Evolutionary Computation (CEC 2009)*, Trondheim (2009)
- [66] Zykov, V., Phelps, W., Lassabe, N., Lipson, H.: Molecubes Extended: Diversifying Capabilities of Open-Source Modular Robotics. In: *Self-Reconfigurable Robots Workshop, IROS 2008* (2008)
- [67] Mange, D.: Bio-Inspired Computing Machines. In: Mange, D., Tomassini, M, eds. (1998) ISBN 978-2-88074-371-0
- [68] Von Neumann, J.: The Theory of Self-reproducing Automata. In: Burks, A. (ed.). Univ. of Illinois Press, Urbana (1966)
- [69] Ray, T.S.: Population dynamics of digital organisms. In: Langton, C.G. (ed.) *Artificial Life II Video Proceedings*. Addison Wesley, Redwood City (1991)
- [70] Ray, T.S.: Evolution and optimization of digital organisms. In: Billingsley, K.R., et al. (eds.) *Scientific Excellence in Supercomputing: The IBM 1990 Contest Prize Papers*, Athens, GA, 30602, December 1991, pp. 489–531. The Baldwin Press, The University of Georgia (1991)
- [71] Ventrella, J.: Gene Pool: Exploring the Interaction Between Natural Selection and Sexual Selection. In: Adamatzky, A., Komosinski, M. (eds.) *Artificial Life Models in Software*. Springer, Heidelberg (2005)
- [72] Bongard, J., Paul, C.: Investigating morphological symmetry and locomotive efficiency using virtual embodied evolution. In: Meyer, J.-A., et al. (eds.) *From Animals to Animats: The Sixth International Conference on the Simulation of Adaptive Behaviour* (2000)

A Cutaway Visualization Technique for Understanding Scenes with Embedded Objects

Nancy Dandachy, Dimitri Plemenos, Safwan Chendeb, and Bachar El Hassan

Abstract. This paper is dedicated to virtual world an understanding technique that helps humans to understand a 3D scene. The paper presents a new technique that creates an alternative new mode of visualization in order to make it possible to have at the same time a total idea of the scene as well as a possibility of exploring objects which are inside, while saving time and memory. The algorithm is applied for any type of scenes modeled on a polygonal mesh. It extends the ray tracing algorithm by removing, from including objects, facets which hide inside ones.

Keywords: Computer Graphics, scene understanding, extracting knowledge, 3D visualization, Ray tracing.

1 Introduction

The quick change and development of materials in speed and memory with the advanced and optimized techniques of visualization, allows now the computer graphics area to reach a level of perfection enabling it to be used in various fields. As the existent visualization techniques were not sufficient alone to respond to all

Nancy Dandachy · Dimitri Plemenos
XLIM Laboratory UMR CNRS 6172,
University of Limoges, 83 Rue d'Isle,
87000 Limoges, France
e-mail: nancy.dandachy@gmail.com
plemenos@numericable.com

Safwan Chendeb
CITU Laboratory, University of Paris 8,
93526 Saint Denis, France
e-mail: safwan.chendeb@citu.fr

Bachar El Hassan
Faculty of Engineering, Lebanese University,
Rue AL Arz, Al Kobbeh, Tripoli, Liban
e-mail: bachar_elhassan@ul.edu.lb

objectives, many techniques for understanding 3D virtual world were invented in order to improve the result of visualization by computing a good camera position [1 and 2] or by doing a local or global animation of the camera around the scene [4,5,6 and 7] in order to extract knowledge about it.

However, these techniques are not able alone to answer all understanding problems of 3D complex scenes. There is still limits in some applications which prevent us from having a good comprehension of the scene. That's why some cases need be processed in a different manner. In such cases, augmented reality (AR) techniques may be used to make scene understanding easier. For example, an AR visualization technique was used in [8], where apparent contour lines are added to realistic images of scenes with reflections and transparencies, in order to permit the user to better distinguish what is real and what is illusion in the image.

Another case where special AR techniques have to be used, in order to make the user able to understand scene from an image, is the one of scenes with embedded objects. Sometimes, in newspapers, one can read aircraft models descriptions. Very often, journalists use special illustrations of their descriptions with pictures where it is possible to see the vehicle from outside and, at the same time, to see details inside the vehicle, thanks to holes designed on the picture (Fig.1). In this manner, the lectors can have a general idea of the hole vehicle.

All these illustrations are manually designed and this is a very difficult work. In volume rendering, it is very difficult to simultaneously interior and exterior structures while preserving clear shape cues. Some works are based on volume clipping techniques [9, 10], transfer functions [11], or illustrative techniques [12,13]. However these techniques suffer from the drawback of missing context information.

We think that such a work could be made automatically. That is why, we are going in this paper to discuss the problem of scenes with embedded objects and describe an alternative new mode of realistic visualization which extend the ray tracing algorithm in order to remove facets of the including objects which hide inside ones while saving time and memory.

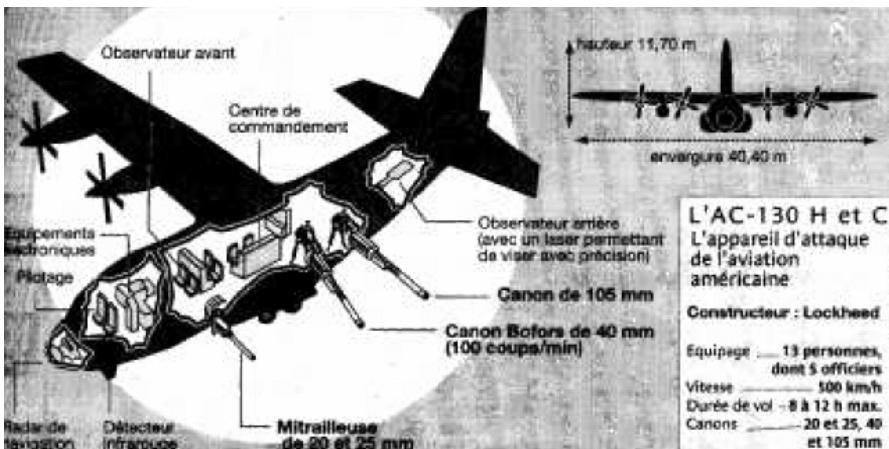


Fig. 1 Plane with Augmented Reality (AR)

The paper is organized in the following manner: In section **2**, a study of the existing techniques of visualization is presented. We will distinguish in this section, between the traditional and realistic visualization and we will talk briefly about existing ray tracing algorithms. In section **3**, an alternative novel methods of visualization is introduced. In section **4**, the first results obtained with the proposed method are presented and commented. In section **5**, a conclusion on the pertinence of our method will be made and possible future work will be considered.

2 Visualization Techniques

During the first years of the development of the computer graphics area, the effort was related primarily to visualization algorithms. Thus the majority of the techniques known and used currently have their origins in the sixties and seventies. However, Since the end of the seventies, practically no one new basic technique had seen the day and the efforts of the researchers were dedicated on the possibilities to improve the already existing techniques either by reducing the computing times, or by reducing the memory requirements, or still by inventing more sophisticated photometric models allowing a better quality of the obtained images. Schematically, there are two classes of visualization techniques.

2.1 Traditional Visualization

The first one is the class of traditional techniques which is divided in two modes of visualization:

- a. The wireframe mode where only all contours of objects or facets are displayed in the screen (see Fig. 2).

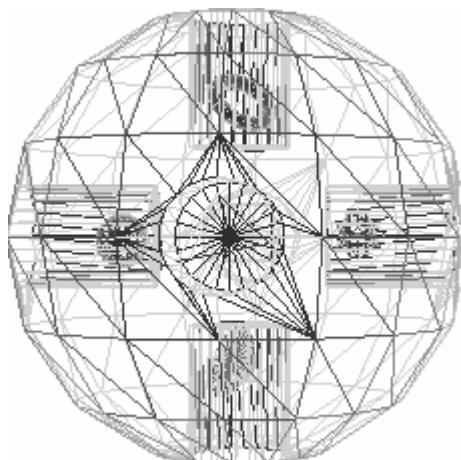


Fig. 2 Wireframe mode

- b. The full mode with an elimination of hidden parts where only objects or facets, considered to be closest to the position of a given point of view, are visualized (see Fig. 3).

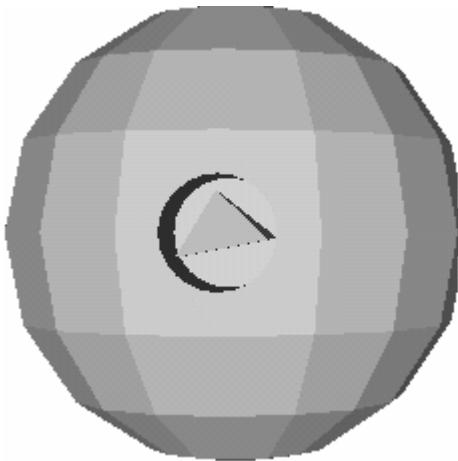


Fig. 3 Full mode with elimination of hidden surface

The problem of hidden parts removal was considered as one of the most difficult problem and has been the fundamental research problem in computer graphics up to the end of the seventies. The first algorithms were developed in the Sixties. The goal of these algorithms was to determine the lines, or surfaces which are visible for an observer located in a given point of the space. Sutherland and al. divided these algorithm into two classes [14]: Object space approaches and Image space approaches.

Among the algorithms the most known in object approaches are the algorithms of ROBERTS [15], the algorithms using BSP trees and the algorithm of WEILER-ATHERTON [16] using depth sorting.

Within the framework of the algorithms in image approaches, there exist three big families of algorithms of visualization with elimination of the hidden parts [17]:

- The Scan line algorithms.
- The Subdivision algorithm.
- The in-depth classification Algorithms or the algorithm with list of priorities (Z-BUFFER).

Unfortunately, the majority of algorithms for the elimination of hidden parts require sorting [14], which makes them generally slow, and needs often several minutes, even several hours of calculations.

2.2 Realistic Visualization

The second one is the realistic visualization. It visualizes objects on full mode with the elimination of hidden parts by adding light effects, shades, mirrors or transparencies, textures, reflexions and refractions, in order to deliver solutions very close to reality (see Fig.4). The algorithm of ray tracing and its varieties, the photon mapping and the radiosity are based on realistic visualization.

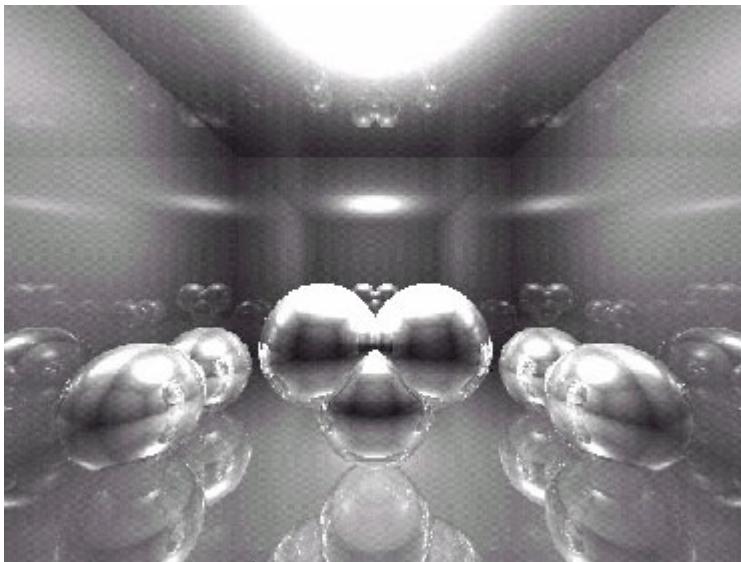


Fig. 4 Realistic visualization

As we are going to use the ray tracing algorithm, it is important to talk briefly about the related work in this field.

The algorithm of ray tracing, so popular today, is not a recent one. It was initially suggested by Appel [18] in 1968. Its first implementation goes back to 1971, in the software of three-dimensional visualization MAGI. It makes it possible to calculate the visibility of the objects at the same time as their illumination. It is able to manage the shades, the transparencies, the processing of textures and the interactions between objects. Moreover, it is adapted to any type of graphic primitives. This improvement has a cost of course: the computing times are much more important than for the algorithms seen previously.

The principle of the algorithm is the following:

- We consider a beam of imaginary rays connecting the eye of the observer to the center of each pixel (elementary square) of space image.
- For each ray, its intersections with all surfaces of the scene are calculated, in order to determine the point of intersection nearest to the observer.

- The luminous intensity of this point of intersection is assigned to the corresponding pixel.

Once the visible by the observer point is given, it is necessary to deal with the problem of shadows while a ray of the visible point is sent to the source of light. If this ray cuts a surface before the visible point, the visible point will then be considered in shade (see Fig.5).

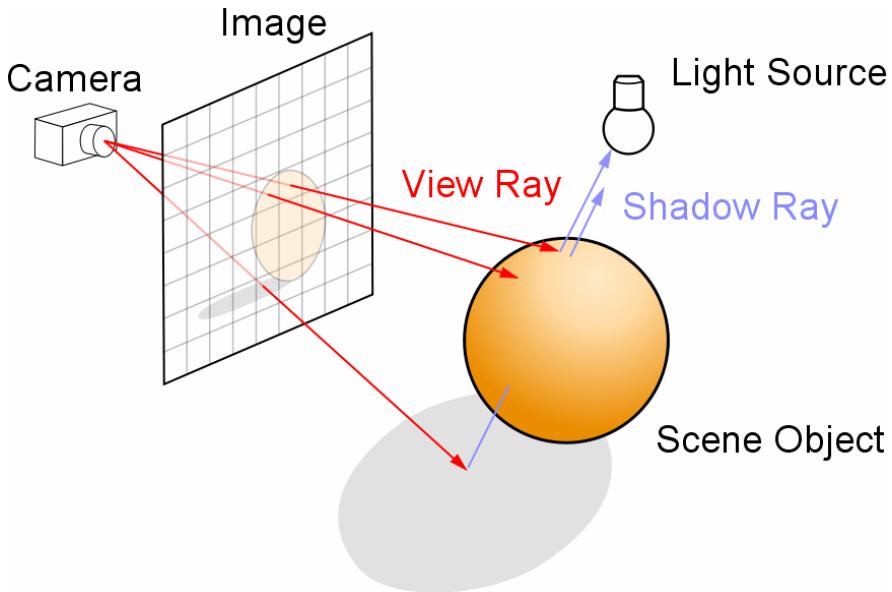


Fig. 5 The ray tracing algorithm

KAY and GREENBERG [19] proposed an extension of the ray tracing algorithm allowing the taking into account of the refraction of the ray when it crosses transparent surfaces. WHITTED [20] proposed the decomposition of the luminous intensity of a point in a specular component of reflexion S and a component of transmission T . This decomposition gives a binary tree that the algorithm must traverse for each ray sent (see Fig.6).

The follow-up of the lines of vision is carried out in the opposite direction of the light, from the eye towards the scene, which is called the “backward ray tracing”.

Other improvements were carried to the algorithm of ray tracing where the goal is to reduce computing time and the number or rays sent while preserving an acceptable quality of the obtained images.

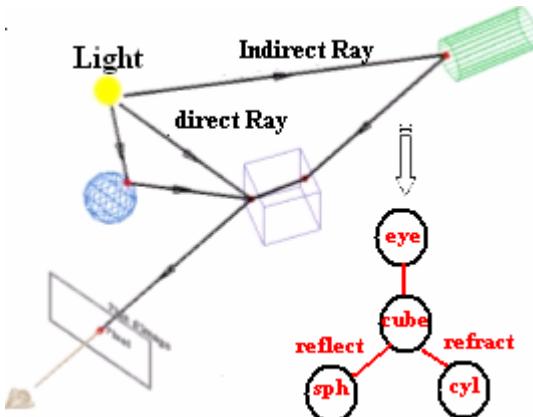


Fig. 6 Trajectory of rays in the ray tracing algorithm

In addition to these techniques there exist two rather important techniques: the selective refinement and the selective expansion [1]. Their goal is to obtain a significant reduction of the computing time, while preserving an acceptable quality of the obtained images.

3 Alternative Techniques of Visualization

Unfortunately, traditional and realistic visualization are not able alone to regulate the visualization problem of scenes with embedded objects. The elimination of hidden parts in both of them prevents us from seeing inside parts since they are always judged far from the eye position.

It might be sometimes sufficient, in order to see inside parts, to make including object transparent (see Fig.7), or to visualize including objects in Wireframe mode and inside ones in the full mode (see Fig.8).

However, it is not so realistic, and when it is about to see specific inside parts, both methods fail because they show all the inside of the including objects which means to show desired and undesired inside parts.

On the other hand, when the including object is described with a big number of facets, like in Fig.9, other problems occur. The sphere of Fig.9 is visualized on wireframe mode. However, its 327680 polygons make it appear like in full mode because of the big amount of polygons and it is difficult to distinguish inside parts.

But, the most important problem is that we want to see simultaneously a realistic image of both outside part of the scene and selected inside objects. That is why we thought about an alternative visualization technique which is more realistic and that creates different holes proportional to inside objects that we want to make visible.

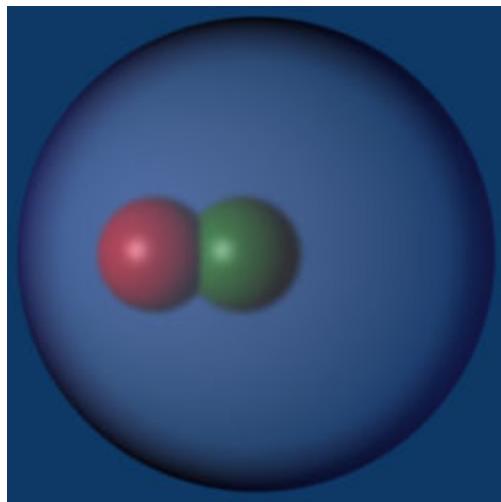


Fig. 7 Transparent including object

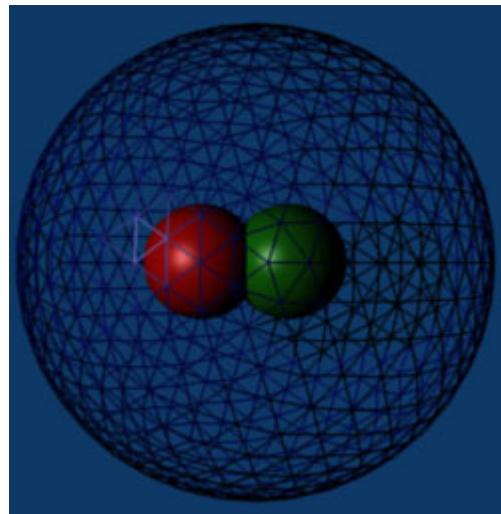


Fig. 8 Including object in Wireframe mode (1280 polygon) and inside ones on full mode

First of all, in order to be able to apply our method, it is necessary to know which are inside parts that we want to make visible. These parts can be described in any geometry model. They can be described by a polygonal mesh or by an analytical representation. However, our including objects of the scene have to be defined by a polygonal mesh.

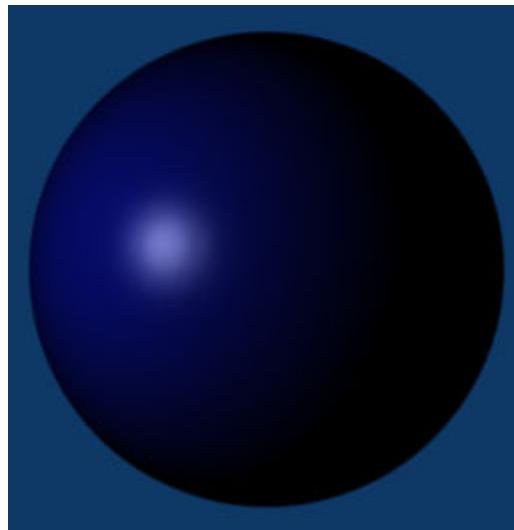


Fig. 9 Including object in wireframe mode with 327680 polygons

Our intervention lies in the part which consists on the determination of visible points or parts of the scene. Once a visible point is given, the calculation of its luminosity is made in the same way with the ordinary ray tracing algorithm.

The traditional ray tracing visualization algorithm resolves the visibility problem by:

- Searching for the nearest intersected object (or polygon).
- Calculating the nearest intersection point
- Assigning its luminosity to the corresponding pixel.

Rays R_1 and R_2 shown in Fig.10 are processed in same manner.

1. R_1 is the case of rays that cross both an including and an inside object
2. R_2 is the case of rays that cross an including without crossing any inside object

With this algorithm, R_1 and R_2 are both assigned to the closest intersected point of the including object. However, it is not effective since inside objects will never be visible as they are always occulted by including object facets. That is why we

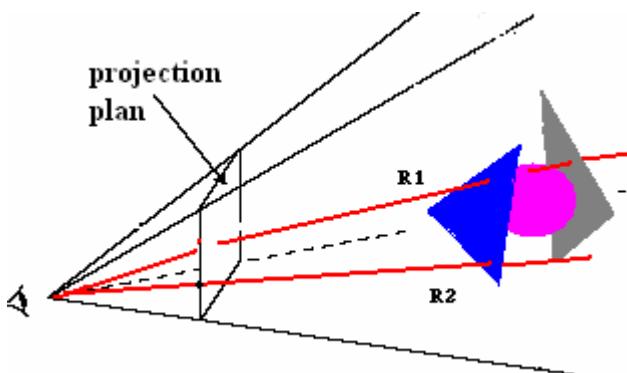


Fig. 10 Ray tracing method

propose to extend this algorithm by processing R_1 and R_2 in different manners, in order to remove including facets which hide inside parts that we want to make visible.

Thus, our method processes visibility as follows:

- R_1 ignores the intersected polygon of the including object and is associated to the closest intersected polygon of the inside object.
- R_2 is associated to the closest intersected polygon of the scene.
- Calculate in each case the nearest intersection point.
- Assign its luminosity to the corresponding pixel.

The most important task of the algorithm is to determine facets that have to be removed from the including mesh. Once we get the new mesh without facets that hide inside objects, we can apply the traditional ray tracing algorithm. In order to make R_1 ignore the including intersected facet, we associate to each facet of the including object, a selecting buffer initially set to zero that we call the sel-buffer.

Each time, we have the R_1 case, we update the sel-buffer of the including intersected facet. Finally, all facets having a sel-buffer set to 1 are removed from the mesh.

The algorithm describing this method is the following:

Eliminating facets algorithms**for** each facet in IncludeMesh **do**

facet.sel=0

end for**for** each pixel **do**

Create ray from the Eyepoint passing through this pixel

MinDist _ INFINITY

NearestObject _ NULL

for each IncFacet in IncludeMesh **do****if** ray intersect IncFacet **then****if** IntersectDist1<MinDist **then**

MinDist _ intersectDist1

for each IntFacet in insideMesh **do****if** ray intersect IntFacet **then****if** IntersectDist1<IntersectDist2 **then**

IncFacet.sel=1

end for**end for****end for****for** each facet in IncludeMesh **do****if** facet.sel=1 **then**

remove facet from IncludeMesh

end for

update IncludeMesh

Render Scene

However the test of all pixels of the image space is very time and memory consuming process because we are just interested on including polygons that are in the neighborhood of the inside object that we want to make visible. That is why, we propose to use instead of the complete projection plan, the sub spaces obtained by the projection of inside objects bounding box in sending rays.(see Fig.11).

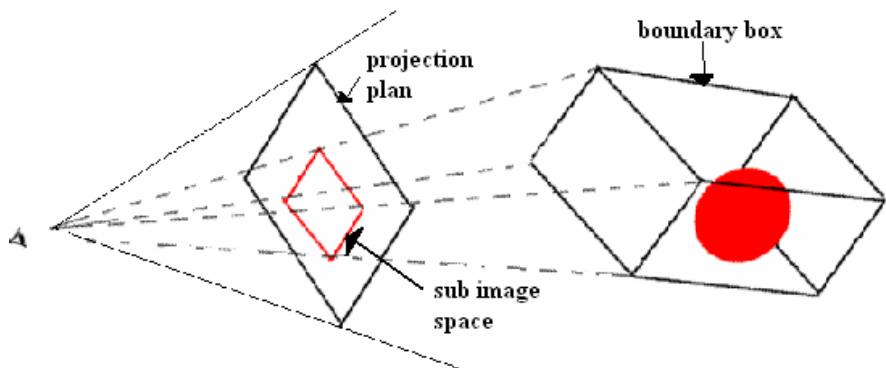


Fig. 11 Projection of the boundary box on the image space

There is a number of projection sub-spaces equal to the number of inside objects to make visible. Each projection space is defined by its right up RU corner. In order to obtain the RU corner coordinates in the world space, we use the intersection theorem of Thales which highlights proportionality relations in parallelism.

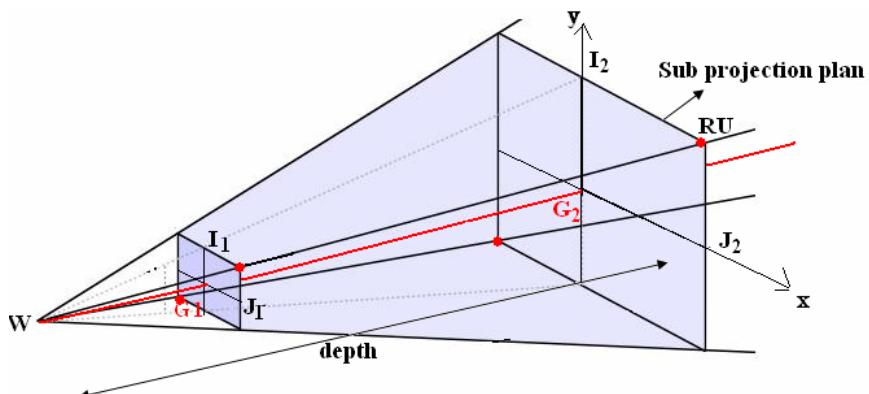


Fig. 12 Sub projection plan

In Fig.12, we suppose that the bounding box of the inside object and the projection plan are both parallel to the world space OXY. The viewpoint W is in the positive direction of the OZ axis. G₂ and G₁ are respectively the center of the back bounding box plan (rectangle) and its projection.

We have:

$$\frac{WG_1}{WG_2} = \frac{G_1 I_1}{G_2 I_2} = \frac{G_1 J_1}{G_2 J_2}$$

Let's suppose that (x , y) are the world coordinates of the RU corner of the sub projection plan and (X , Y) are the world coordinates of the RU corner of the back bounding box plan of an inside object.

So,

$$\frac{Z_{viewpoint} + Z_{box}}{depth} = \frac{x}{X} = \frac{y}{Y}$$

x and y are then determined by the following formulas:

$$x = X \times \frac{Z_{viewpoint} + Z_{box}}{depth}$$

$$y = Y \times \frac{Z_{viewpoint} + Z_{box}}{depth}$$

The depth is given by the following formula (see Fig.13):

$$depth = \frac{s}{2} \times \tan \frac{\theta}{2}$$

where:

- s is the height (size) of the image space
- θ is the fovy angle of the camera

Since we get x and y of RU, the coordinates of other corners are then obtained as following:

$$LD(-x, y), LD(-x, -y) \text{ and } RD(x, -y)$$

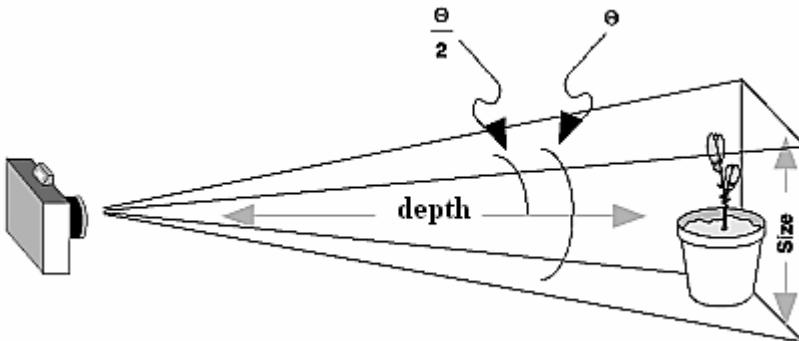


Fig. 13 Depth computation

If the bounding box of the inside object and the projection plan aren't parallel to the world space OXY, our x and y that we have just computed are in the local space related to the projection plan. In order to get the coordinates in the world space, we have to compute the product of x and y with the corresponding transformation matrix. The final algorithm with the sub projection plans optimization is described as following:

```

for each inside object do
  for each pixel of the sub image space do
    Create ray from the Eyepoint through this pixel
    MinDist ← INFINITY
    NearestObject ← NULL
    for each IncFacet in IncludeMesh do
      if ray intersect IncFacet then
        if IntersectDist1 < MinDist then
          MinDist ← intersectDist1
        for each IntFacet in insideMesh do
          if ray intersect IntFacet then
            if IntersectDist1 < IntersectDist2 then
              IncFacet.sel=1
      end for
    end for
  
```

```

end for
enf for

for each facet in IncludeMesh do
  if facet.sel=1 then
    remove facet from IncludeMesh
  end for
  update IncludeMesh
Render Scene

```

4 First Results

In this paper, we tried to resolve the problem of scenes with embedded objects. In this case of scenes, there are including and inside parts. In such cases, it is difficult to understand and explore with the existing techniques of visualization, especially if we want to extract knowledge or add information to some specific inside parts of the scene.

In order to enhance the visualization of this type of scenes, we propose an alternative technique of visualization that creates a hole around an inside object by removing facets of the including object which hide it.

The obtained results are encouraging since obtained images are in a good resolution (450 x 450 pixels) and allow concluding that it is possible to enhance the understanding and the extracting of knowledge of 3D complex scenes with embedded objects.

The method presented in section 3 has been implemented using the 3D graphics application of Blender with the Python scripts. The scenes can comprise one or more including objects as well as one or more inside objects. The method functions correctly in each case by removing hidden parts as it is necessary. In the case of more than one including object, the algorithm creates one including mesh that join all the including meshes of the including objects present in scene. When it is about more than one inside object, each one is processed separately by sending rays to its projected bounding box in order to save time and memory.

From Fig.14 to Fig. 18, all including and inside objects are described by a polygonal mesh of 1860 polygons. The images are rendered in the 450 x 450 pixels resolution, using the Blender Raytracer with the addition of the Subsurf modifier. The rendering in most of them, was taken without shadowing, because, according to the position of the view point, inside objects might sometimes appear very dark.

In order to optimize the calculation and avoid the transformation matrices, all scenes are centered on the O(0,0,0) position, the view point position is computed on the positive direction of the OZ axes and the projection plan is chosen parallel to the OXY plan. The following images (from Fig.14 to Fig.20) represent different scene cases:

The Scene 1 in Fig.14 represents the case of one including and one inside object.

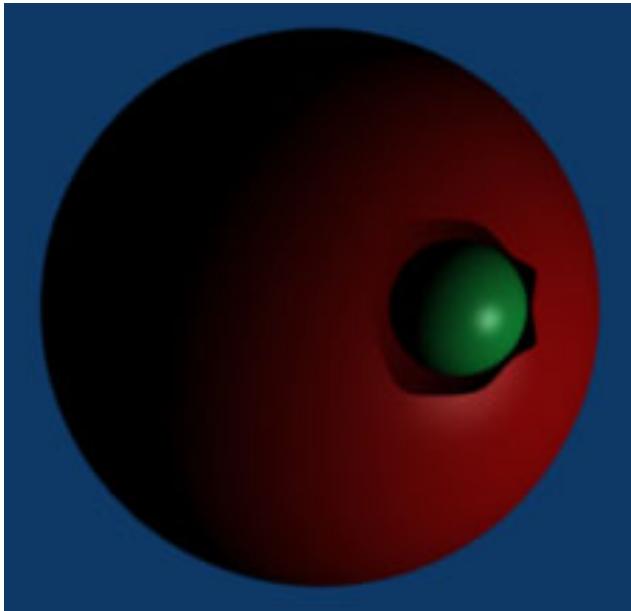


Fig. 14 Scene 1: One including and one inside object

The Scene 2 in Fig. 15 and the Scene 3 in Fig. 16 contain both one including object and two inside objects. However the two inside objects in the first case, are stuck together. In the second case, they are far one from the other. This represents two different cases that have to be processed differently. In the first case, the two inside objects are processed as one inside object because they have one common bounding box. The second case, as the two objects are far one from the other, each one has its own bounding box which is processed by the algorithm separately.

Scene 4 in Fig.17, contains two including and one inside objects while Scene 5 in Fig.18 contains two inside objects. In both cases, the including objects are joined in one including mesh and they are processed as one including object. When we have inside parts that we don't want to see, we join them in the rendering, with

including parts. In order to optimize the calculation, we don't consider them in the algorithm while eliminating facets.

In Scene 6, we choose to make the blue sphere as an undesired inside object and the green one as the inside part that we want to make visible.

The Fig.19 and Fig.20 show both the same scene after the elimination of hidden parts around the green sphere. The first one shows the including object in wire-frame mode in order to see all inside parts.

Fig.20 shows the same scene in full mode. The algorithm works correctly and eliminates facets which are around the blue sphere while ignoring the yellow one.

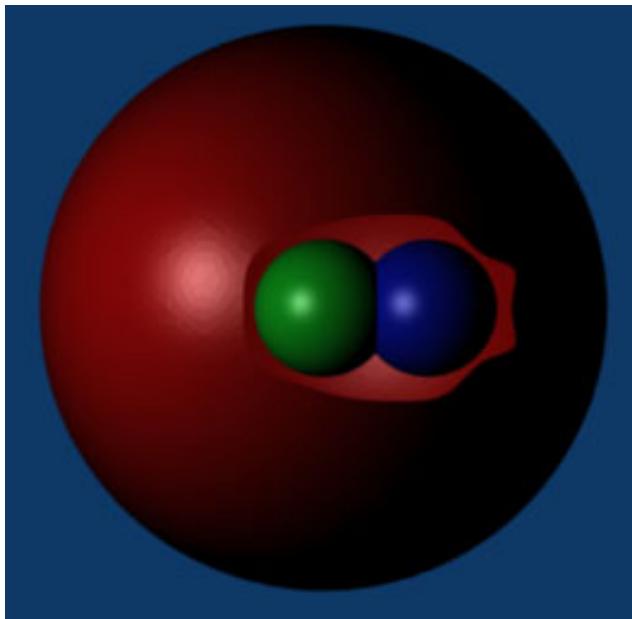


Fig. 15 Scene2: One including object and two inside objects

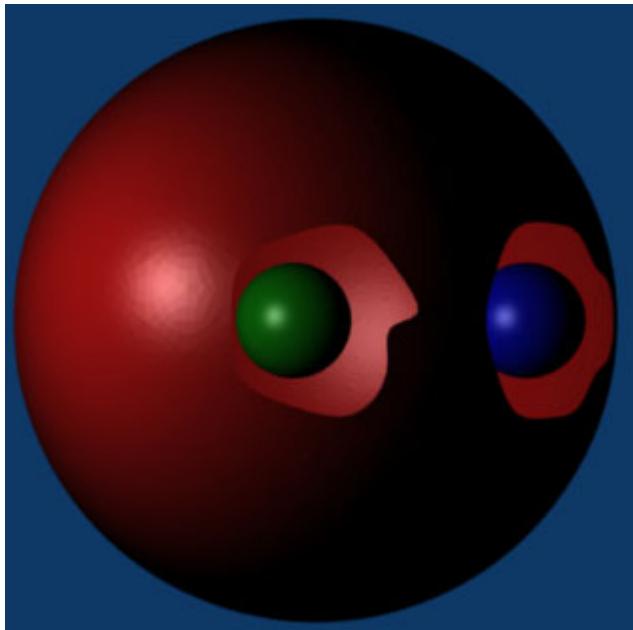


Fig. 16 Scene 3: One including object and two inside objects

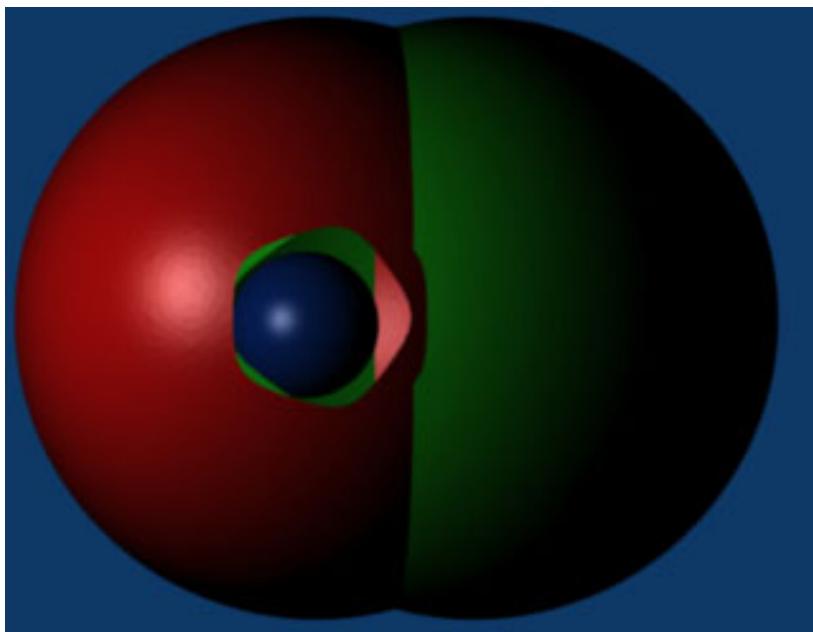


Fig. 17 Scene 4: two including objects and one inside object

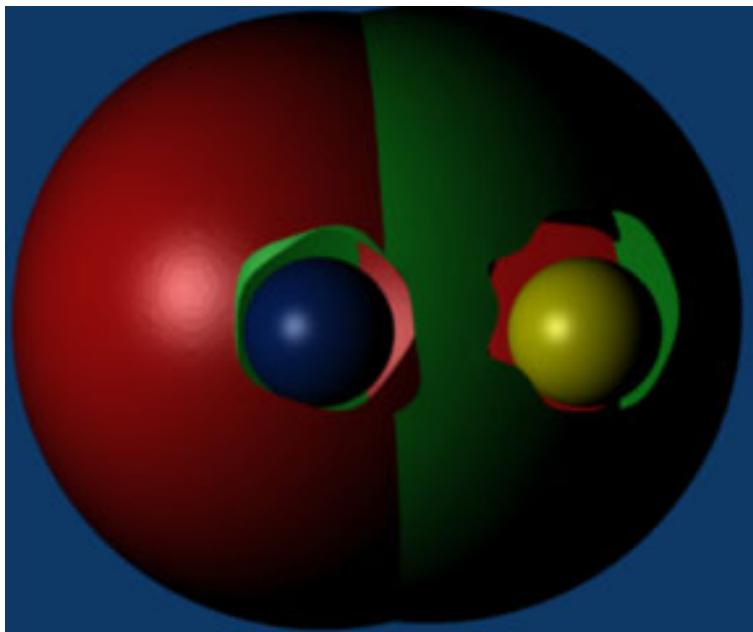


Fig. 18 Scene 5: two including objects and two inside objects

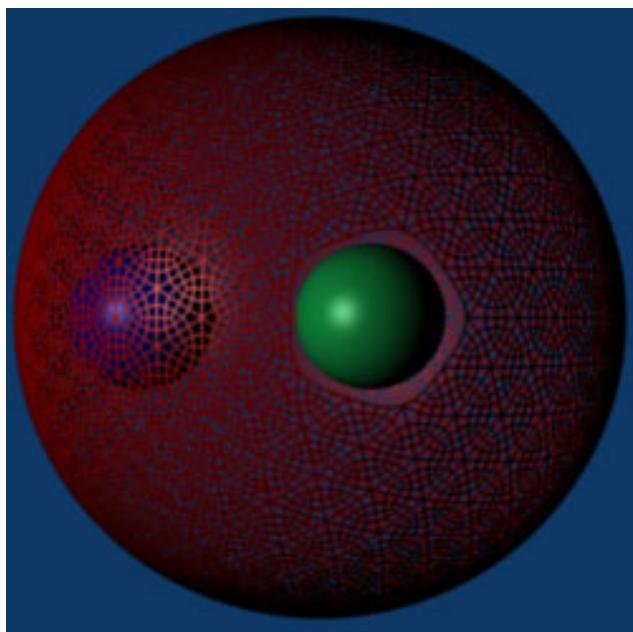


Fig. 19 Scene 6: Including object on wire frame mode

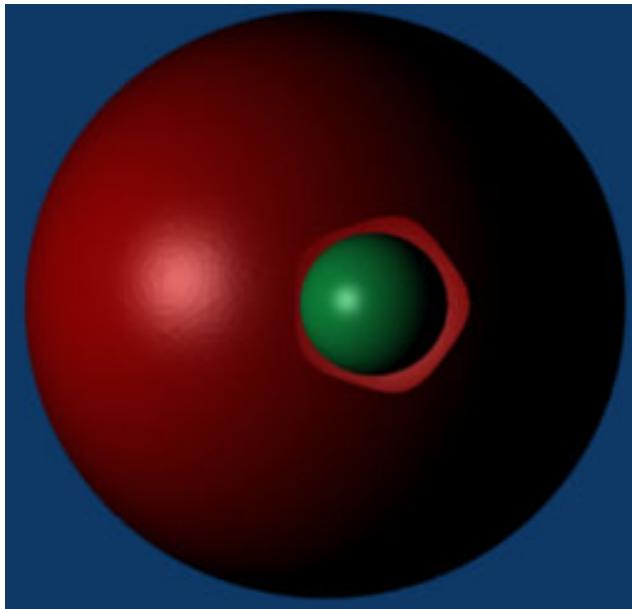


Fig. 20 Scene 6: Including object on full mode

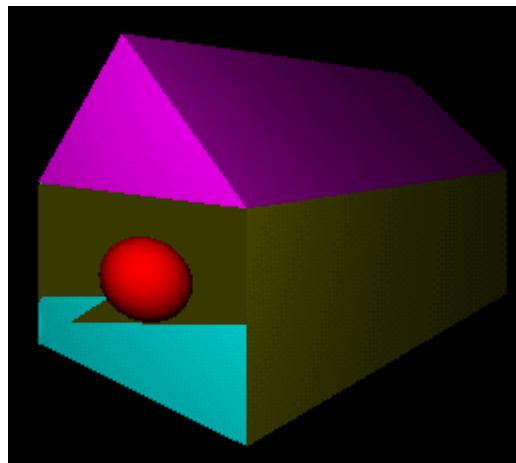


Fig. 21 The Scene House

5 Conclusion and Future Work

The method is efficient and gives good results. However, this work leaves several interesting problems. It gives sometimes a very big hole when the scene is modeled by a small number of big polygons (see Fig.21). It can be regulated by using one of the mesh subdivision or refinement algorithms which tend to subdivide facets into one or more sub facets.

Certainly, the considered methods are far from ideal and could be improved or one could imagine other possible solutions. Sometimes, the view point position is not suitable in order to see all inside parts that we want to make visible. That is why it would be interesting in the future, to think about a technique that calculates, before starting our algorithm, a good viewpoint position in order to have a better knowledge about the scene and a bigger amount of information.

References

1. Bouzefrane, R., Plemenos, D., Pueyo, X.: Les grande familles d'algorithmes de visualisation 3D Revue internationale de CFAO et d'Infographie, vol. 1(2) (1986)
2. Barral, P., Dorme, G., Plemenos, D.: Visual understanding of a scene by automatic movement of a camera. In: Graphi Con. 1999, Moscow, Russia, August 26 - September 3 (1999)
3. Sokolov, D., Plemenos, D.: Viewpoint quality and scene understanding. In: VAST 2005 Eurographics Symposium Proceedingss, Pisa, Italy, pp. 67–73 (2005)
4. Sokolov, D., Plemenos, D., Tamine, K.: Methods and data structures for virtual world exploration. *The Visual Computer* (2006)
5. Vasquez, P.P., Sbert, M.: Automatic indoor scene exploration. In: 3IA Computer graphics and Artificial intelligence Conference 2003, Limoges, France, pp. 13–24 (May 2003)
6. Jaubert, B., Plemenos, D., Grasset, G., Tamine, K.: Intelligent visibility-based 3D scene processing techniques for computer games. In: GraphiCon 2005, Novosibirsk, Russia (June 2005)
7. Jaubert, B., Plemenos, D., Tamine, K.: Techniques for off-line exploration using a virtual camera. In: International Conference 3IA 2006, Limoges, France, May 23–24 (2006)
8. Dandachy, N., Plemenos, D., El Hassan, B.: Scene understanding by apparent contour extraction. In: International Conference 3IA 2007, Greece (May 2007)
9. Eiskopf, D., Engel, K., Ertl, T.: Interactive clipping techniques for texture based volume visualization and volume shading. In: IEEE Visualization Conf. 2002, pp. 93–100 (2002)
10. Viola, I., Kanitsar, A., Gröller, M.E.: Importance driven feature enhancement, in volume visualization. *IEEE Trans. Visualization and Computer Graphics* 11(4), 408–418 (2005)
11. Lum, E.B., Ma, K.-L.: Lighting transfer functions using gradient aligned sampling. In: Proc. IEEE Visualization Conf. 2004, pp. 289–296 (2004)
12. Bruckner, S., Gröller, M.E.: VolumeShop: An interactive system for direct volume illustration. In: Proc. of IEEE Visualization 2005, pp. 671–678 (2005)

13. Bruckner, S., Gröller, M.E.: Exploded views for volume data. *IEEE Trans. on Visualization and Computer Graphics* 12(5), 1077–1084 (2006)
14. Sutherland, I.E., Sproull, R.F., Schumacker, R.A.: A characterization of ten hidden surface algorithms. *Computing Survey* 6, 1–55 (1974)
15. Roberts, L.G.: Machine perception of three dimensional solids. MIT Lincoln Lab. Rep., TR 315 (May 1963)
16. Weiler, K., Atherton, P.: Hidden surface removal using polygon area sorting. In: *Computer Graphics Proc. SIGGRAPH*, vol. 11, pp. 214–222 (1977)
17. Bouzefrane, R., Plemenos, D., Pueyo, X.: Les grandes familles d’algorithmes de visualisation 3D. *Revue internationale de CFAO et d’Infographie* 1(2) (1986)
18. Appel, A.: Some techniques for shading machine renderings of solids. In: *Proc. AFIPS Spring Joint Computer Conference* (1968)
19. Kay, D.S., Greenberg, D.: Transparency for computer synthesized images. *CACM* 4 (1979)
20. Whitted, T.: An improved illumination model for shaded display. *Communications of the ACM* 23(6), 343–349 (1980)

Surface Reconstruction with Smart Growing Cells

Hendrik Annuth and Christian-A. Bohn

Abstract. We propose *Growing Cells Meshing* (GCM) — a reconstruction algorithm which creates triangle meshes from clouds of arbitrary point samples registered on object surfaces. GCM is different to classical approaches in the way that it uses an artificial neural network together with an iterative learning technique to represent the triangle mesh. Based on the *Growing Cell Structures* (GCS) approach [3] we introduce the *Smart Growing Cells* (SGC) network as extension to fulfill the requirements of surface reconstruction. Our method profits from the well-known benefits entailed by neural networks, like autonomy, robustness, scalability, the ability of retrieving information from very complex data, and adaptability. On the downside, typical drawbacks like undesirable smoothing of information, inability to exactly model detailed, discontinuous data, or a vast amount of computing resources at big network sizes are overcome for the application of surface reconstruction. The GCM approach creates high-quality triangulations of billions of points in few minutes. It perfectly covers any amount and distribution of samples, holes, and inconsistent data. It discovers and represents edges, manages clusters of input sample points, and it is capable of dynamically adapting to incremental sample data.

1 Introduction

The demand for efficient high quality reconstruction algorithms has grown significantly in the last decade, since the usage of 3D point scans has widely been spread into new application areas. These include geometric modeling to supplement interactive creation of virtual scenes, registering landscapes for navigation devices, tracking of persons or objects in virtual reality applications, medicine, or reverse engineering.

3D points, retrieved by laser scanners or stereo cameras, introduce two vital questions (see Fig. 1 to get an idea of the problem). First, how can one recognize a

Hendrik Annuth · Christian-A. Bohn
Wedel University of Applied Sciences, Wedel, FRG
e-mail: annuth@fh-wedel.de, bohn@fh-wedel.de

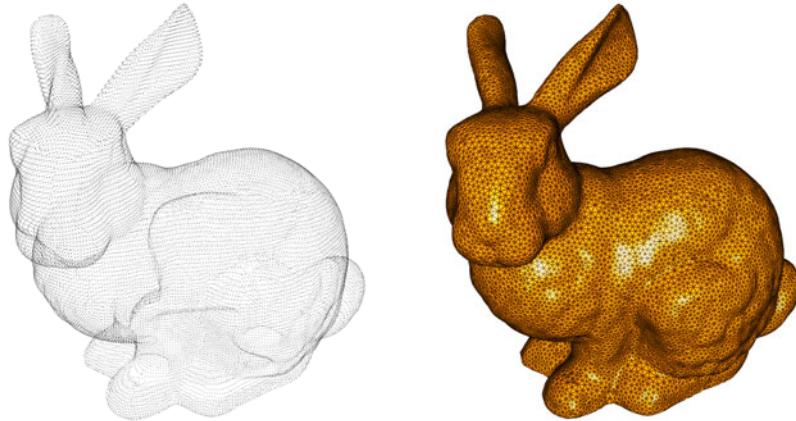


Fig. 1 Reconstruction algorithms cope with the task of recognizing a surface topology (see image on the right) from a set of arbitrary points (see image on the left) which are registered by a laser scanner or stereo photographs from real object surfaces.

topology of the originating 2D surfaces just from 0-dimensional 3D samples and without any other information from the sampled object? Second, for further processing, how is it possible to project this topological information on a data structure like a triangle mesh while meeting given constraints concerning mesh quality and size?

Although these issues have intensely been tackled since the early eighties a general concept that addresses all problems of surface reconstruction has not been determined up to now. Noise contained in the sample data, anisotropic point densities, holes, and discontinuities like edges, and finally, handling vast amounts of sampling data with adequate computing resources are still a big challenge.

Previous Work

The problem of surface reconstruction is a major field in computer graphics. There have been numerous approaches with different algorithmic concepts.

An important method is [8]. They construct an implicit surface using local information from an unorganized point cloud and then compile a mesh with the marching cubes approach. In contrast [6, 17] use radial basis functions as bases for a global implicit surface. A different method for surface reconstruction is proposed in [2, 15] where a delaunay tetrahedralization of a point cloud is successively reduced until the model is carved out. Further approaches, like [7], do not create a mesh but a piecewise smooth surface which then is converted into a mesh as described for example in [20]. Another class of algorithms suggest techniques based on the Bayes' theorem, like [18, 9]. These approaches are comparable with the GCS concept (see below) and they share some advantages like noise resistance.

In the area of artificial neural networks a famous work is [14] where the *Self Organizing Map* (SOM) is proposed which iteratively adapts its internal structure — a

2D mesh — to the distribution of a set of samples. It enables clustering or dimensionality reduction of the sample data. While a SOM has a fixed number of elements, the growing cell structures concept [3, 4] allows the network for dynamically fitting its size to the sample data complexity. SOM and GCS are likely to process and represent vector data like point samples on surfaces. [5] uses a SOM and [21, 23] use a GCS for the purpose of surface reconstruction. Further improvements in this field are proposed by [13] where constant Laplacian smoothing [19] of surfaces is introduced, and in [11] the curvature described by the input sample distribution is taken to control the mesh density. In [10] the GCS reconstruction process is further enhanced in order to account for more complex topologies. [12] use several meshes of the same model for a mesh optimization process, and [22] presents a concept for combining common deterministic approaches and the advantages of the GCS approach. In [1] the GCS approach is used for focusing on mesh optimization, and in [16] a surface optimization process based on an edge swap operation for the GCS approach is presented.

In the following, we describe the basis of our approach — the growing cell structures — and then derive our idea of the smart growing cells which matches the specific requirements of reconstruction.

2 Mesh Generation with Smart Growing Cells

GCM is based on using the internal structure of a smart growing cells network as triangulation of an object surface which is described by a set of surface sample points. The SGC is a neural network approach which is derived from the growing cells structures network.

The reason for using a neural network scheme for reconstruction tasks are its obvious advantages compared to deterministic approaches.

- They robustly handle arbitrary sample set sizes and distributions which is important in case of billions of unstructured points from scanned objects.
- They are capable of reducing noise and ply discontinuities in the input data.
- They are capable of adaption — it is not required to regard all points of the sample set on the whole. Further, incrementally retrieved samples can be used to retrain the network without starting the triangulation process from scratch.
- They guarantee to — theoretically — find the best solution possible. Approximation accuracy and mesh quality are automatically driven to the maximum.
- Training can be designed by simple learning rules which nevertheless solve complex problems. This is important for our application case, since the converged neural network structure and the general learning algorithm only match few requirements of a reconstruction task.

Nevertheless, these advantages partly clash with our needs. On the one hand, discontinuities are often desired (for example, in case of edges or very small structures on object surfaces). On the other hand, smoothing often destroys important aspects of the model under consideration (for example, if holes are patched, if separate parts

of the underlying objects melt into one object, or if the object has a very complex, detailed structure). In such cases, neural networks tend to generalize which may be advantageous from the physical point of view, but which mostly lets vanish visually important features which the human is quite sensitized to.

Growing cells meshing, from our point of view, is a breakthrough in this area. Through the use of a neural network, it delivers a general, robust, high-quality reconstruction algorithm which entails several advantages compared to classical approaches. Problems which often arise when using neural networks are solved for the application of surface reconstruction.

In the following, we outline the SGC algorithm. Then, our modifications to meet the requirements of surface triangulation are described.

2.1 Unsupervised Learning and Growing Cells

Unsupervised learning is accomplished by certain types of artificial neural networks which are able to organize its internal structure automatically depending on an arbitrary input sample distribution. After training, a set of *reference vectors* match the input sample distribution — classification is a typical task for this type of networks. In case of this work, reference vectors are interpreted as a set of vertices located on object surfaces.

Adaption of reference vectors in iterative unsupervised learning approaches is called *learning* or *training* and is generally accomplished by randomly presenting single n -dimensional samples from the input sample set to n -dimensional reference vectors and moving them in n -dimensional space (see Fig. 2).

Surface reconstruction with pure unsupervised learning would place a set of reference vectors on object surfaces, but does not determine information about the underlying surface topology. This leads to the Kohonen self organizing map described in the following.

```

Place  $k$  reference vectors  $\mathbf{c}_i \in \mathbb{R}^n$ ,  $i \in \{0..k - 1\}$  randomly in
the  $n$ D space of input samples

repeat

    Choose sample  $\mathbf{s}_j \in \mathbb{R}^n$  randomly from the input set
    Determine reference vector  $\mathbf{c}_b$  (the “best matching vector”
    or the “winning unit”) which lies closest to  $\mathbf{s}_j$ 
    Move  $\mathbf{c}_b$  in the direction of  $\mathbf{s}_j$  according to a certain strength
     $\varepsilon_{bm}$ , like  $\mathbf{c}_b^{\text{new}} = \mathbf{c}_b^{\text{old}}(1 - \varepsilon_{bm}) + \mathbf{s}_j \cdot \varepsilon_{bm}$ 
    Decrease  $\varepsilon_{bm}$ 

until  $\varepsilon_{bm} \leq$  a certain threshold  $\varepsilon_0$ 

```

Fig. 2 Training loop of general unsupervised training

Kohonen Self Organizing Map

The SOM is based on reference vectors which are randomly connected in a regular 2D mesh. The learning rule is extended to account for the direct neighborhood of a best matching unit (see Fig. 3)

```

for all  $c_{nb} \in$  neighborhood of  $c_b$  do
    Move  $c_{nb}$  in the direction of  $s_j$  according to a certain
    strength  $\varepsilon_{nb}$ , like  $c_{nb}^{\text{new}} = c_{nb}^{\text{old}}(1 - \varepsilon_{nb}) + s_j \cdot \varepsilon_{nb}$ 
    Decrease  $\varepsilon_{nb}$ 
end for
```

Fig. 3 The “neighborhood loop” of the Kohonen SOM to be added at the general unsupervised learning scheme

Insertion of this “neighborhood loop” into the general unsupervised learning algorithm (after moving of c_b in Fig. 2) leads to the phenomenon that the reference vertices are moved by accounting for the regular 2D mesh topology of the SOM. For example, training a plane-like arranged sample set leads to an adaption of the SOM grid to this implicit plane — the sample topology is recognized and finally represented by the SOM mesh.

Nevertheless, mesh size of a SOM is fixed and cannot adjust to the sample structure complexity. The growing cell structures approach overcomes this drawback.

Growing Cell Structures

To a certain degree, GCS may be seen as SOM which additionally is capable of growing and shrinking according to the problem under consideration defined by the sample distribution. This mechanism is based on a so called *resource term* contained in every reference vector, which is a simple counter. It counts the reference vector being a best matching unit and a high counter value signalizes the requirement of insertion of new reference vectors in that region. Generally, “resource” may be defined differently, for example by accumulating an approximation error.

But using a GCS for surface reconstruction still exposes vital problems. There is only one given type of topology available. In other words, starting training with a regular mesh can best approximate the topology of plane-like structures, and a tetrahedron would be adequate for sphere-like objects — only objects which are homeomorphic to the start object can be represented satisfactorily. Further problems arise at discontinuities like sharp edges and holes. These are commonly difficult to model through neural network type algorithms like mentioned above. This brings us to our approach, the smart growing cells.

2.2 Smart Growing Cells for Reconstruction

Smart Growing Cells overcome the most problems mentioned above by introducing the three basic mechanisms of (a) *aggressive cut out* (ACO), (b) *discontinuity vertices*, and (c) a specific curvature criterion, explained below.

The SGC basic structure is identical to general GCS. There are n -dimensional cells which we now term *neural vertices* connected by *links* through an m -dimensional topology. We let $n = 3$ since neural vertices are directly taken as vertices of the triangulation mesh and $m = 2$ since we aim at 2D surfaces to be reconstructed.

2.2.1 General Training

The simplified main training loop is very similar to the classical GCS approach (see section 2.1) and outlined in Fig. 4. For adaption of the neighboring vertices, a smoothing process like described in [13] and [19] is applied which replaces the classical movement, and which makes the adaption of the topology more robust.

As initial network, usually a tetrahedron or a plane with random vertices is suitable. A *vertex split* and an *edge collapse* operation adjust the network size dynamically. See Fig. 5 to get an idea of what happens in surface reconstruction with the SGC approach.

```

repeat
  for  $j = 1$  to  $k_{del}$  do
    for  $i = 1$  to  $k_{ins}$  do
      Choose a sample randomly, find closest neural vertex  $c_b$ 
      and move it with its neighbors towards the sample
      Increase signal counter of  $c_b$  (the resource term men-
      tioned above) and decrease the signal counters of all
      other vertices
    end for
    Find the best performing neural vertex — which is the one
    with the highest signal counter value — and add a new ver-
    tex at this position
  end for
  Find the worst performing neural vertices, delete them and af-
  fected edges
until a certain limit like approximation error, or number of vertices
is reached

```

Fig. 4 The main loop of the SGC approach is similar to that of the GCS

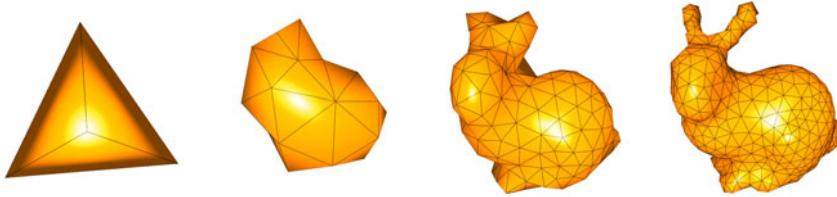


Fig. 5 Given a distribution of samples the SGC develops an initial simple polyeder into a final mesh which represents the underlying object topology

Vertex Split

A vertex split operation adds three edges, two faces and a new neural vertex. The longest edge at the neural vertex with the highest resource term is split and a new vertex is added in the middle. The signal counter value is equally spread between the two vertices (see Fig. 6).

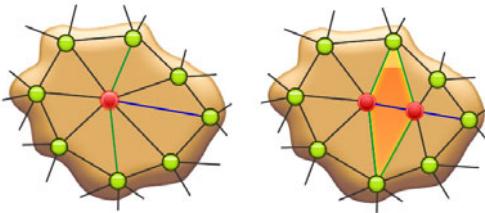


Fig. 6 Vertex split operation (from left to right) to increase mesh granularity locally. Edge collapse runs inversely.

Edge Collapse

Neural vertices with resource terms below a threshold r_{min} are removed together with three edges and two connected faces (see Fig. 6). Determination of the edge to be removed is driven by connectivity irregularities as proposed in [13].

2.2.2 Aggressive Cut Out

Aggressive Cut Out may be the most important new scheme we added to the standard GCS algorithm for achieving sufficient flexibility to match any topology of the training samples (the “homomorphism problem”). Before the edge collapse operation is applied to a vertex, it is tested if the vertex is contained in a *degenerated mesh region* (definition follows below). If so, the ACO process deletes it and additionally all connected faces.

It has been shown that degeneration of a part of the mesh serves as perfect indicator for a mesh topology which does not fit the underlying sample distribution correctly. For example, consider a region where sample density equals zero. Although vertices are not directly drawn into it by training adjustment, their neighbors may be moved there through their mesh connections. After a certain time, the resource term triggers the deletion of these vertices by edge collapse operations but

their links remain alive. These links mistakenly represent the existence of a topology and their structure is degenerated, i.e., it usually shows a surpassing number of edges with *acute-angled*¹ vertices (see Fig. 7).

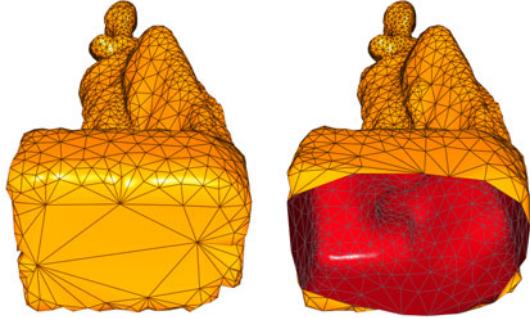


Fig. 7 The statue’s bottom on the left is not represented by samples. Its acute-angled triangles refer to a degenerated mesh region. On the right, ACO has detected this region and deleted the related mesh elements.

The term ”aggressive” is chosen since triggering properties are matched during training more often — suspicious neural vertices are cut out early.

Criterion for Degenerated Mesh Regions

In [10] a large area of a triangle is taken as sign for a degenerated mesh structure, but it has been shown that this criterion warns very late. Also, triangles generated from anisotropic sample densities are mistakenly interpreted as degenerated mesh regions. Our proposal is a combination of *vertex valence*², triangle quality, and quality of neighboring vertices. If all of the following conditions hold, an ACO is started at the vertex to be deleted.

1. Vertex valence rises above a certain threshold $n_{degvalence}$.
2. Vertex is connected to at least $n_{degacute}$ acute-angled triangles.
3. Vertex has more than n_{degnb} neighbors for which conditions (1) or (2) hold.

The latter condition says that ACO is only started if at least one or two neighbors show the same inconsistencies in their local mesh structure. This is reasonable since single degenerated vertices do not necessarily expose a problem, but may arise by accident.

Curing Boundaries after an ACO

Obviously after an aggressive extinction of a neural vertex and its surrounding faces, a boundary will be left behind which may consist of unfavorable mesh structure elements. Curing detects structures like a “spike”, a “nasty vertex”, a “needle eye”, or a “bridge” along a boundary and patches them.

¹ A triangle is termed acute-angled if the ratio of its area and the area which is spanned by a second equilateral triangle built from the longest edge of the first lies below a certain threshold ε_{acute} .

² The term vertex valence represents the number of connected vertices.

Spike. A boundary vertex with a valence of 2 (see Fig. 8) is termed spike. Such a vertex is very unlikely supporting a correct reconstruction process since it will be adjusted to an acute-angled triangle after few more iteration steps. A spike must be deleted in any case.

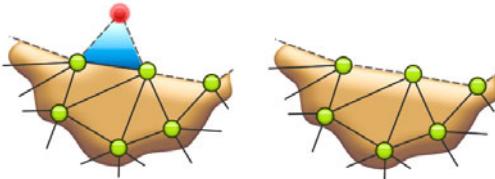


Fig. 8 Curing a boundary which contains a “spike” (see image on the left) results in the deletion of the involved vertex (see image on the right).

Nasty Vertex. A nasty vertex is a neural vertex with at least $n_{nastyacute}$ acute-angled triangles and/or triangles with a valence greater than $n_{nastyval}$ (see Fig. 9). These vertices are suspected to be part of a degenerated mesh region and are deleted.

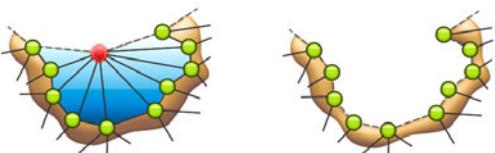


Fig. 9 A “nasty vertex” is a vertex with too many neighbors (see image on the left) and has to be deleted (image on the right).

Needle Eye. A needle eye is a neural vertex that is connected to at least two boundaries (see Fig. 10). At these locations the mesh does not have a valid mesh structure. To delete a needle eye, all groups of connected faces are determined. From these, the group with the most faces is kept and all others are deleted.

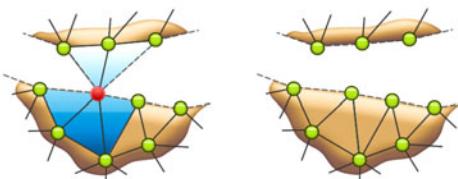
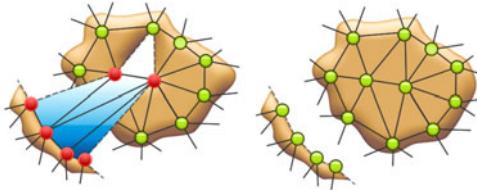


Fig. 10 A single vertex connected to two separate boundaries is termed a “needle eye” (see image on the left) and has to be deleted (see image on the right).

Bridge. A bridge is very likely to be part of a degenerated mesh region. If a mesh has a hole that consist of three vertices, then it would soon be closed by a coalescing process (see section 2.2.3). This is not allowed if exactly one of the edges of this hole is additionally connected to a face (which we term “bridge”, see Fig. 11) since an invalid edge with three faces would arise. The entire bridge structure is deleted and the hole will be closed with a new face.

Fig. 11 If an edge of a hole is connected to a second boundary and the hole is to be closed (termed a “bridge”, see image on the left), then the connection to the second boundary should be destroyed (see image on the right)



Multiple Boundary Search Through

After deletion of a neural vertex by the ACO process the curing mechanism will look for unfavorable structures along the boundary. There is more than one boundary to be considered, if the deletion destroys a coherent set of faces and multiple separate groups of faces arise due to other deletion processes.

Four cases may appear. First, the usual case with no additional boundaries. Second, when a needle eye is destroyed, the boundaries of all groups of connected faces need to be tested. Third, when surrounding faces of a vertex are interrupted by boundaries. And fourth, when a needle eye is connected to the surrounding faces of a vertex (see Fig. 12).

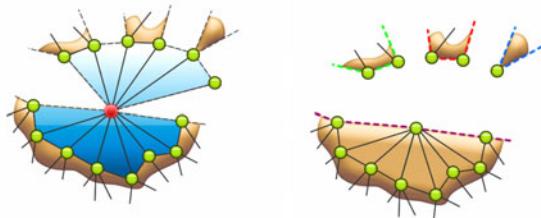


Fig. 12 Example for the cut out of a needle eye at a row of faces. Faces are not necessarily connected to other faces.

2.2.3 Coalescing

Like the mesh can be split by deletion of vertices or by the ACO process, it must also be possible to merge two mesh boundaries during the training process. For that, a coalescing test is accomplished each time a boundary vertex is moved.

Coalescing Test

This test recognizes if two boundaries are likely to be connected to one coherent area. For that, a sphere is created with the following parameters. Given the neighboring boundary vertices \mathbf{v}_1 and \mathbf{v}_2 of \mathbf{c}_b , then we define $\mathbf{c} = \mathbf{1}/2(\mathbf{v}_1 + \mathbf{v}_2)$. A *boundary normal* \mathbf{n}_c is calculated as the average of all vectors originating at \mathbf{c} and ending at neighbors of \mathbf{c}_b , where \mathbf{v}_1 and \mathbf{v}_2 are not taken into account. The boundary normal can be seen as a direction pointing to the opposite side of the boundary. We define a sphere with the center at $\mathbf{c} + \mathbf{n}_c r$ with a radius r as the average length of the edges at \mathbf{c}_b .

The coalescing condition at two boundaries hold, i.e., merging of the boundaries containing \mathbf{c}_b and \mathbf{q} on the opposite side happens, if \mathbf{q} is contained in the defined sphere, and the dot product of the boundary normals at \mathbf{c}_b and \mathbf{q} is negative.

Coalescing Process

After detecting the neural vertex \mathbf{q} to be connected with \mathbf{c}_b , the according faces must be created, starting with one edge from \mathbf{c}_b to \mathbf{q} . There are two cases — “corner” and “long side” — which have to be considered.

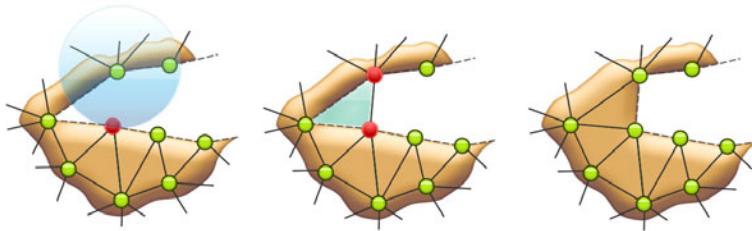


Fig. 13 Coalescing process at a mesh corner. On the left, the search process of a coalescing candidate. In the middle, one edge is created, on the right, the only face capable of being added is the corner face.

Corner. A corner of the same boundary arises when \mathbf{c}_b and \mathbf{q} have one neighboring vertex in common. Here, it is only possible to generate the face which closes the corner (see Fig. 13). Otherwise a needle eye would be created which is not a satisfactory mesh structure.

Long Side. Here, two boundaries appear to be separated. After determining the new edge, there are four possibilities for insertion of a new face containing the edge (see Fig. 14b). The triangle with edge lengths which vary fewest is taken in our approach (see Fig. 14c) since it is the triangle with the best features concerning triangle quality. Finally, to avoid a needle eye, a further triangle must be added — again, we take, the face with the highest edge similarity (see Fig. 14d). Coalescing

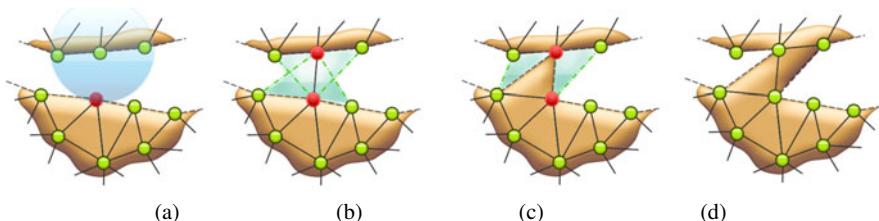


Fig. 14 Coalescing of two separate boundaries. In (a) and (b) the edge is determined, in (c) the triangle with the smallest variance of edge lengths is added, in (d) another triangle is created to avoid a needle eye.

sometimes creates unfavorable bridges that need to be deleted through a further ACO process.

ACO is a vital scheme which allows for changing the topology of the initial SGC mesh to match the topology of the underlying object. If ACO sees inconsistencies in the mesh it clears them by destroying parts of the mesh at the involved locations. Coalescing is the counterpart of this deletion. It forms new connections to create new topological structures (see Fig. 15).

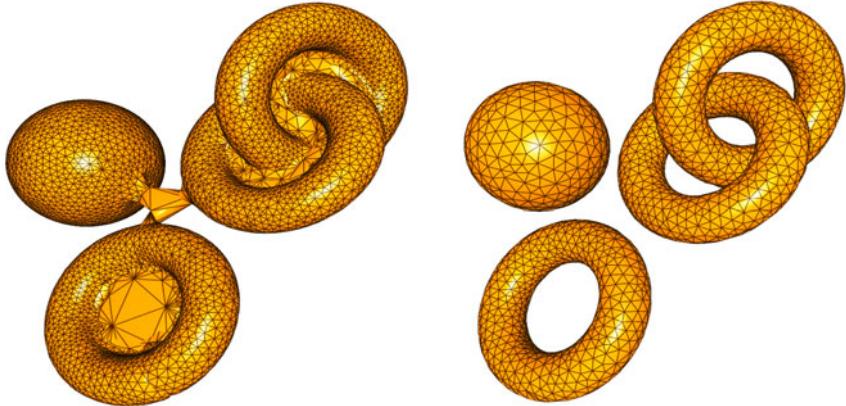


Fig. 15 Aggressive cut out detects inconsistent regions in the SGC mesh (visible in the image on the left side at the centers of the tori), deletes them and melts the remaining boundaries (see image on the right side). Thus, the mesh topology is adapted during training.

2.2.4 Granularity Adaption

Up to now, the SGC is able to approximate an arbitrary sample set by a 2D mesh. Now we present an efficient local adaption scheme of the mesh density in a way that areas with a strong curvature are modeled by a finer mesh resolution (see Fig. 16). This also relieves the influence of the sample density on the mesh granularity making the SGC less vulnerable to sampling artefacts like holes or regions which are not represented by a uniform sample distribution.

Each time a vertex is adapted by a new sample we calculate the estimated normal \mathbf{n}_k at a neural vertex \mathbf{v}_k by the average of the normals at the surrounding faces. The curvature $c_k \in \mathbb{R}$ at a vertex is determined like

$$c_k = 1 - \frac{1}{|\mathcal{N}_k|} \sum_{\forall \mathbf{n} \in \mathcal{N}_k} \mathbf{n}_k \cdot \mathbf{n} \quad (1)$$

with the set \mathcal{N}_k containing the normals of the neighboring neural vertices of \mathbf{v}_k . Each time a neural vertex is chosen as winner, its curvature value is calculated and a

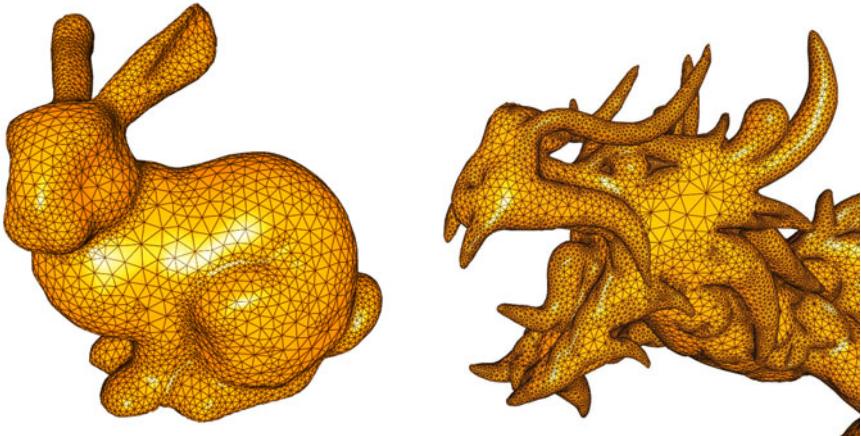


Fig. 16 Granularity adaption correlates surface curvature to mesh density. See for example, the accentuated mesh density at the bunny’s leg compared to its torso (left side), or the dragon’s horns compared to its cheek (right side)

global curvature value \bar{c} is adjusted. Finally, the curvature dependent resource term r_k at \mathbf{v}_k is adjusted through $r_k^{new} = r_k^{old} + \Delta r_k$, with

$$\Delta r_k = \begin{cases} 1, & \text{if } (c_k < \bar{c} + \sigma_{r_k}) \\ [c_k / (\bar{c} + \sigma_{r_k})] (1 - r_{min}) + r_{min} & \text{else} \end{cases}$$

and the deviation σ_{r_k} of the resource term r_k , and a constant resource r_{min} that guarantees that the mesh does not completely vanish at plane regions with a very low curvature.

2.2.5 Discontinuity Modeling

A sampled model that exposes discontinuities like edges is difficult to be approximated by a neural network mesh. Discontinuities are smoothed out since the network tries to create a surface over them. This might be acceptable in many application areas since the approximation error is fairly small, but the effect is unfavorable in computer graphics since it is clearly visible. And even worse: edges are quite common in real world scenarios.

Therefore, we propose handling vertices at regions classified as discontinuities of the underlying scanned object differently — first, they are only allowed to move in the direction of an object edge to represent it more properly, and second, the smoothing process from section 2.2.1 is not applied at them (see Fig. 17, the same model with and without discontinuity modeling).

Recognizing these *discontinuity vertices* is accomplished as follows. We determine the curvature values of those neighbors which have a distance of two

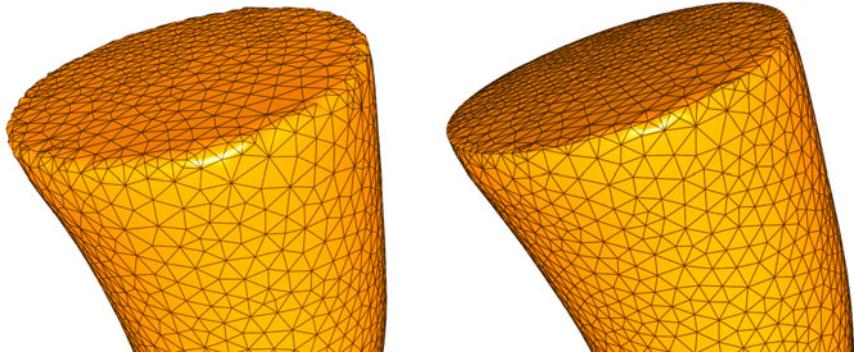


Fig. 17 Discontinuity modeling lets vertices move mainly in the direction of discontinuities. On the left, a result from the SGC approach without discontinuity modeling is exposed, on the right, with discontinuity modeling cells have grown into the corner of the plate.

connections from the vertex v under consideration — the “second ring” of neighbors. Then the average δ_{ring} of the squared differences of consecutive curvature values on the ring is calculated.

If a curvature value clearly deviates from the average curvature value, then a discontinuity at this location is assumed if the average of the neighbors’ (second ring) curvature gradient differs to a certain amount, i.e., if

$$(c_k > 2\sigma_{c_k}) \wedge (\forall c \in \mathcal{C}_k : \delta_{ring} > 4\sigma_{c_k}^2)$$

with \mathcal{C}_k the set of curvature values of the second ring of neighbors.

For approximating the edge normal at a discontinuity vertex v , we take the average of the normals of two of the neighboring vertices, either those with the highest curvature value, or those which are already detected as being a discontinuity vertex. Finally, the normal is mirrored if the edge angle lies above 180° , which is indicated by the average of the surrounding vertex normals; in the first case it points in the direction of v .

Fig. 18 shows two further examples for the discontinuity modeling facility of GCM.

3 Results

In Fig. 19 the whole GCM algorithm is listed as pseudocode. To keep it comprehensive, the outermost loop of the algorithm is neglected, vertex split and edge collapse operations are triggered by using counters (compared with the basic algorithm from section 2.2.1).

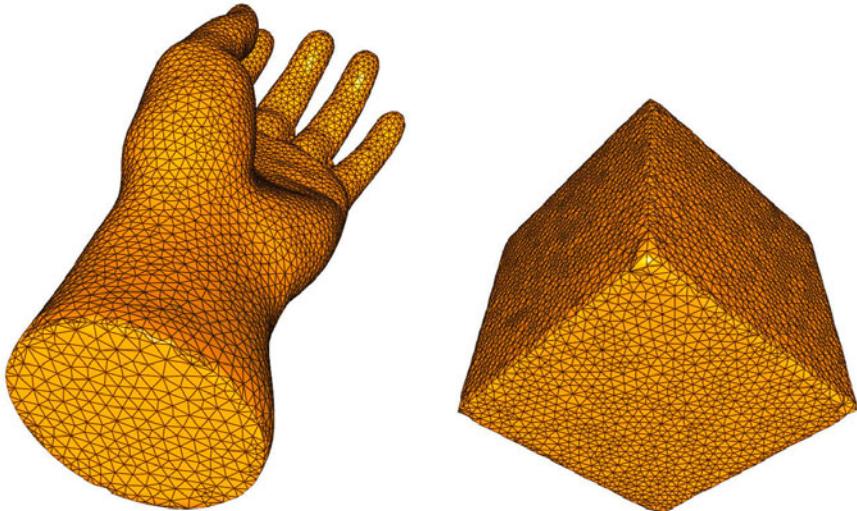


Fig. 18 Two example objects which expose the discontinuity modeling facility of the SGC

Parameters

The following parameters have been proven to be reliable for almost all sample sets we took for reconstruction: $\varepsilon_{bm} = 0.1$, $\varepsilon_{nb} = 0.08$, $r_{min} = 0.3$, $\varepsilon_{acute} = 0.5$, $n_{degacute} = 4$, $k_{ins} = 100$, $k_{del} = 5$, $n_{degnb} = 1$, $n_{nastyacute} = 4$, $n_{nastyval} = 3$.

Test Hardware

We used a *Dell® Precision M6400* Notebook with *Intel® Core 2 Extreme Quad Core QX9300* (2.53GHz, 1066MHz, 12MB) processor with 8MB 1066 MHz DDR3 Dual Channel RAM. The algorithm is not parallelized.

Some visual results are exposed in Fig. 21. All pictures are drawn from one SGC mesh and they are rendered using Phong Shading. Most models stem from the *Stanford 3D Scanning Repository*.

Besides visual results, GCM comes up with impressive numbers, listed in Fig. 20. It can be seen that mesh quality, i.e. the percentage of perfect triangles in the mesh lies at 96% at average. This is an outstanding result, nevertheless this is typically expected when using an approach from the field of unsupervised learning, since it guarantees an ideal representation of the underlying training sample distribution.

Further, the distance (RMS/object size) between samples and mesh surface is negligible low — far below 1% of the object size at average. This is even more pleasant, since usually, the problem at edges generate large error terms. It proves that discontinuity vertices are worth it. Also the computing times needed are very short, i.e., few minutes in almost all cases.

```

Adjust samples regarding curvature
Calculate average curvature and deviations
Determine and sign discontinuity vertices

for all Boundary vertices do
  if  $\exists$  coalescing candidate then
    Melt boundary
    for all ACO candidates do
      Clean by ACO
    end for
  end if
end for

if Edge collapse operation triggered then
  Collapse edge
  for all ACO candidates do
    ACO-cleaning
  end for
end if

if Vertex split operation triggered then
  Split vertex
end if

```

Fig. 19 The complete GCM algorithm. The outermost loop of the algorithm is neglected, vertex split and edge collapse operations are triggered by counters

All those measurements seem to be far better than those from classical approaches, as long as we could extract them from the regarding papers. The presented algorithm works very robustly. There are nearly no outliers visible in the mesh.

4 Summary and Future Work

We presented a new neural network approach for surface reconstruction purposes — the smart growing cells. The algorithm, on the one hand, profits from the common outstanding facilities of neural networks, on the other hand, well-known problems when using such iterative techniques are avoided successfully.

Growing cells meshing is capable of recognizing and representing arbitrary surface topologies. The network training generates valid meshes at any size and it is able to handle 3D point samples at an arbitrary amount. The mesh accounts for discontinuities like sharp edges and holes, smoothes over missing sample data, handles difficult distributions of sample data, and its granularity efficiently adapts to the underlying curvature of the object shape. The iterative approach enables dynamically changing the training sample set to adjust the network to small changes in a shorter time.

To achieve this, we introduced several mechanisms like, first, aggressive cut out, which can be seen as a brute cleaning mechanism for ill-formed structures. We

	Samples	Vertices	Time [min:sec]	Quality [Delaunay]	RMS/ Size
	36K	30K	1:19	95.6%	4.7e-5
	438K	100K	5:33	95.5%	3.3e-5
	544K	260K	18:33	93.1%	1.7e-5
	14,028K	320K	24:27	98.5%	1.3e-5
	5,000K	500K	42:10	95.9%	2.7e-5
	511K	10K	0:21	99.8%	6.6e-5
	38K	5K	0:12	99.0%	15e-5
	346K	5K	0:12	98.3%	0.7e-5

Fig. 20 Results with objects from the Stanford 3D Scanning Repository. “Quality” means the percentage of triangles which hold the Delaunay criterion. RMS/Size is the root of the squared distances between the original point samples and the triangle mesh, divided by the longest edge of the maximum expansion of the sample set.

found out that these structures are also a great hint for changing connectivities to match the underlying surface topology. Second, face normals are regarded and included in the neural network training loop to assimilate mesh granularity and to make the reconstruction process independent from the sample distribution. Third, we proposed a specific learning strategy for vertices which lie in a sample region which represents discontinuities of the scanned model.

The proof of concept of our approach is enriched by the achieved quality and performance measures. For the tested geometries which each hold specific challenges of reconstruction, we get approximation errors for comparable mesh resolutions that lie far below 1% at average. Mesh quality, measured by the percentage of triangles which comply the Delaunay criterion, lies at 96% at average, and the time needed to compute meshes of several hundreds of thousands of polygons is just few minutes. Thus, with these results, we are confident that we could show that the growing cells meshing approach is a considerable alternative to common reconstruction methods.

Future Work

There are some object structures which can hardly be modeled adequately by the presented method and which will be challenged in future work. These are, for example, planes which are thinner than the average sample distance. From the SGC point of view, these surfaces are not clearly distinguishable.

Computation times are that small that one could think of a real-time reconstruction approach. Scanning users in VR applications for sending their shape to other

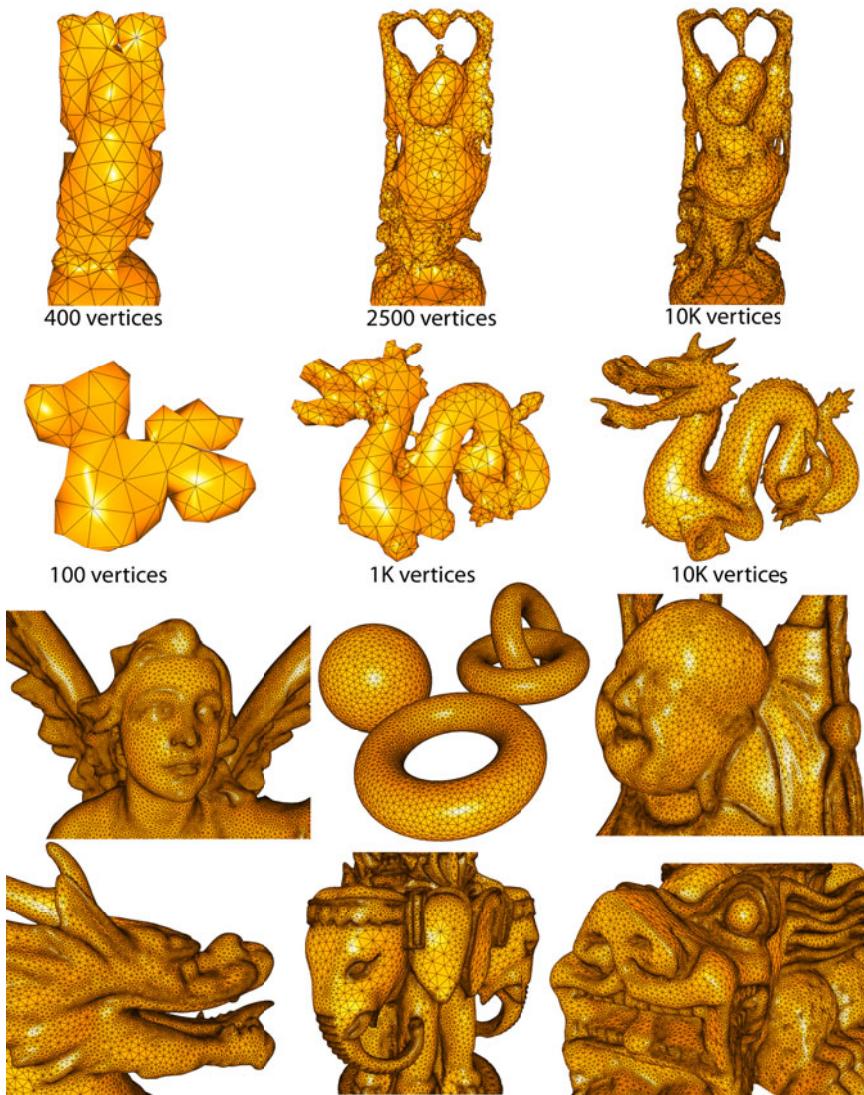


Fig. 21 The top two lines show each three states during a complete training process together with the number of vertices which have been created at that time. The lower lines expose some pictures of reconstructed models.

places could be a impressive feature. Here, an effective parallelization of the algorithm would be required and could be accomplished by executing sample adjustment on several processing units concurrently.

Generally, the SGC approach should be able to bring a lot of new ideas because of its flexibility and since it is capable of learning arbitrary sample sets in a very short time.

Acknowledgements. The authors would like to thank Kai Burjack for his support in rendering the presented images.

References

1. Álvarez, R., Noguera, J.V., Tortosa, L., Zamora, A.: A mesh optimization algorithm based on neural networks. *Inf. Sci.* 177(23), 5347–5364 (2007)
2. Edelsbrunner, H., Mcke, E.P.: Three-dimensional alpha shapes (1994)
3. Fritzke, B.: Growing cell structures - a self-organizing network for unsupervised and supervised learning. *Neural Networks* 7, 1441–1460 (1993)
4. Fritzke, B.: A growing neural gas network learns topologies. In: Tesauro, G., Touretzky, D.S., Leen, T.K. (eds.) *Advances in Neural Information Processing Systems 7*, pp. 625–632. MIT Press, Cambridge (1995)
5. Hoffmann, M., Vrady, L.: Free-form surfaces for scattered data by neural networks. *Journal for Geometry and Graphics* 2, 1–6 (1998)
6. Hoppe, H.: Poisson surface reconstruction and its applications. In: *SPM 2008: Proceedings of the 2008 ACM symposium on Solid and physical modeling*, p. 10. ACM, New York (2008)
7. Hoppe, H., Derose, T., Duchamp, T., Halstead, M., Jin, H., McDonald, J., Schweitzer, J., Stuetzle, W.: Piecewise smooth surface reconstruction, pp. 295–302 (1994)
8. Hoppe, H., DeRose, T., Duchamp, T., McDonald, J.A., Stuetzle, W.: Surface reconstruction from unorganized points. In: Thomas, J.J. (ed.) *SIGGRAPH 1992*, pp. 71–78. ACM, New York (1992)
9. Huang, Q.-X., Adams, B., Wand, M.: Bayesian surface reconstruction via iterative scan alignment to an optimized prototype. In: *SGP 2007: Proceedings of the fifth Eurographics symposium on Geometry processing*, pp. 213–223. Eurographics Association, Aire-la-Ville, Switzerland (2007)
10. Ivriessimtzis, I., Jeong, W.K., Lee, S., Lee, Y., Seidel, H.P.: Neural meshes: surface reconstruction with a learning algorithm. Research Report MPI-I-2004-4-005, Max-Planck-Institut für Informatik, Stuhlsatzengasseweg 85, 66123 Saarbrücken, Germany (2004)
11. Ivriessimtzis, I., Jeong, W.K., Seidel, H.P.: Neural meshes: Statistical learning methods in surface reconstruction. Tech. Rep. MPI-I-2003-4-007, Max-Planck-Institut für Informatik, Saarbrücken (2003)
12. Ivriessimtzis, I., Lee, Y., Lee, S., Jeong, W.-K., Seidel, H.-P.: Neural mesh ensembles. In: *3DPVT 2004: Proceedings of the 3D Data Processing, Visualization, and Transmission, 2nd International Symposium*, pp. 308–315. IEEE Computer Society, Washington (2004)
13. Ivriessimtzis, I.P., Jeong, W.K., Seidel, H.P.: Using growing cell structures for surface reconstruction. In: *SMI 2003: Proceedings of the Shape Modeling International 2003*, p. 78. IEEE Computer Society, Washington (2003)
14. Kohonen, T.: Self-Organized Formation of Topologically Correct Feature Maps. *Biological Cybernetics* 43, 59–69 (1982)

15. Kolluri, R., Shewchuk, J.R., O'Brien, J.F.: Spectral surface reconstruction from noisy point clouds. In: SGP 2004: Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing, pp. 11–21. ACM, New York (2004)
16. Mari, J.F., Saito, J.H., Poli, G., Zorzan, M.R., Levada, A.L.M.: Improving the neural meshes algorithm for 3d surface reconstruction with edge swap operations. In: SAC 2008, pp. 1236–1240. ACM, New York (2008)
17. Muraki, S.: Volumetric shape description of range data using “blobby model”. In: SIGGRAPH 1991: Proceedings of the 18th annual conference on Computer graphics and interactive techniques, pp. 227–235. ACM, New York (1991)
18. Storvik, G.: Bayesian surface reconstruction from noisy images. In: Interface 1996 (1996)
19. Taubin, G.: A signal processing approach to fair surface design. In: SIGGRAPH, pp. 351–358 (1995)
20. Tournais, J., Wormser, C., Alliez, P., Desbrun, M.: Interleaving delaunay refinement and optimization for practical isotropic tetrahedron mesh generation. In: SIGGRAPH 2009: ACM SIGGRAPH 2009 papers, pp. 1–9. ACM, New York (2009)
21. Vrady, L., Hoffmann, M., Kovcs, E.: Improved free-form modelling of scattered data by dynamic neural networks. Journal for Geometry and Graphics 3, 177–183 (1999)
22. Yoon, M., Lee, Y., Lee, S., Ivrissimtzis, I., Seidel, H.P.: Surface and normal ensembles for surface reconstruction. Comput. Aided Des. 39(5), 408–420 (2007)
23. Yu, Y.: Surface reconstruction from unorganized points using self-organizing neural networks. In: IEEE Visualization 1999, Conference Proceedings, pp. 61–64 (1999)

Collaborative Foreground Background Object Isolation and Tracking

John Dragonas, Anastasios Doulamis, George Miaoulis, and Dimitri Plemenos

Abstract. In this chapter, we propose a collaborative framework for efficient isolation and tracking of foreground objects. The algorithm is able to operate in very complex and dynamic background environments. In case that a large deviation of the foreground and/or the background object is encountered, new model classifiers are retrieved or dynamically created to satisfy the current visual characteristics. For this reason, on-line learning classification schemes are incorporated with the purpose of dynamically adjust the performance of a model classifier to the current visual statistics. Object evaluation is accomplished using spatial and temporal criteria. In particular, in case that the motion compensated mask deviates a lot from the current detected mask, the new model selection module is activated. Approximation of the foreground and the background object is performed using the mutual exclusion properties between the two masks as well as the motion information in case that neither the background nor the foreground object is accurate. Experimental results are presented, which indicates the robust foreground detection even in case of complex background content with high dynamic changes.

John Dragonas · George Miaoulis
Technological Education Institute of Athens
Department of Informatics
Ag.Spyridonos St., 122 10 Egaleo, Greece
Tel.: (+30) 2 10 53 85 312; Fax: (+30) 2 10 59 10 975
e-mail: ndoulam@cs.ntua.gr, idrag@teiath.gr

Anastasios Doulamis
Technical University of Crete
Polytechniopolis,
Chania
Tel.: + 28210 37430
e-mail: adoulam@cs.ntua.gr

Dimitri Plemenos
Laboratoire XLIM
Faculté des Sciences, Université de Limoges
83, rue d'Isle, 87060 Limoges cedex, France
Tel.: (+33) 5 55 43 69 74; Fax: (+33) 5 55 43 69 77
e-mail: plemenos@unilim.fr

Keywords: foreground detection, on-line learning strategies, object tracking, collaborative foreground – background detection.

1 Introduction

Foreground extraction in complex and dynamic background conditions is a critical task for many computer vision applications. Examples include methods for industrial planning [1], intelligent video surveillance in a largely unsupervised way [2], crisis management and environmental modelling tools for efficiently handling of natural disasters [3], detection of abnormal behaviours in surveyed videos [4] and a semantic characterization and annotation of video streams in various domains (e.g., sports, news, documentaries, movies and surveillance).

The traditional approaches for event detection in videos assume well-structured environments. Therefore, they fail to operate in complex background environments where the background content is dynamically change through time. In addition, the current approaches need manual re-initialization through the usage of supervised training algorithms in order to increase their performance efficiency in conditions where the applied cognitive systems have not been trained. For example, the current tracking approaches assume initial object detection (segmentation). This inherently means that a tracking re-initialization in case that the visual characteristics [such as color, shape, texture] of either the foreground or the background content dynamically change.

In real life applications, the assumption of a well structured environment is usually not valid. The background content is too complex consisting of several objects of different colour, texture and shape properties. In addition, new objects enter / leave the scene. Another very important aspect is the occlusion that significantly deteriorates the tracking efficiency.

For all the above reasons, foreground extraction in complex and dynamic background environment is a challenging research agenda. The common approaches used for isolating foreground objects, using “Background Subtraction” methodologies [6], fail to operate. Though these methods have been enriched recently by new techniques able to handle the object dynamics (see Section 1.1), till now there is no, according to the authors’ knowledge, a robust foreground detection and tracking algorithm that can yield efficient results even in *dynamic background changes, operating in a largely unsupervised way*. This means that the system will be able to dynamically adjust its performance data outside of the initial training set.

1.1 Previous Work

Several algorithms have been proposed in the literature for object detection and tracking. However, due to the complexity of the problem, efficient foreground detection in dynamic and complex background conditions and for a long time period still remains an open research issue in the computer vision society.

Object Detection: One method for efficient object detection is through the usage of on-line learning strategies. These approaches dynamically learn the properties of the background and/or foreground objects using adaptable training mechanisms. The work of [7] presents an adaptable feedforward neural network with the capability of automatically adjusting its weights to every change of the environment. The authors of [7] have applied this innovative neural network structure for isolating foreground objects in teleconference video streams [8], in which at least one person is present in the scene.

However, occlusions are probably the biggest problem in object detection when using on-line learning strategies. The main difficulty is that parts of objects under consideration may appear in the training examples, deteriorating the detection performance. This way, guesses may be erroneous due to the fact that examples assigned to one class may be resembled with examples of other classes. Moreover, it is quite possible the lack of training samples that efficiently resemble the occluded object. The best way to deal with this problem is to perform a partial detection which is less fault-tolerant process but if occlusion is unavoidable we have no other choice [9].

Other approaches exploit background modelling techniques [10], [11]. However, the main bottleneck of such methods is that they fail to generalize well in real life application of dynamic background conditions. For this reason, specialized object detection methods have been applied, like the *part based object detector* [12]. The system detects deformable parts in a hierarchical fashion ranging from the coarsest to the finest resolution. The image is described with a histogram of gradient features at two different scales. The coarse template is rigid and covers the full search window the other finer template relies on parts and requires a higher resolution analysis. The final detection score takes into account a combination of both matched templates.

One idea is to combine the aforementioned two approaches; *the background modelling and the part based person detection*. This way, we can speed up the process of object localization. Additionally, the person detector will be improved as well.

Object Tracking: Tracking of moving objects is the process of locating a moving object (or several ones) in a video sequence. Algorithms that analyze the input frames and give the location of moving targets as an output are used to solve the object tracking problem. Various approaches have been developed in the literature during the previous decades. However, despite being intensively studied, tracking is still a difficult task for many real-world applications. Problems are mainly due to the presence of ambiguities, the noise in image measurements, the variability of the object scene and the presence of multiple objects that may move in opposite direction and overlap one the other. Furthermore, each application introduces different challenges in terms of motion complexity, periodic movements, etc which are still open research issues.

Apart from tracking of objects using Kalman filter techniques [13], Particle filtering (PF) has been proven to be a good solution for tracking moving objects in various surveillance applications [14]. Other methodologies consider the usage of probabilistic colour models [15]. Very recently tracking has been approached

using classification techniques [15] in which tracking is considered as a binary classification problem between the foreground objects and the background.

The main drawbacks of all the aforementioned approaches are that they cannot efficiently track the objects of interest for a long time period. In other words, tracking requires re-initialization after a limited number of frames since its performance severely deteriorates. Therefore, it is necessary the aforementioned schemes to be coupled with other schemes dealing with efficient object localization in complex and dynamic background conditions.

Dynamic foreground detection for identification of person's falls has been reported in the literature very recently in [16]. However, the foreground detection method of this chapter is quite different from that proposed in [16]. More particular, [16] uses a dynamic background modelling algorithm by exploiting iterative motion estimation constrained by time and shape characteristics. On the contrary, this chapter uses a synergetic foreground and background detection approach based on a recursive adaptive boosting classification scheme. This synergetic schemes present advances on foreground detection. However, it is not suitable for the Person Fall application scenario that it is presented in [16]. This is mainly due to the fact that the combined foreground / background approach reduces the applicability of the system to a larger number of foreground objects.

1.2 Contribution

To address the aforementioned issue, an efficient foreground detection algorithm has been presented in one of our previous work [17]. In this chapter, we proposed an adaptable combined foreground / background classifier for extracting the foreground object. The method is based on an intelligent decision mechanism able to detect accurate foreground/ background mask in a largely unsupervised way. In case that either the foreground or the background mask is inconsistent, visual descriptors are extracted from the complimentary mask and an adaptable learning method is used for on-line capturing the foreground / background dynamics. However, in case that neither the foreground nor the background content is precisely extracted, the method exploits the motion information for detecting the foreground object and thus the background .

In this chapter, we extend this idea (see [17]), by using a *collaborative framework for the foreground and the background* modelling. The Foreground object classifier assists the background modelling and vice versa. In this chapter, we evaluate different stored model objects. New model selection is activated in case that a large deviation of the foreground and/or background object is encountered. Foreground / background assessment is accomplished using spatial and temporal criteria. In case that there is no available classifier able to isolate the foreground object, a new model is automatically created from the current visual descriptors of the approximate foreground / background masks. Approximation is accomplished by exploiting the complimentarily property between the foreground and background object [i.e., mutual exclusion] or motion information in case of no accurate foreground, background detection. An iterative implementation of the Lucas Kanade optical flow algorithm [18], [19] is used for detecting motion in the scene.

The system is able to dynamically learn different possible variations of the foreground / background content *in a largely unsupervised way*. In case that an appropriate foreground/ background model exists, the proposed scheme modifies the current classifier accordingly. Instead, in case that the stored models are not able to capture the current foreground/ background characteristics, a new model is created and stored in the model-database.

The algorithm is able to track the foreground object of interest for a long time period, presented small accumulation error. This is due to the fact the tracking method is incorporated with *advanced on line learning and largely unsupervised strategies*, with the capability of adapting the model classifier to new object conditions. The proposed method performs *a kind of an on-line tracking re-initialization*, increasing tracking accuracy.

Our scheme extends the traditional approaches which are based on background subtraction for foreground isolation and tracking. However, background subtraction assumes static scene conditions. On the contrary, the proposed algorithm can operate on complex and dynamic content conditions. Each time a modification of the foreground object occurs, the algorithm seeks from the model-database to find out the most appropriate classifier for the new content [foreground or background]. The new selected model [or the new created model in case that no appropriate model exists] are dynamically adapted to face the current visual characteristics using on-line learning strategies.

The concept of tracking re-initialization has been introduced in [20]. However, in this chapter the focus is to handle the occlusions issues in object tracking. For this reason, object tracking is performed using advanced particle filters that on-line incorporated with neural network classifiers. In case that an object is not able to be tracked due to occlusions, the neural classifier re-establish object tracking by exploiting the visual characteristics of the tracked object. Instead, in this chapter, a completely different focus is introduced; handle the dynamics of foreground/ background content. Therefore, object tracking is accomplished using a collaborative framework of foreground / background detection. This framework is enhanced by a pool of available model classifiers and on-line learning strategies for updating model object parameters to the current visual characteristics.

The proposed algorithm has been testing in real life conditions for monitoring people. The experimental results indicate a very accurate performance, though the dynamics of the background and foreground objects.

This chapter is organized as follows: Section 2 presents an overview of the proposed architecture. Section 3 describes the collaborative framework used for dynamic modification of the model classifiers using on-line learning approaches and creation and/or selection of new object models. Section 4 presents the object evaluation module, while Section 5 the experiments results which reveals the accuracy of the method in real life conditions where the background and foreground content dynamically change. Finally, Section 6 draws the conclusions of this chapter.

2 Overview of the Collaborative Foreground - Background Approach

In this section, we present the main concept of the proposed methodology for robust foreground extraction in complex and dynamic background environments and operation in a largely unsupervised way. The main concept of the proposed scheme is presented in Fig. 1.

As is observed the heart of the proposed methodology is the *collaborative object detection framework*, the purpose of which is to dynamically update and select appropriate background/ foreground models according to the current object visual statistics. The term collaboration stems from the fact that foreground segmentation is assisted from the background modelling and vice versa.

In particular, an image frame is segmented using current object models. The results are forwarded to a decision mechanism with the purpose of evaluating the detected foreground / background regions. Evaluation is performed using *spatial and temporal criteria*. In case that one of the detected mask is not appropriate, the “*object evaluation*” mechanism re-calculates the incorrect mask as the complementary of the corrected one. In this case, appropriate visual statistics are extracted from the incorrect region. These visual characteristics are used for dynamic (on-line) model updating so that the current visual characteristics are taken into consideration into the object model.

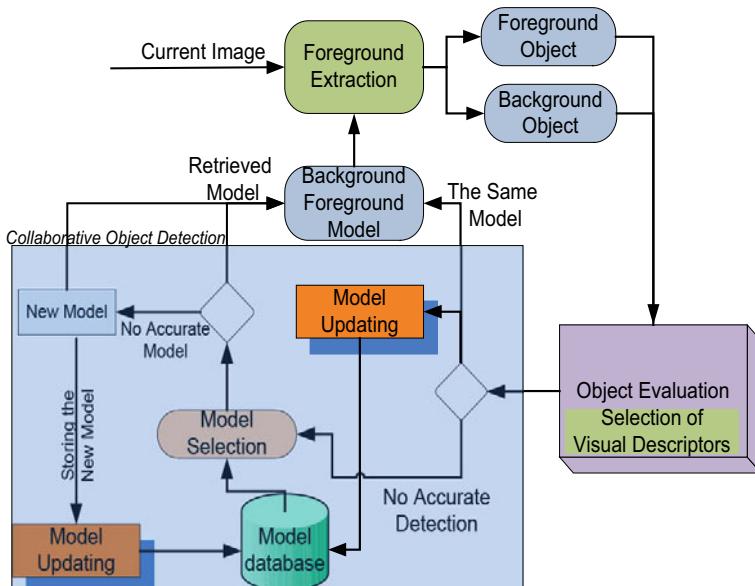


Fig. 1 Overall architecture of the proposed methodology

On the other hand, in case that neither the foreground nor the background mask is accurately detected, the object evaluation module activates the model selection module with the purpose of estimating a more appropriate model from the database. The retrieved model is used as input to background/foreground segmentation task.

Both the new model selection and the model updating are incorporated into the *collaborative framework* (see Fig. 1). The purpose of this framework is to assess different object models in order to retrieve the most appropriate classifier for the current environment. It is clear that in case that no suitable model exists from the set of available classifiers, a new model instance is created and stored in the model database. The new created object model is designed to address the current captured visual characteristics of the content.

The retrieved models are undergone an on-line updating in order to take into consideration the current visual characteristics of the image. Since no accurate foreground/ background masks have been detected in this case, we assume that the moving regions corresponds to the foreground content, while regions far away from the moving object are the background content.

3 Collaborative Foreground – Background Isolation and Tracking

The collaborative foreground – background isolation framework is the heart of the proposed algorithm. The framework is able to alter the foreground and/or background model in order to face the dynamics of the environment. It is clear that in real-life application scenarios, new foreground objects enter a scene and the visual characteristics of the background content are highly changing through time. Consequently, there is no a robust model able to face all the different variations of the foreground and/or background content. Different foreground objects present quite different and contradictory visual characteristics, like colour, shape and texture information. To face these difficulties, we need to dynamically adjust the model parameters to the current visual characteristics through the usage of on-line learning strategies. However, such approach presents the drawback that the previous learned knowledge is not used in the object detection process, a fact that deteriorates the tracking performance. It is clear that it is no possible to learn an efficient classifier using only few samples regarding the current visual characteristics of the environment. Therefore, it is more preferable to retrieve a similar object model and then the adaptation is accomplished on that model.

Another notice is that the foreground and the background objects are complementary. This means that the mutual exclusion property is held for the foreground and background mask. Therefore, it is possible accurate detection and tracking of one of the two masks to assist the detection and tracking of the other. This *collaborative framework* is used in this chapter robust detection and tracking of foreground regions in complex and dynamic scene conditions.

The collaborative framework consists of three main modules; *the on-line model selection, the on-line model updating and the dynamic model creation module*.

These three functions are responsible for (i) retrieving a new background and/or foreground model, suitable for the current situation, (ii) altering the parameters of the current model according to the current visual characteristics, and (iii) for creating new models in case that the stored ones are not able to provide satisfactory results. In the following subsection, we describe these three main modules of the proposed collaborative framework.

3.1 On-Line Model Selection

The purpose of this module is to select the most relevant model for the background and/or foreground content from a set of available models. In case that there is no suitable model a new object model is created. The model selection module is activated according to the results of the “*object evaluation*” module. In cases where the foreground and/or the background mask is not detected with high accuracy, we activate the model selection module. The accuracy of the mask detection is performed using spatial and temporal consistency criteria as described in Section 4.

Let us denote as $M_f(I)$ and $M_b(I)$ the detected foreground and background mask for the current image frame I . It is clear that these masks are mutual exclusive sets in the sense that $M_b(I) \cap M_f(I) = \emptyset$ and $M_b(I) \cup M_f(I) = I$. Using this property, we need to accurately detect only one of two masks; the other can be easily estimated as the complimentary set. However, there are cases where neither the foreground nor the background is detected with high accuracy; new foreground objects enter the scene and the luminosity of the background changes. In this case, we approximately estimate the foreground object as the moving segment of the image. The background is estimated as the complimentary mask of the foreground. To reduce possible noise, we spatially extend the foreground mask, using for example dilation morphological filters, and then we estimate the background object.

Selection of Visual Descriptors: For each of the detected foreground and background mask, we extract visual descriptors to characterize the object content. The object evaluation module assesses the foreground and the background masks and using the mutual exclusive property and motion information, extracts high confident foreground/ background regions. In this chapter, the first zigzag scanned coefficients of the Discrete Cosine Transform are used as visual descriptors of an image block of 8x8 pixels. In particular, we split the estimate foreground / background mask into 8x8 blocks and we extract a visual descriptor for each image block. The main advantage of this approach is the fact that visual descriptors can be directly estimated from the compressed image, video data. Usually, video monitoring systems capture and transmit video content using the MPEG compression standard. In the MPEG format, video content is presented by the DCT coefficients in an 8x8 block resolution. Thus no further, video decoding is required.

Model Retrieval: Once a set of visual descriptors have been extracted from either the foreground and/or background content, the next step is to select an appropriate

object model. Let us denote as \mathbf{d}_f and \mathbf{d}_d the set of visual descriptors for an image block belonging to the foreground and background object respectively. Let us also denote as $f(\cdot)$ the non-linear object model. Then, it is held the following equation

$$P = f(\mathbf{d}_{\{f,b\}}) \quad (1)$$

where we refer as $\mathbf{d}_{\{f,b\}}$ the visual descriptor of an image block belonging either to the foreground or the background content. Variable P denotes the probability of the respective block to belong to the foreground and/or background object. It is clear that equation (1) produces a greyscale mask about the foreground and background if it is applied for all image blocks of the frame. Then, the best function can be estimated as the one produces the minimum error between the produced greyscale image and the estimated foreground or background mask.

$$\hat{f} = \arg \min_f E = \arg \min_f (P - M_{\{f,b\}}(I)) \quad (2)$$

In case that the minimum error is very large, there is no a suitable object model in the database for the current environment. In this case, we create a new model and we store it in the database. The parameters of the model are estimated using the visual descriptors, extracted from the current image content.

3.2 On-Line Model Updating

Let us assume that using the aforementioned methodology, an appropriate model is selected and retrieved from the database. It is expected that the selected object model requires a parameter updating, since the current visual characteristics are different from the one used for training the model. Model updating is accomplished using on-line learning strategies. Using model updating, we can perform efficient object detection and tracking in a largely unsupervised way, that is in conditions outside of the training set.

In this chapter, on line learning is accomplished by the application of the AdaBoost algorithm. AdaBoost (Adaptive Boosting) is a machine learning algorithm, formulated by Yoav Freund and Robert Schapire [24]. The algorithm takes as input a training set that describes the current visual statistics. The set contains the visual descriptors \mathbf{d}_f of an image block as well as the respective probability of belonging that block to the foreground and background object. AdaBoost is an algorithm of constructing a strong classifier as linear combination of simple weak classifiers.

$$f(x) = \sum_{t=1}^T a_t h_t(x) \quad (3)$$

where we denote as $h_t(\cdot)$ the output of the weak classifier [22], [23]. The weak learner's job is to find a weak hypothesis $h_t(x)$ appropriate for the distribution D_t . Each weak classifier learns by considering one simple feature [24].

3.3 Dynamic Model Creation

In case that there is no accurate model in the database, a new object model is created. The model is trained using only the visual descriptors of the current image frame. In this case, a model suitable for the current visual statistics is created. The model is able to adapt its behaviour using the adaptive algorithm, described in the previous section. In this case, new visual statistics are incorporated in the model, making the model more robust in detecting the current foreground/ background object.

4 Object Evaluation –Foreground / Background Approximation

The object evaluation module monitors the performance of the collaborative foreground/ background detection algorithm. The performance is measured using spatial and temporal consistency of the detected mask. For this reason, the current detected foreground (background) object is compared against the previous foreground/ background detection results. In case that a large deviation takes place, the evaluation mechanism, re-estimate the background/ foreground position and activates the collaborative framework with the purpose of retrieving a more suitable object model.

Let us denote as $M(I)$ a detected mask on image frame I . Let us also assume that we have estimated the motion vector fields for the current image frame. Then, we can estimate the new location of the detected mask at the next frame using the following equation

$$M(I+1) = MC[M(I)] \quad (4)$$

Where $MC[\cdot]$ is the motion compensation operation of the current image frame. It is straightforward to estimate the motion compensation mask, if we know the motion vector field of the image. We apply an iterative implementation of the Lucas Kanade Optical flow method for estimating of the motion field of the current image frame. The iterative implementation is an acceleration approach of the Lucas kanade algorithm.

Using the estimated mask $\hat{M}(I+1)$ from the motion field and the actual detected mask $M(I+1)$ using the object classifiers, we are able to compute an error which illustrates the deviation between these two masks.

$$E = \|M(I+1) - \hat{M}(I+1)\| \quad (5)$$

Error E indicates the number of pixels of the detected mask $M(I+1)$, which have no correspondence with the pixels of the motion compensated mask $\hat{M}(I+1)$.

Equation (5) represents the temporal and the spatial consistency of the object. Therefore, in case of a large error, the object evaluation module activates the collaborative framework in order to select a new object model or update the parameters of the current object. Four different scenarios can be discriminated according to the value of error E over the foreground and/or background content.

1. Both the error of the foreground and the background object are low. This means that both the objects are accurately detected. Consequently, the same object model (foreground and background) is used at the next image frame.
2. The foreground error is high but the background error is low. This means that the foreground classifier does not segment accurately the foreground object. In this case using the mutual exclusion property regarding the foreground and background masks, we are able to accurately estimate the foreground mask as the complimentary mask of the background object. In this case the collaborative framework is activated for updating the foreground classifier (model). For this reason, a set of visual descriptors are extracted from the estimated foreground mask (the complimentary mask of the background content) and the model selection module is activated for retrieving another more suitable content. In case that no suitable content is available for the current visual characteristics of the foreground object, a new classifier is created and stored in the database.
3. The third case is quite similar with the second case. The main difference is that the background content is not detected accurately. In this scenario, the background mask is estimated as the complementary mask of the foreground object and the collaborative framework is activated for selecting the new background model.
4. In the fourth case both the foreground and the background content is not detected accurately. In this case, there is no easy algorithm for estimating the background and/or foreground mask. To address this difficulty, we exploit, in this chapter, the motion information. In particular, we consider as foreground the moving object and as background the complimentary of the foreground. To reduce possible error in the detection process, we spatially extent the foreground mask by applying the dilation morphological filtering.

5 Experimental Results

The proposed algorithm have been applied for an efficient foreground detection and tracking. The experiments have been contacted in complex and dynamic background content. Since our method operates on a largely unsupervised manner, we do not need initial supervision for the object models. Initially, no models are

available to perform the foreground segmentation. Therefore, the proposed platform creates new object structures, using classifiers that have been trained from the current estimate visual image descriptors. The algorithm is evaluated at different complex conditions. In the first experiment, we adjust the luminosity of the background. This means that the scene content dynamically changes. On the contrary, the second experiment indicates a new foreground object in the scene. however, in this experiment, we also modify the background content.

Fig. 2 presents an isolation of the foreground object using the proposed methodology. As is observed the foreground is accurately detected even in a complex background environment. Fig. 2(a) illustrates the content of the current captured image frame. In Fig. 2(b), we depict the background content as it has been modeled by the background classifier. Using this background content, we are able to isolate the foreground object using a background subtraction method [see Fig. 2(c)]. The background content is split into a set of 8x8 image blocks and the first zigzag coefficients of the DCT transform is used for presenting the visual content of the background. In a similar way, we represent the visual content of the foreground object using the DCT coefficients of 8x8 image blocks. Using these visual information, we adapt the two classifiers; the background and the foreground model. Finally, Fig. 2(d) presents the motion information of the current image



Fig. 2 Foreground isolation using the proposed methodology. (a) The original captured image frame. (b) The background content. (c) The detected foreground object. (d) The motion information of this image frame.

frame of Fig. 2(a). Motion vectors are depicted as red or green arrows at certain image pixels. These pixels have been appropriate selected from the captured image frame. In this chapter, we have implemented the method of Shi and Tomasi for selected candidate image pixels over which we compute the iterative implementation of the Lucas Kanade Optical Flow algorithm [21].

5.1 Change Of the Scene Content

The first experiment regards the modification of the luminosity conditions of the captured scene. For this reason, we turn off the half of the lights in the room where the experiment takes place. This is depicted in Fig. 3. Fig. 3(a) presents the content of the captured image frame before the change of the luminosity conditions.



Fig. 3 Luminosity Change. (a) The original captured image frame before the luminosity changes. (b) The original captured image frame after the luminosity changes.

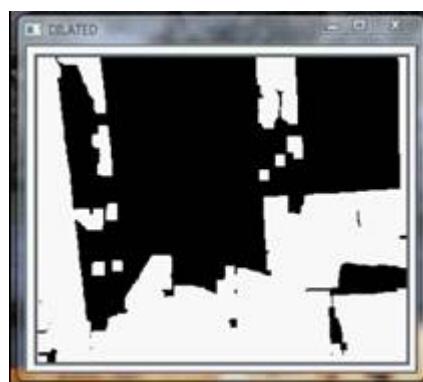


Fig. 4 Erroneous foreground detection. In this case the luminosity of the scene has dramatically changed and the current object models does not provide accurate results.

Instead, Fig. 3(b) presents the content after the turn off of the half of the lights in the room. As is observer, both the foreground and background content (i.e., values of the image pixels) has been changed (reduced).

Fig. 4 presents the performance of the background classifier at this time instance. It is clear that no accurate background mask is detected. This is due to the fact that modifications of the values of the background pixels confuses the background classifier. The same is happened to the foreground model.

In this case, the object evaluation mask is triggered and assess that the detected foreground and background masks are not accurate. For example, it is not held the condition of mutual exclusions for the foreground /background object. In addition, there is no spatial and temporal consistency between the current detected and the previous mask (see Figs 3 and 4). Therefore, the collaborative framework is activated to select a more suitable object model. In our example, we assume that there is no model in the model database. For this reason, the model selection module creates a new classifier from the current selected visual descriptors. In particular, we consider the motion information for foreground object representation. In other words, we compute the motion vectors over “good” image pixels (using the Shi and Tomasi image pixel selection algorithm). Then, we create a boundary box around the detected motion region and we assume that this segment is the foreground object. Then, using the mutual exclusion property, we detect the background region. To avoid possible noise in the approximation of the foreground /background mask, we spatially extend the foreground mask and then we take the complimentary as the background region.

For each approximate mask, we extract visual feature vectors and we train two new classifiers able to capture the visual properties of the foreground and background content. Fig. 5 presents the final foreground, background isolation using the new model classifier. As is observed, the foreground is accurately detected though a high change of image luminosity is encountered.

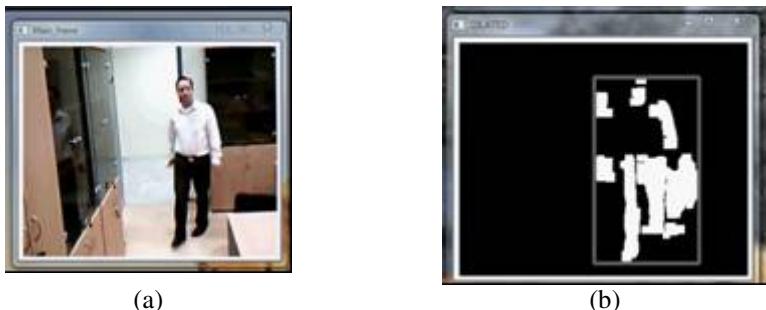


Fig. 5 Foreground and background detection after a luminosity change. (a) the current captured frame after the luminosity change. (b) the detected foreground mask.

5.2 Change of Foreground Content

In the second experiment, we evaluate the proposed algorithm in case of a new foreground object enters the scene. In this case, the object evaluation module activates the collaborative framework, since no accurate foreground mask is detected. In this particular example, we have also modify the background content (see Figs 5 and 3). For this reason, the background classifier also needs updating. The model selection module retrieves from the database a more suitable objet model for the foreground object. Instead, the background classifier is updated to satisfy the new background conditions. In particular, the new retrieved foreground model accurately classifies the foreground object. Therefore, the complimentary mask is used for approximation of the background region. In the following, visual descriptors are extracted from 8x8 image blocks of the background region for model updating. The model adjustment is performed using the Adaboost algorithm. The new model parameters are used for background modelling.

Fig. 5(a) presents the new image content. It is clear that a new foreground object appears in the scene. Fig. 5(b) presents the new background content. Finally, Fig. 5(c) presents the foreground detection mask as well as the background isolation. It seems that the proposed methods accurately segment and extract the foreground mask.

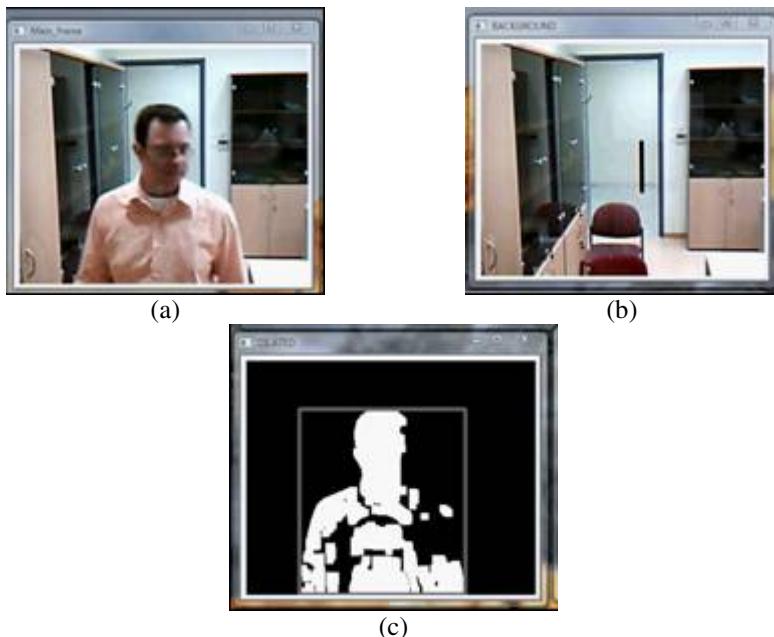


Fig. 6 Foreground and background detection in case that a new foreground object enters the scene. (a) the current captured frame, (b) the new background content. (c) the final foreground/ background detection.

In this chapter, we have presented a method for efficient foreground detection and tracking in complex and dynamic background conditions. The proposed algorithm is based on a collaborative framework interface, which is able to dynamically modify the object classifier used for modelling the background and foreground content according to the current image visual characteristics. In this way, it is possible to perform accurate foreground detection even in case of dynamic and complex environmental conditions. The proposed algorithm is able to detect the foreground object in case of dynamic luminosity changes, new foreground objects entrance, background content modification and active camera scenarios, where the focus and the orientation of the camera dynamically change.

Our conclusions are the following

1. Using static background subtraction approaches, it is not possible to accurately extract the foreground object in case of a dynamic background change. In this case, consistent tracking of foreground object requires new methods for dynamic background (or foreground) updating
2. The proposed methodology is able to accurately detect the foreground object by dynamically re-initializing both the background and the foreground model. Object modification is accomplished using either new model classifiers or dynamic on-line learning strategies for updating the model parameters.
3. Without the usage of on-line learning strategies it is not possible to have accurate foreground tracking. This is due to the fact that minor
4. Collaborative foreground – background detection is a very promising detection and tracking approach. This is mainly due to the fact that foreground and background objects are complimentary and therefore accurate detection of one object can lead to the approximation of the other. Therefore, it is possible to exploit the model performance of one object for improving the detection and tracking efficiency of the other. This collaboration framework is discussed in this chapter, by proposing on-line learning methods for isolating foreground content from complex and dynamic background scenes.

References

- [1] Self Configurable Cognitive Video Supervision, SCOVIS, EU funded project no. 216465 (2007-2013)
- [2] Kosmopoulos, D.I., Doulamis, A., Makris, A., Doulamis, N., Chatzis, S., Middleton, S.E.: Vision-based production of personalized video. *Signal Processing: Image Communication* 24(3), 158–176 (2009)
- [3] Dimitropoulos, K., Semertzidis, T., Grammalidis, N.: Video and signal based surveillance for airport applications. In: 6th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Genova, Italy, September 2-4
- [4] Xiang, T., Gong, S.: Activity based surveillance video content modeling. *Pattern Recognition* 41(7), 2309–2326 (2008)

- [5] Doulamis, N.: Coupled Multi-Object Tracking and Labeling for Vehicle Trajectory Estimation and Matching. *Multimedia Tools and Applications*, 1380-7501 (Print) 1573-7721 (Online available)
- [6] Wu, Q.-Z., Jeng, B.-S.: Background subtraction based on logarithmic intensities. *Pattern Recognition Letters* 23, 1529–1536 (2002)
- [7] Doulamis, A.D., Doulamis, N.D., Kollias, S.D.: On Line Retrainable Neural Networks: Improving the Performance of Neural Network in Image Analysis problems. *IEEE Trans. on Neural Networks* 11(1), 137–155 (2000)
- [8] Doulamis, A., Ntalianis, K., Doulamis, N., Kollias, S.: An Efficient Fully-Unsupervised Video Object Segmentation Scheme Using an Adaptive Neural Network Classifier Architecture. *IEEE Trans. on Neural Networks* 14(3), 616–630 (2003)
- [9] Wu, B., Nevatia, R.: Detection and Tracking of Multiple, Partially Occluded Humans by Bayesian combination of Edgelet part detectors". *International Journal of Computer Vision* 75(2), 247–266 (2007)
- [10] Thongkamwitton, T., Aramvith, S., Chalidabhongse, T.H.: Non-linear learning factor control for statistical adaptive background subtraction algorithm. In: *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS 2005)*, Kobe, Japan, May 23-26, pp. 3785–3788 (2005)
- [11] Culibrk, D., Marques, O., Socek, D., Kalva, H., Furht, B.: Neural Network Approach to Background Modeling for Video Object Segmentation. *IEEE Transactions on Neural Networks* 18(6), 1614–1627 (2007)
- [12] Felzenszwalb, P., McAllester, D., Ramanan, D.: A Discriminatively Trained, Multiscale, Deformable Part Model. In: *Proceedings of the IEEE Computer Vision and Pattern Recognition* (2008)
- [13] Gao, J., Kosaka, A., Kak, A.C.: A multi-Kalman filtering approach for video tracking of human-delineated objects in cluttered environments. *Computer Vision and Image Understanding* 99(1), 1–57 (2005)
- [14] Wang, Z., Yang, X., Xu, Y., Yu, S.: CamShift guided particle filter for visual tracking. *Pattern Recognition Letters* 30(4), 407–413 (2009)
- [15] Roth, D., Koller-Meier, E., Van Gool, L.: Multi-Object Tracking driven Event Detection for Evaluation. In: *AREA workshop in ACM Multimedia*, Vancouver, Canada (October 2008)
- [16] Doulamis, N.: Iterative Motion Estimation Constrained by Time and Shape for Detecting Person's Falls. In: *ACM 3rd Inter. conference on Pervasive Technologies Related to Assistive Environments*, Samos (2010)
- [17] Doulamis, N., Doulamis, A., Dragonas, J., Miaoulis, G.: Robust Foreground Detection and Tracking using Dynamic Foreground-Background Modeling Based on Adaptive Boosting. In: *International Conference on Computer Graphics and Artificial Intelligence*, Athens (May 2010)
- [18] Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: *Proceedings of Imaging Understanding Workshop*, pp. 121–130 (1981)
- [19] Bouguet, J.Y.: Pyramidal Implementation of the Lucas Kanade Feature Tracker. Intel Corporation, Microprocessor Research Labs (2000)
- [20] Doulamis, N.: Coupled multi-object tracking and labeling for vehicle trajectory estimation and matching. In: *Multimedia Tools and Applications*. Springer, Netherlands (2009), doi:10.1007/s11042-009-0370-0

- [21] Shi, J., Tomasi, C.: Good Features to Track. In: Proc. of the IEEE Inter. Conference on Computer Vision and Pattern Recognition, pp. 593–600 (1994)
- [22] Grabner, H., Leistner, C., Bischof, H.: Semi-Supervised On-line Boosting for robust Tracking. In: European Conference on Computer Vision (2008)
- [23] Grabner, H., Sochman, J., Bischof, H., Matas, J.: Sequential trained On-line Boosted Classifiers for Visual Tracking. In: International Conference on Pattern Recognition (2008)
- [24] Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences 1, 119–139 (1997)

Using Fuzzy Logic to Describe Vague Solid Models

Angel A. Zaldivar, Juan C. Torres, and Alejandro Leon

Abstract. In the initial stages of a design process, information is intrinsically imprecise due to the vagueness of designers' ideas and their lack of initial information. The geometric modeling capabilities provided by current CAD systems are unable to manipulate an initial rough model or objects the geometry of which is incomplete. Our approach aims to close the gap between the inexact and vague ideas that designers have in early stages and the exact geometric representation that the conventional modeling methods provide. This can be achieved using the Fuzzy Sets Theory to link linguistic terms that portray geometric forms with the parameters that control the model's geometry. This paper proposes a new, alternative paradigm for geometric shapes modeling in CAD systems based on fuzzy solid models, allowing designers to specify and retain fuzzy characteristics in shape while performing a number of essential tasks during the conceptual design stage.

1 Introduction

Design is an activity that has been carried out by humans for thousands of years, traditionally with the goal of finding the optimal solution to specific problems. Many design tasks and activities involve decision making. The quantity and the quality of available information is different at each stage of the design process. Wang evidences the need of manipulating varied kinds of uncertainty and vagueness which

Angel A. Zaldivar
University of Holguin, Cuba
e-mail: angel@facinf.uho.edu.cu

Juan C. Torres
University of Granada, Spain
e-mail: jctorres@ugr.es

Alejandro Leon
University of Granada, Spain
e-mail: aleon@ugr.es

are implicit in the early stage of the design process through systematic techniques and methods [27].

1.1 *Vagueness during Geometric Modeling*

At the preliminary/conceptual stage very little information is available to the designer. Gradually, more information is produced and the knowledge is continuously enriched and refined. At this stage, design information is intrinsically imprecise due to the vagueness of designers' ideas and their lack of initial information. At this point, there exists only a set of abstract and partial proposals and a group of wrong defined product requirements. The objects are imagined neither with exact forms nor perfectly dimensioned, but as vague and subjective shapes. This phase consists in generating design concepts while exteriorizing mental imagery in observable representations in order to evaluate them and identify the best.

Geometric conceptualization of a product is an attempt at expressing creative ideas quickly, without the need for precision. Geometric details are added when sketches are drawn using a CAD/CAM system for advanced manipulation and visualization.

However, while CAD systems have evolved considerably in functionality, expressiveness and modeling power over the last few decades, the principles that govern the user interface have remained tied to old modeling methods. Classical modeling language is oriented to well formulated object definition, allowing a real body to be built with mathematical precision. Therefore the designer must have a clear idea of how the object will look before specifying crisp geometric entities, its topological relation and operations to construct the model.

The exact geometric representation does not allow a designer to reflect the pointless and inexact ideas that are manipulated on the model. The introduction of an initial rough sketch via fuzzy criteria is impossible, making it rather unintuitive and thereby diminishing its performance. The representation schemes in classic solid modeling were conceived under Differentiable Manifolds theory [6] and tied to conventional formalisms such as uniqueness and unambiguity in the abstract domain of Euclidean geometry. The fact that these geometric models are deterministic and precise, and leave no room for uncertainty or modality, means that when an early idea is refined, the original is simply substituted with a precise model. The low-level geometric entities prevent the inclusion and representation of explicit, heuristic, procedural, cognitive and perceptive design knowledge about the shape. This leads to a premature loss of vague information and a distortion of the designer's ideas.

Furthermore, the modeling systems are unable to manipulate objects whose geometry is incomplete and have pronounced learning curves, use annoying and restrictive dialogues, include hundreds of commands, and do not assist in the choice of design alternatives.

This drawback creates a communication barrier between two spaces: the subjective and imprecise language and the precise mathematical representation, which impedes the abstraction process. Thus, there is an immediate need for new and

alternative representation formalisms of design knowledge, in order to manipulate the different kinds of uncertainty and inherent vagueness of this phase through techniques and systematic methods that take into account the incomplete and uncertain nature of geometric information and provide more flexible, intuitive and user-friendly tools to maximize designer creativity and ingenuity.

The designer can qualitatively identify, describe, and analyze shapes and spatial relations through natural language. Therefore, it is appropriate to use words to depict forms, transformation, positions and orientation. This not only makes tools more intuitive, but also provides a better understanding and appreciation of the outcomes.

1.2 Related Work

Over the last three decades, different efforts have been made in this field of research, with works that include vagueness in computational representation, as well as those that relax the modeling language in CAD tools.

As a result of these efforts, a number of shifting geometrical representation schemes have been proposed (such as non manifold models) that break with traditional solid formalism and improve the scope and support of modeling methods [28, 12, 25].

Knowledge-intensive systems are an attempt to represent real knowledge for reasoning using an inference engine. They propose a framework for man-machine communications [10].

Conceptualization has been discussed in the literature, with paradigms guided by drawings and sketches that evoke the freedom to draw with paper and pencil [17]. The sketch based user interfaces have been used to describe 3D models using instruments of natural communication, such as the interpretation of calligraphic expressions in order to generate objects and exteriorize concepts [23]. Others reconstruct geometric objects in 3D from drawings in 2D using axonometric bloated methods [8].

The application of Fuzzy Sets Theory to geometric modeling is not a new field of investigation. Many works have focused on two dimensional fuzzy shape modeling, using the approach in image processing and computer vision problems [24, 7].

Important results in this area have been made by Yamaguchi and Nakamura, who propose a representation of uncertain forms in the early stage of design through Fuzzy Sets. The membership function represents the probability that one point belongs to a set. The probabilistic solids are created from crisp solids [30]. Pham explores the interaction between design variables and aesthetic properties, and explains how fuzzy logic can be used to specify forms that satisfy aesthetic requirements [19]. In another paper, Pham proposes a theoretic approach based on fuzzy logic obtaining a shape specification system, which allows designers to handle rough models in an intuitive way [22]. Horvath proposes imprecise models that are defined with a discrete representation of particles to express uncertainty (singularity net) [14].

Other works use natural language to describe forms. Gandhi and Myklebust discuss feature based modeling methods with natural language. This approach uses adjectives and forms transformation verbs to place additional constraint on model parameters. The internal geometric model is based on bspline patches. Positions and orientations are also specified with natural language phrases [11]. Batchelor provides a list of useful terms to describe shape. Many of these are only suitable for simple geometry [4]. Martin illustrates a way to describe inexact shape with natural language for inexperienced users [16].

A variety of approaches to qualitative spatial reasoning have been based on Allen's temporal reasoning approach. He introduced a temporal logic based on intervals and their qualitative relationships in time. The basic elements of Allen's theory are intervals corresponding to events, qualitative relations between these intervals, and an algebra for reasoning about relations between intervals [11]. Other works expand the approach to the spatial domain, arriving at a solid qualitative representation method based on qualitative spatial relations [13].

According to Hernandez the cognitive spatial concept is qualitative in nature. It is not based on exact magnitudes but on comparisons of perceived magnitude. The author proposes a method for spatial knowledge representation and a method for spatial reasoning [13].

In practice, these applications automatize processes rather than achieve highly interactive methods for conceptualization. An alternative approximation is needed, which incorporates interpretation mechanisms for the subjective perception of the designed object, as well as the development of tools that increase creativity in a more intuitive climate. It is also necessary that methods and representations not only exploit the low level of abstraction but take into account the portion of high-level knowledge of the object as concepts or association among concepts, and permit their interpretation in distinct abstract levels. Not enough emphasis has been placed on the use of the expressive richness of natural language to describe geometric forms, which is more in tune with the subjective perception of human thought. To achieve this, the following criteria are required:

- The ability to process a linguistic term set that describes geometric forms.
- A natural mechanism of forms input (not orientated to geometric entities).
- A geometric model that shares a duality between the qualitative criteria and the quantitative parameters, with the goal to establish the bridge between subjective attributes and mathematical precision.
- A computational representation that resolves vagueness and is able to interpret and process it.

In order to achieve a revolution in CAD tools for conceptual geometric design, computational models are required that can be equally well interpreted by computer and human, making the transition from the conceptual stage to the geometric modeling stage easier. The conceptual model created by composition of concepts must be transferred to a CAD/E system without loss of information and without the need for extreme intervention of the user.

2 Fuzzy Solid Models

The core of the geometrical conceptualization framework demands a system able to specify the qualitative ideas about fuzzy geometric design in order to obtain a rough model of the desired shape. The system must also be able to improve the performance of tasks using this model while maintaining fuzzy features.

In order to model a solid computationally it is necessary to create an idealization of the physical object, which must have an obvious and intuitive connection with the solid. The problem is reduced to find the function $f : T \rightarrow P$ that maps or translates between two universes: the subjective terms space T of natural language that describes forms, and the geometric parameters space P that quantifies and governs changes in the shape of the object. This process must be clear to the user.

2.1 Fuzzy System Design

The Fuzzy Sets Theory is useful for accommodating and representing problems whose data definition and rules have implicit vagueness, closely simulating human mental behavior. The geometric properties of fuzzy sets have been investigated since the 1960s. With the Fuzzy Sets Theory we can deal with the description, location and reasoning process in the shape design using linguistic terms. The fuzzy model can qualitatively depict a system with linguistic terms; however, these terms are associated with fuzzy amounts, allowing crisp values to be obtained as a quantitative expression of the model.

Traditional logic based systems permit restricted knowledge representation because these systems have two states: yes or no, all or nothing, the fulfillment rule is true or false. In comparison, a fuzzy logic based system accepts that its rules are fulfilled in a certain measure. If these rules describe the system and are fulfilled to a certain extent, they must influence the result of the system.

Parametric geons is a model proposed by Barr [2], improved by Bierdman [5], and established definitively by Wu and Levine [29]. It is inspired by simple, symmetric and regular shapes, represented by super-ellipsoids and their deformations, as shown in Figure 1.

Bierdman uses the geons model and its qualitative dimension for object recognition. He proposes that a complex object can be represented as a geometrical entities hierarchy of geons [5]. The approach is based on the fact that our visual system uses co-linearity, curvature, symmetry and co-termination to discriminate and recognize shapes. This suggests the capability of the brain to deconstruct a complex shape in simple entities using its qualitative properties; with this structure, possible alternative interpretations of an element or object could be more visible.

This model has several volumetric properties that can be controlled by a set of parameters, where each one is attached to qualitative geometric properties such as roundish, squarish and size, as well as global shape constraint, so that the model's attributes share quantitative and qualitative duality, as can be seen in Figure 2. In its geometric domain are included most of the geometric primitives (box, sphere, cylinder).

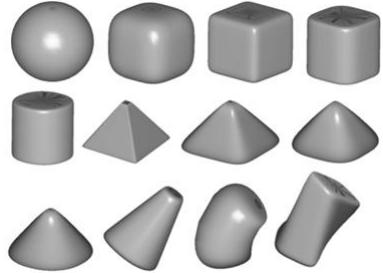


Fig. 1 Parametric geons

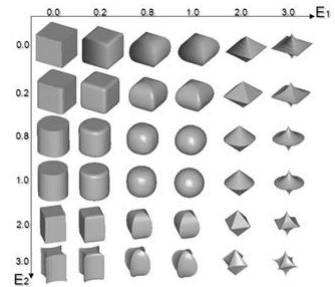


Fig. 2 Super-ellipsoids in function of the parameters E_1 and E_2

The implicit formulation is widely employed since it possesses a set of mathematical properties (1) that facilitate the inclusion test of a point in the model. If then the point is inside the super-ellipsoid, if then the point is at its boundary and if then the point is outside the super-ellipsoid.

$$f(x, y, z) = \left(\left| \frac{X}{a_x} \right|^{\frac{2}{E_2}} + \left| \frac{Y}{a_y} \right|^{\frac{2}{E_2}} \right)^{\frac{E_2}{E_1}} + \left| \frac{Z}{a_z} \right|^{\frac{2}{E_1}} \quad (1)$$

The parameters E_1 and E_2 control the roundish/squarish shape properties. The parameters a_x , a_y and a_z are scalar parameters in each axis.

The geons can also be expressed as the spherical product of two super-conic curves in a parametrical equation (2). This expression is more feasible than the implicit model for sampling the surface, generating a mesh and rendering it. In practice both formulations are useful in the computational implementation of the model.

$$\begin{aligned} x &= a_x \cos^{E_1} \phi \cos^{E_2} \beta \\ y &= a_y \cos^{E_1} \phi \sin^{E_2} \beta \\ z &= a_z \sin^{E_1} \phi \end{aligned} \quad (2)$$

Where $-\pi/2 \leq \phi \leq \pi/2$; $-\pi \leq \beta \leq \pi$; $0 < E_1, E_2 < \infty$. The variables ϕ and β constitute angles in a polar coordinates system, representing the longitude and the latitude respectively, both allow to cover the whole surface of the geon.

To increase the geometric domain of the model, the following transformations are applied: bending, controlled by k parameter (3); twisting, controlled by t parameter (4); and tapering, controlled by K_x and K_y parameters in the cross section (5). The deformation operations add the remaining elementary geometric primitive (cone, pyramid, wedge). So that, every geon is expressed completely by means of nine parameters ($E_1, E_2, a_x, a_y, a_z, K_x, K_y, k, t$), which is a desirable representation for its computational manipulation [3].

$$\begin{aligned} X' &= k^{-1} - \cos\theta(k^{-1} - x) \\ Y' &= y \\ Z' &= (k^{-1} - x)\sin\theta \end{aligned} \quad (3)$$

Where $\theta = kz$ is the bending angle.

$$\begin{aligned} X' &= x\cos\theta - y\sin\theta \\ Y' &= x\sin\theta - y\cos\theta \\ Z' &= z \end{aligned} \quad (4)$$

Where $\theta = t(z/a_z + 1)\pi/2$ is the twisting angle.

$$\begin{aligned} X' &= \left(\frac{K_x}{a_z}z + 1\right)x \\ Y' &= \left(\frac{K_y}{a_z}z + 1\right)y \\ Z' &= z \end{aligned} \quad (5)$$

It is necessary to obtain a triangular mesh to render and edit the model. The generation of the triangular mesh of the geon is executed by sampling the surface of the geon by means of its parametrical equation. Later the data structure is covered to apply at every point of the mesh the deformation operations.

The qualitative influence on the shape provided by each one of the geons' parameters and its quantitative duality make then an ideal geometric model to establish the desired bridge between subjective qualities and mathematical precision through a set of rules that connect both dimensions.

During the transformation process, the volumetric qualities are not varying monotonically according to a linear scale, also the input variables (qualitative descriptors) are implicitly vague and the rules to be established (relation between the qualitative descriptors and the quantitative parameters) are of linguistic nature. Under these circumstances, a system based on boolean rules permits very limited representation of the knowledge of the system. By comparison, a system based on fuzzy rules produces more tolerant results in view of the uncertainties of variables and rules, so that imprecise conclusions are drawn from an imprecise premise collection.

A fuzzy system is composed of a knowledge base, which comprises the logical core in the form of linguistic rules; a fuzzification interface, which has the effect of transforming crisp data into fuzzy sets; an inference system, which uses these together with the knowledge base to make inference by means of a reasoning method; and a defuzzification interface, which translates the fuzzy action thus obtained into a real action using a defuzzification method. The knowledge base comprises the expert knowledge of the system.

Here, we introduce and design a fuzzy system in which the input variables are geometric shape descriptors, and the inference engine accommodates the fuzzy rule base that relates input variables with output variables which are defined in terms of the parametric geons model, as shown in Figure 3.

The fuzzification interface converts the input information to linguistic values adapted for the system manipulation, each input variable is partitioned into overlapping discreet intervals, according to the semantic meaning and the discourse universe of each subjective descriptor. In this way, part of the system behavior is already described (for example, that a shape is simultaneously slightly roundish and moderately roundish to different extents). All the input variables take values in the range from 0 to 1 to express the proportion in which the linguistic term is evident in the output form. For the variables “roundish”, “squarish”, “ellipsoidish”, “cylindrical”, “flatten”, “thin”, “narrow” and “bevel” we define four linguistic partitions: “nothing”, “slightly”, “moderately” and “extremely”, as shown in Figure 4.

For the variables “conic”, “wide”, “longish”, “deep”, “twisted”, “pinched”, “bend”, “tapered”, “big” and “small” we define six linguistic partitions due to their geometric meaning: “nothing”, “very slightly”, “slightly”, “moderately”, “very extreme” and “extremely”, as shown in Figure 5. It should be mentioned that we use triangular and trapezoidal membership functions to simplify inference process calculation. The “nothing” linguistic partition is a singleton that takes the value 1.0

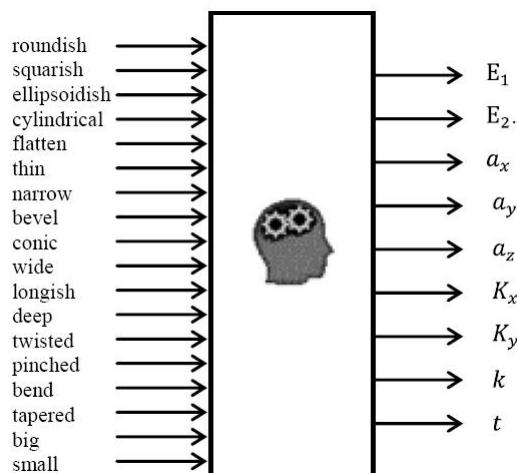


Fig. 3 Fuzzy inference system

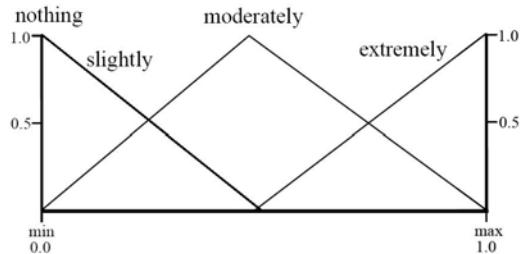


Fig. 4 Linguistic partitions of the variables “roundish”, “squarish”, “ellipsoidish”, “cylindrical”, “flatten”, “thin”, “narrow” and “bevel”

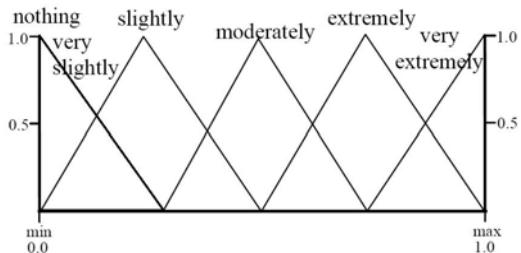


Fig. 5 Linguistic partitions of the variables “conic”, “wide”, “longish”, “deep”, “twisted”, “pinched”, “bend”, “tapered”, “big” and “small”

when the variable value is 0.0, which expresses the state when the output form does not show this modifier in any quantity.

With the goal to give fuzzy nature to the output variables, we divide its domain in linguistic categories with a granulation according to the geometric effect that exert the parameter over the output shape. The output variables E_1 and E_2 produce valid 3D shapes in a range from 0.0 to 3.0, we divided its space into four linguistic partitions: “zero”, “small”, “medium” and “large”, as show in Figure 6. For the variables a_x , a_y and a_z we set their range value from 1.0 to 3.0 and divided their space into six linguistic partitions: “zero”, “very small”, “small”, “medium”, “large” and “very large”, as shown in Figure 7. It should be mentioned that the “zero” partition is a singleton. For the variables K_x , K_y , k and t we set their range value from 0.0 to 1.0 and divided their space into six linguistic partitions similar to the scalar parameters.

The fuzzy inference rules require fundamentally geometric coherence between the descriptive terms and the obtained shape expressed by their parameters. For the design of this system we extend the XFuzzy laboratory. Which is a GNU development environment for fuzzy-inference-based systems. The interface of monitoring module of the Xfuzzy allows checking interactively the behaviour of the system describing its internal functioning. For visual testing of resulting output shape of the inference process we add 3D viewer to the XFuzzy monitoring module, which is updated with the system output variables, displaying the corresponding 3D models. We design a inference rule base to link these descriptors with the shape parameters. Taking the following criteria:

For example, the input variables “roundish”, “squarish”, “cylindrical”, “bevel”, “conic” and “ellipsoidish” are related to the output variables E_1 and E_2 . They can also be connected with a_z . The input variables “pinched”, “conic” and “tapered” are

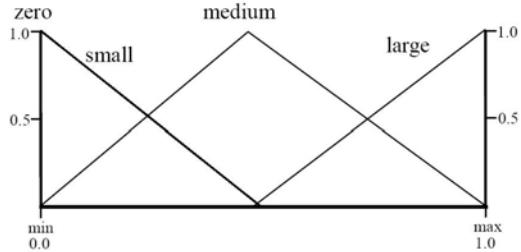


Fig. 6 Linguistic partitions of the variables E_1 and E_2

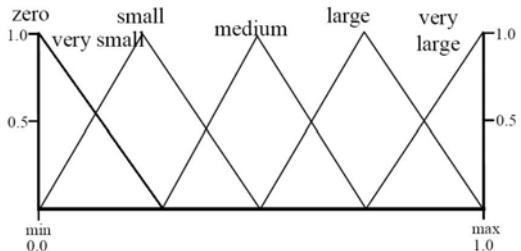


Fig. 7 Linguistic partitions of the variables a_x , a_y , a_z , K_x , K_y , k and t

attached to the output variables K_x and K_y . The input variables “longish”, “wide”, “deep”, “flatten”, “thin”, “narrow”, “big” and “small” are associated with the scalar parameters a_x , a_y and a_z . And the input variables “bend” and “twisted” are linked with the parameters k and t respectively. The system has 64 fuzzy rules.

For the input variable “roundish” we established the following rule:

```

if (roundish = extremely) then E1= small,
                           E2= small;
if (roundish = moderately) then E1= small,
                           E2= zero;
if (roundish = moderately) then E1= zero,
                           E2= small;
if (roundish = slightly) then E1= large,
                           E2= large;

```

The Mandani implication method is the most appropriate because it offers the most flexible way to describe the system. We chose the union operation as the most appropriate aggregation method. As a defuzzification method we used the Fuzzy Mean, called average center (6), where “min” and “max” are the minimum and maximum values of the x variable and *center* is the middle value of the membership function.

$$\text{FuzzyMean}(x) = \frac{\sum_{i=\min}^{\max} \mu_x(i) \times \text{center}}{\sum_{i=\min}^{\max} \mu_x(i)} \quad (6)$$

With this method the output crisp value tends to move softly around the fuzzy output region from one solution to the other. Not much computational cost is demanded and

it is applicable to any output geometry. The expected value can become the extreme of the output variable. The defuzzification process does not have to be executed immediately after the fuzzy inference, the fuzzy information can be retained and used to execute tasks of the modeling process.

By means of a fuzzy approach, we take into account much more of our knowledge of the system in its output values. In this way we include in the final solution the influence of all the rules that can be fulfilled to any extent. Each rule contributes with an area in dependence of the antecedent fulfillment degree and the consequent form that is interpreted as the influence of the rule in the final value; which is computed through an aggregation method (the total sum of the areas of each rule for each variable).

2.2 Formal Look at the Fuzzy Solid Models

A formal language is a set of finite length words (string character) formed by a finite alphabet (characters set). The name language is justified because its structures have well defined rules (grammar) and semantic interpretation (meaning) in a similar way to a spoken language.

Given a formal language D whose semantic interpretation depicts geometric entities in the space, it is important to use the uncertainty of formalized concepts. The grammar can be represented as the set D of pairs (v_i, t) where v_i is a linguistic variable that belongs to the set V of all input variables of the fuzzy system described in the previous section (7), and t is a linguistic term belonging to the set $T(v_i)$ of linguistic labels with which the discourse universe of the variable v_i was partitioned.

$$\begin{aligned} D &= \{(v_i, t) : v_i \in V, t \in T(v_i)\} \\ V &= \{\text{roundish}, \text{squarish}, \text{conic}, \dots\} \end{aligned} \quad (7)$$

It is assumed that the variables of the set V that do not appear in the description D take the partition “nothing” as their associated label (8).

$$\begin{aligned} D &= \{(\text{roundish}, \text{extremely}), (\text{squarish}, \text{slightly}), \\ &\quad (\text{pinched}, \text{moderately}), (\text{conic}, \text{nothing}), \dots\} \end{aligned} \quad (8)$$

M is a function that associates each pair (v_i, t) of the D language with its range $r_i = [u_1, u_2]$ corresponding to the support of the partition t within the discourse universe of the variable v_i (9). If the partition t corresponds to the “nothing” label of singleton-type, then the range will contain a single value $r_i = [0.0]$. The function M performs a query of the fuzzy system knowledge base to generate the ranges set R .

$$M : D \rightarrow R \quad (9)$$

Then for each element of the set D there is a range r_i , which defines a continuous space through which the crisp input value of the variable may move. The Cartesian product of all ranges r_i generate the set K , where each $k_j = (e_1, e_2, e_3, \dots, e_{18})$ is a

crisp input element of the fuzzy system (10), fitting into fuzzy definition determined by formal language D .

$$\begin{aligned} r_1 \times r_2 \times r_3 \times \dots \times r_{18} &= K \\ K &= \{(e_1, e_2, e_3, \dots, e_{18}) : e_i \in r_i\} \end{aligned} \quad (10)$$

Each element e_i is associated with a membership degree $\mu(e_i)$, which is interpreted as the membership degree of the element to the partition t of the variable v_i . In order to compute the membership degree of k_j to the fuzzy definition settled by D it is necessary to use a t-norma as operator \min (11).

$$\mu_D(k_j) = \min(\mu(e_i)) \quad (11)$$

When evaluating each k_j in the fuzzy system $I()$ output values are obtained that describe geometrically the crisp form expressed by this element in terms of the geons parameters. Therefore the inference mechanism translates each element k_j of the set K into an element f_j of the set F (12), which is formed by geometric shapes expressed in term of its parameters $f_j = (E_1, E_2, a_x, a_y, a_z, K_x, K_y, k, t)$. This translation also transfers to each element f_j the membership degree $\mu_D(k_j)$, which is interpreted as the membership degree of the shape to the fuzzy definition D .

$$\begin{aligned} I : K &\rightarrow F \\ F &= \{f_j : I^{-1}(f_j) \in K\} \end{aligned} \quad (12)$$

As a result of this process we obtain fuzzy geometric entities as an extension of the traditional crisp geometric entities or exact entities. Likewise, a fuzzy geometric model could be interpreted as a set of crisp models approximated among them and their variations, where each possesses a distinct membership degree to the fuzzy set. Each set is not a group of arbitrary elements due to the fact that different members of the set have some similitude and the differences between them can be quantified [22]. The set must be considered as a simple entity described by its similitude. It constitutes implicitly a search space that contains all the crisp forms that are included in the fuzzy definition as shown in Figure 8.

If we depict a form as “extremely tapered”, it is obtained a family of shapes that spans the range of values r_i in the domain of the input variable “tapered”, corresponding to the partition “extremely”, and each belongs with degree μ_k to the fuzzy definition as shown in Figure 9.

Analogous with a traditional fuzzy set, a *Fuzzy primitive* described by the formal language D , defines a set of ordered pairs $(f_j, \mu(f_j))$ that constitutes implicitly a family of crisp shapes and approximations among them within the geometric domain $F = \{f_1, f_2, \dots, f_n\}$, where each possesses distinct membership degree $\{\mu_1, \mu_2, \dots, \mu_n\}$ to the fuzzy concept expressed by D (13).

$$\begin{aligned} \text{Fuzzy primitive} &= \{(f_j, \mu(f_j)) : f_j \in F\} \\ \mu &: F \rightarrow [0, 1] \end{aligned} \quad (13)$$

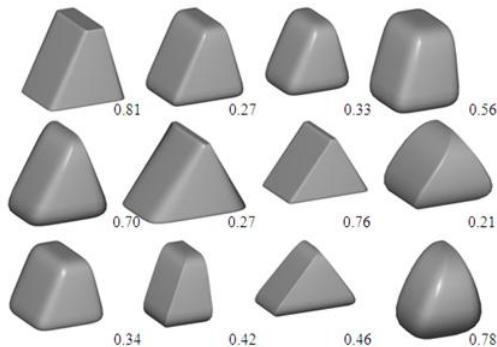


Fig. 8 Slightly squarish and moderately pinched shape with its membership degree

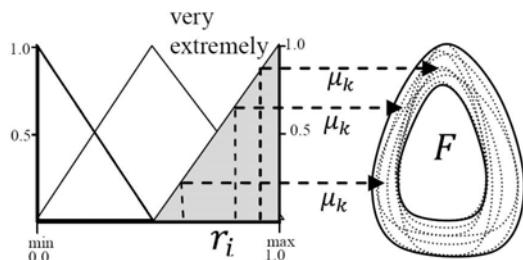


Fig. 9 Very extremely tapered shape family

The fuzzy membership degree of the crisp point p to the fuzzy shape F generated by language D is equal to the highest membership degree of a crisp shape f_j of F where the point p is included (14). The problem with computing this value can be formulated as multidimensional search.

$$\mu_D(p) = \max(\mu(f_i) : p \in f_i) \quad (14)$$

In order to compute the membership degree of a crisp point for a compound fuzzy definition (“forms extremely tapered and moderately squarish”), we use as t-norma operator $\min(\mu_m; \mu_n)$.

When a magnitude is measured with an imprecise instrument, a value interval is obtained that can contain the measured value with a given statistical probability. During conceptualization, human thought always contains a margin of inaccuracy or error when considering the design. A fuzzy geometrical entity can be seen as an abstraction of the geometrical space covered by a shell of geometrical vagueness, expressed by the degree of membership of the points in the space to the solid through a membership function. The higher the membership degree of the points in a portion of the space, the higher the certainty of the membership of that point to the modeling entity will be.

The geometric domain of the parametric geons is short, so in practice, it is necessary to obtain more complex solid models with mechanisms that can propagate the uncertainty of the fuzzy primitive entities to more complex compound models.

Fuzzy set theory is an extension of classical set theory that allows the membership of elements to range from 0 to 1. Membership in classical solid modeling is often viewed as a characteristic function, taking a value 0 or 1 depending on whether the point is inside the solid or not. By analogy, a fuzzy solid can be represented implicitly through a fuzzy membership function $\mu(p)$ (17). The returned value for the fuzzy membership function is interpreted as the membership degree of the point to the fuzzy solid. By definition, all the points of the space belong to the fuzzy solid but some of them have degree 0 (22).

We thus define implicitly the fuzzy solid S as all the points that satisfy the condition $\mu(p) > 0$ and its fuzzy shell that obeys the condition $0 < \mu(p) < 1$. In the definition, the solids are considered regular and not empty (15).

$$\begin{aligned} S &= \{p \in R^3 : \mu(p) > 0\} \\ \mu &: R^3 \rightarrow R \end{aligned} \quad (15)$$

$$\begin{aligned} B &= \{p \in R^3 : 0 < \mu(p) < 1\} \\ I &= \{p \in R^3 : \mu(p) = 1\} \end{aligned} \quad (16)$$

The set of points of the solid S are next divided into two regions: the subset of points I that are completely inside the solid and the subset of points B that are partially inside or in its fuzzy shell (16).

$$\mu(p) = \begin{cases} 0 & \text{si } p \notin S \\ (0, 1) & \text{si } p \in B \\ 1 & \text{si } p \in I \end{cases} \quad (17)$$

$$I \subset S, B \subset S, p \in E^3$$

Therefore, there are two different approximations with respect to the analytical interpretation of a fuzzy solid: an implicit representation that characterizes the solid as a set of points in E^3 , called the topological model, that decides the rules to test which points belong to the set. This is supported by a continuous idealization of the set of points as shown in Figure 10a. The second interpretation is an enumerative representation that characterizes mathematically the solid from the boundary that encloses it. This representation specifies that the rules generate only the set of points, and is supported by a discrete idealization of the set of points. The fuzzy solid can also be seen as a family of crisp shapes, as shown in Figure 10b.

Constructive Solid Geometry (CSG) is the most popular constructive representation in the solid modeling field. Its primitives may be simple shapes or more complex features. Primitive instances and groups may be transformed through rigid geometric transformations (T_g). The transformed instances may be combined through regularized boolean operations (OpB): union, intersection, and difference.

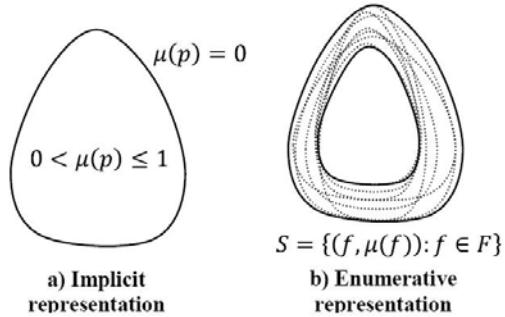


Fig. 10 Fuzzy solid models representations

The construction process of complex fuzzy shapes using CSG is conceptually the same as the traditional solid modeling. The only difference is that the primitive entities are fuzzy; therefore the boolean operations change consequently (18). Based on this definition, we specified progressively more complex geometric shapes by means of artifices of implicit modeling. A natural approach for semi-analytical sets is to define constructively more complicated membership functions treating them as a fuzzy membership function and operating them with t-norma and t-conormas, equivalent in fuzzy set theory to using standard crisp set operators ($\cap, \cup, -$) (32). The strategy that reduces the computational cost is to represent constructive geometry in the typical tree data structure, with defined primitive membership functions stored in the leaf nodes and fuzzy set operators (t-norma and t-conorma) in the interior nodes. Thus, the algorithm for evaluating the membership degree from a point to a complex fuzzy shape can be naturally implemented while evaluating the tree and inverting each boolean operation in terms of the correspondent logical operation (\odot) (19).

$$\text{Solid} = T_{g1}(S_1) \text{ OpB } T_{g2}(S_2) \mid T_{g3}(\text{Primitive}) \quad (18)$$

$$\text{Fuzzy Solid} = T_{g1}(FS_1) \odot T_{g2}(FS_2) \mid T_{g3}(\text{Fuzzy Primitive})$$

$$p \in (S_1 \text{ OpB } S_2) \implies (p \in S_1) \odot (p \in S_2) \quad (19)$$

The rigid transformations are accumulated in a matrix T , and its inverse operates to transform the audited point (20).

$$p \in T[S_x] \implies T^{-1}[p] \in S_x \quad (20)$$

3 Applications and Framework

The objective behind fuzzy solid modeling is to permit the user to specify his or her qualitative ideas of vague geometric design, to obtain an approximated representation

of the desired form and to carry out tasks with the model while retaining its fuzzy characteristics.

3.1 Descriptive Modeling Environment

Based on previous issues, we designed a solid modeler prototype. This is a CSG based elemental environment where the primitive entities are defined in a descriptive way that uses a formal language to depict the geometric entity, as shown in Figure 11.

The fuzzy entities description interface has a tree (left) that shows all fuzzy geometric descriptors defined as input variables of the fuzzy system described previously as items of the tree and its partitions inside of it as shown in Figure 12, the descriptors are grouped in the tree to facilitate its location (shape, scale, deformation). The combination of the descriptor with the corresponding linguistic partition, constitutes the glossary of terms of the formal language that describes the fuzzy geometric entity. The primitive is assembled by dragging the leaf node from the tree (moderately longish, extremely pinched, ...) to the canvas on the right. While dragging, the primitive that is displayed acquires this description and adds it to the bottom list, which shows the textual description of the shape that is defined. The primitive visualization in the viewfinder is shown in a cyclic animation that scans the space of parameters that correspond to the description. Every shape showed by the animated sequence corresponds to a crisp entity that belong to the fuzzy entity.

Each description added (see “moderately square” in the figure) is listed in the interface next to the range that covers this linguistic label ([0.0-1.0]). The interface also shows the crisp value of the input parameter ($x=0.08$) that characterizes the crisp shape currently displayed, as well as the membership degree of the shape to

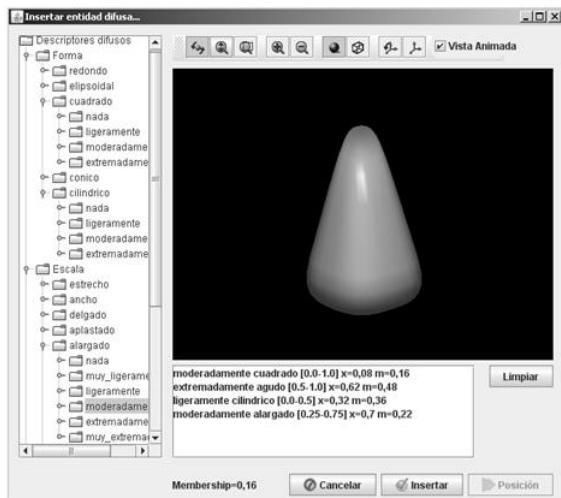


Fig. 11 Descriptive interface of fuzzy entity



Fig. 12 Tree of fuzzy geometrical descriptors

the descriptors ($m=0.16$). In addition, the membership degree of the rendered crisp shape is showed on the interface (Membership=0.16).

When the fuzzy primitive description is concluded, it is inserted in the work space to be combined with other primitives using boolean operations to obtain a more complex fuzzy shape, as shown in Figure 13. The inserted fuzzy entities are included in a tree (left), where it is possible to apply a rigid transformation and can be combined later by means of boolean operations. After executed the boolean operation between two fuzzy entities, the resultant fuzzy solid is displayed in a cyclical animation with the goal of illustrates the set of crisp solids that belong to the fuzzy solid model.

From a practical perspective the fuzzy solid models possess two fundamental qualities, the first one is that they are generated from a set of shape linguistic descriptors, given by a defined formal language that breaks with the traditional way of description that is in terms of exact geometric entities; the second one is that its

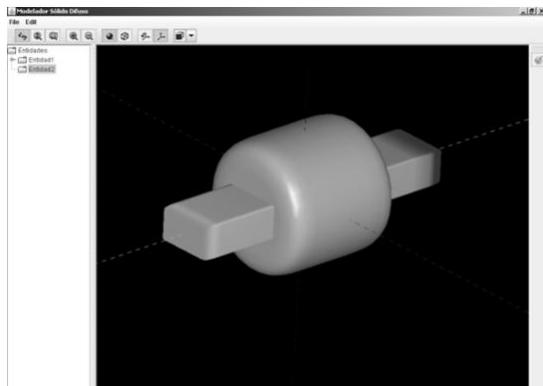


Fig. 13 CSG environment of fuzzy solid modeling

representation captures the geometric ambiguity immersed in the act of conceptualizing mentally an idealization of the space. So, we propose a new paradigm to depict solid geometric shapes in a more intuitive climate and close to the human thought, which captures a portion of the knowledge that the classic solid models did not perceive, or were not able to capture because this portion of the knowledge remained in the mind of the designer without being part of the model.

The goal of the fuzzy solid modeling is to make use of these qualities to facilitate the tasks of the designer in the early stage, thinking about how to express creative ideas in a rapid way, without the expressive ballast of the geometric precision and additionally to make use of this uncertainty to execute some tasks on the model.

The most basic use for the geometric vagueness derived from this model is to provide shape exploration mechanisms for selection and comparison by the designer, as well as automatic search of shapes within the implicit space designed by means of optimization methods. The approach can also be useful for tasks that make use of qualitative data and geometric uncertainty. Some examples are:

- The description of shapes using natural language for geometric conceptualization.
- The indexing and retrieval of geometric shapes using fuzzy criteria.
- The search of optimal shapes under any criterion.
- Classification and automatic recognition of shapes.
- Vague geometric reasoning.
- Modeling methods for non-experts.
- The manipulation of imprecision in measurement or calculation (scan 3D).
- The inclusion of esthetic subjective factors in the geometric design of products.

4 Conclusion and Future Work

By means of a fuzzy approach, a set of subjective terms with natural language to describe geometric shapes are associated with the parameters that control the

geometry of the model. To describe a shape qualitatively is equivalent to defining a family of similar shapes whose membership can be quantified, providing a space of alternatives. A fuzzy geometric model is a set of exact models. The implicit fuzzy solid model permits operation of the fuzzy geometric models in a homogeneous way, propagating the geometric uncertainty to more complex solids. The proposed modeler's prototype allows the depiction of geometric shapes in a descriptive environment that provides a favorable geometric conceptualization climate, as well as the execution of tasks that make use of the imprecision of the model. The proposal must be improved in ways such as: conceiving feedback mechanisms from the knowledge base with the users' perceptive opinion; establishing a glossary of terms to describe shapes at higher abstraction levels; and implementing solutions for the aforementioned tasks that improve the current performance [2].

Acknowledgements. This work has been partially funded by the Spanish Ministry of Science and Technology and EU FEDER funds (project TIN2007-67474-C03-02). Special thanks to AUIP.

References

1. Allen, J.F.: Maintaining knowledge about temporal intervals. *Communications of the ACM* 26, 832–843 (1983)
2. Barr, A.H.: Superquadrics and angle-preserving transformations. *IEEE Computer Graphics and Appl.* 1, 11–23 (1981)
3. Barr, A.H.: Global and Local Deformations of Solid Primitives. *Computer Graphics* 18, 21–30 (1984)
4. Batchelor, B.G.: Intelligent image processing in prolog. Springer, Heidelberg (1981)
5. Bierdman, I.: Human image understanding and: Recent research and a theory. *Computer Vision, Graphics and Image Processing* 32, 29–73 (1985)
6. Boothby, W.: An Introduction to Differentiable Manifolds and Riemannian Geometry, 2nd edn. Academic Press, San Diego (1986)
7. Clark, T., Larson, J.: Fuzzy geometry. *Applying Fuzzy Mathematics to Formal Models in Comparative Politics*, 65–80 (2008)
8. Company, P., Piquer, A., Contero, M.: On the evolution of geometrical reconstruction as a core technology to sketch-based modeling. *Computer-Aided Design* 32, 851–866 (2000)
9. Desmet, P., Tax, S.: Designing a mobile telecommunication device with an emotional surplus value: a case study. *Design Studies* (1998)
10. Fitzhorn, P.: Engineering Design is a Computable Function. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 8, 35–44 (1994)
11. Gandhi, A.R., Myklebust, A.: A natural language approach to feature based modeling. In: *Proc. Conf. on CAD and Computer Graphics*. International Academic Publishers (1989)
12. Gursoz, E.L., Choi, Y., Prinz, F.B.: Vertex-based representation of non-manifold boundaries. In: *Geometric Modeling for Product Engineering Applications of Artificial Intelligence*, pp. 107–130 (1990)
13. Hernandez, D.: Qualitative representation of spatial knowledge. Springer, Berlin (1994)
14. Horvath, I., Ruzak, Z.: Vague modeling for conceptual design. In: *Proceedings of TMCE 2000*, Delft University of Technology, pp. 131–144 (2000)

15. Knoop, W., Pham, B.: Towards computer suported design for aesthetics. In: Proceedings of the 31st ISATA Conference, pp. 403–412 (1998)
16. Martin, R.: Modelling inexact shapes with fuzzy sets. In: Proceedings CSG 1994, pp. 1–26 (1994)
17. Olsen, L., Samavati, F.F., Sousa, M.C., Jorge, J.A.: Sketch based user interfaces have been used to describe 3D models. Computers and Graphics-Elsevier (2009)
18. Pham, B.: Fuzzy logic application in computer aided design. Fuzzy System Design: Social and Engineering (1998)
19. Pham, B.: Design for aesthetics: Interactions of design variables and aesthetic properties. In: Proceedings of Annual Symposium Electronic Imaging 1999 (1999)
20. Pham, B.: Shape specification in design using fuzzy logic. Knowledge-Based Intelligent Information Engineering Systems (1999)
21. Pham, B., Zhang, J.: A fuzzy shape specification system to support design for aesthetics. Soft Computing in Measurement and Information Acquisition (2000)
22. Pham, B.L.: Fuzzy solid modelling. In: Proceedings Tools and Methods of Competitive Engineering (TCME 2002), pp. 273–286 (2002)
23. Qin, S.F., Wright, D.K., Jordanov, I.N.: From on-line sketching to 2D and 3D geometry: A fuzzy knowledge based system. Computer-Aided Design 32, 851–866 (2000)
24. Rosenfeld, A.: Fuzzy geometry: An updated overview. Information Sciences, 110–127 (1998)
25. Rossignac, J.: Structured Topological Complexes: A feature-based API for non-manifold topologies. In: Proceedings of the ACM Symposium on Solid Modeing 1997, pp. 1–9. ACM Press, New York (1997)
26. Torres, J.C., Clares, B.: Graphics objects: A mathematical abstract model for computer graphics. Computer Graphics Forum 12, 311–327 (1993)
27. Wang, L., Shen, W.: Collaborative conceptual design-state of the art and future trends. Computer-Aided Design 34, 981–996 (2002)
28. Weiler, K.: Non-Manifold Geometric Boundary Modeling. In: ACM Siggraph, Tutorial on Advanced Solid Modeling, Anaheim, California (1987)
29. Wu, K., Levine, M.D.: Parametric geons: A discrete set of shapes with parameterised attributes. In: Proceed. SPIE Inter. Symposium on Optical Engineering and Photonics in Aerospace Sensing: Visual Info. Process., pp. 14–26 (1994)
30. Yamaguchi, Y., Nakamura, H.: Probabilistic solid modelling: A new approach for handling uncertain shapes. Geometric Modeling for Product Realisation, 95–108 (1992)
31. Zadeh, L.: Fuzzy sets. Information Control 8, 338–353 (1965)
32. Zhang, J., Pham, B.L., Chen, P.: Modelling and Indexing of Fuzzy Complex Shapes. In: Proceedings 6th IFIP Working Conference on Visual Database System VDB6, Brisbane, Queensland (2002)

A GIS Platform for Automatic Navigation into Georeferenced Scenes Using GIS Scene Explorer (GIS-SE)

Christos P. Yiakoumettis, Georgios Bardis, Georgios Miaoulis,
Dimitri Plemenos, and Djamchid Ghazanfarpour

Abstract. During the last decade the rapid evolution of GIS systems has led to enhanced accessibility and increased availability. The traditional content has been enriched by multimedia information and three dimensional models. Virtual copies of the real world have been created, widely known to the average user, yet the information retrieval and the efficient exploration of 3D large scale scenes are still pose serious problems. In this paper, a GIS platform for automatic navigation is presented, using a prototype application, entitled “GIS-Scene-Explorer” or GIS-SE, addressing these problems. Our platform is based on an existing GIS platform. Google Earth is used for this purpose. The prototype is a standalone application, adopting the Google Earth API to retrieve and visualize data. It takes advantage of both public remote database of the GIS system and a custom, collaborative database. Three dimensional models of buildings and multimedia data are stored to the collaborative database, operating under a server – client model architecture.

Keywords: automatic navigation, virtual globe, GIS systems, scene exploration, virtual tour, scene understanding, visualisation, public participation, city modelling, urban planning.

1 Introduction

The first attempts and ideas of GIS platforms were introduced in the early of 1980s, but the technology limitations of that time shrunk their popularity. After decades commercial GIS platforms with real geo-referenced scenes have been released. These

Christos P. Yiakoumettis · Djamchid Ghazanfarpour
Université de Limoges, Laboratoire XLIM, CNRS, UMR 6172
83, rue d’Isle, 87000 Limoges France
e-mail: {djamchid.ghazanfarpour@unilim.fr}

Christos P. Yiakoumettis · Georgios Bardis · Georgios Miaoulis
Technological Education Institute of Athens
Department of Informatics
Ag.Spyridonos St., 122 10 Egaleo, Greece
Tel.: (+30) 2 10 53 85 826; Fax: (+30) 2 10 59 10 975
e-mail: {chgiak@teiath.gr, gbardis@teiath.gr, gmiaoul@teiath.gr}

applications are known as *virtual globes*. Most of them offer a three dimensional environment, a representation of the real world's surface. For the reconstruction process aerial and satellite photographs were used (Zhang, et al., 2007). Dedicated databases are used to store the data. Data are calibrated on a referred positioning system. In most cases the global positng system is used for that purpose. The data include hypertext, photographs three dimensional models and other georeferenced data. Commercial applications have been released by Google (Google Earth), NASA (NASA World Wind), Microsoft (Virtual Earth 3D), open source community and more, comprising more than simple mapping applications since they represent a 3D morphological model of the surface of the earth. Beyond the commercial approaches there are plenty others from open source community. They include information about the streets, points of interest, 3D models of buildings and multimedia content (Jones, 2007). Each one is using its own two or three dimennsional environment for visualizing the data in the form of a browser.

Virtual Globes applications support a 3D or 3D graphical interface. The user can freely explore the scene and "fly" among the obstacles. However the navigation on the scene can only be manually accomplished by keyboard and mouse. On the other hand the size and the complexity of the whole scene make the task of manual navigation and the insight gained by it generally inadequate. Moreover the user's incomplete knowledge of the scene increases the inefficient of manually exploration of 3D scenes.

A GIS platform for automatic navigation into a 3D georeferened scene will be presented. All required data of the 3D scene are stored in a collaborative database. The basic architecture supports multiple databases and it is based on an existing GIS platform. It may also serve as the platform for future features, for example planning personalized paths of automatic navigation into scenes, considering the user's preferences, profile and requests.

2 Virtual Globes

2.1 Simple Two Dimensional GIS System Architecture

One of the first virtual globes platforms, that became widely popular, was released By Google in 2004. It was Google Earth. A three dimensional environment is offered, that allows user to browse, navigate and explore 3D model of the scene. It is based on server-client GIS system architecture. A typical web browser is used on the user's side. The typical scenario is initiated by the client requesting data based on user's position in the virtual scene. In response to the request, the data is placed on the server side in a GIS database and subsequently the retrieved data are visualized on client's side with the aid of a specialized application. Initial releases offered a two dimensional environment (Wikipedia, Google Maps), (Wikipedia, Virtual Globes). Google Earth started publishing static maps, evolved to static web mapping applications and then to GIS distributed systems (Sripaisal & Pujari). The basic functionality of a server-client GIS system is presented in figure 1. This block diagram presents the functionality of Google maps, a two dimensional GIS platform.

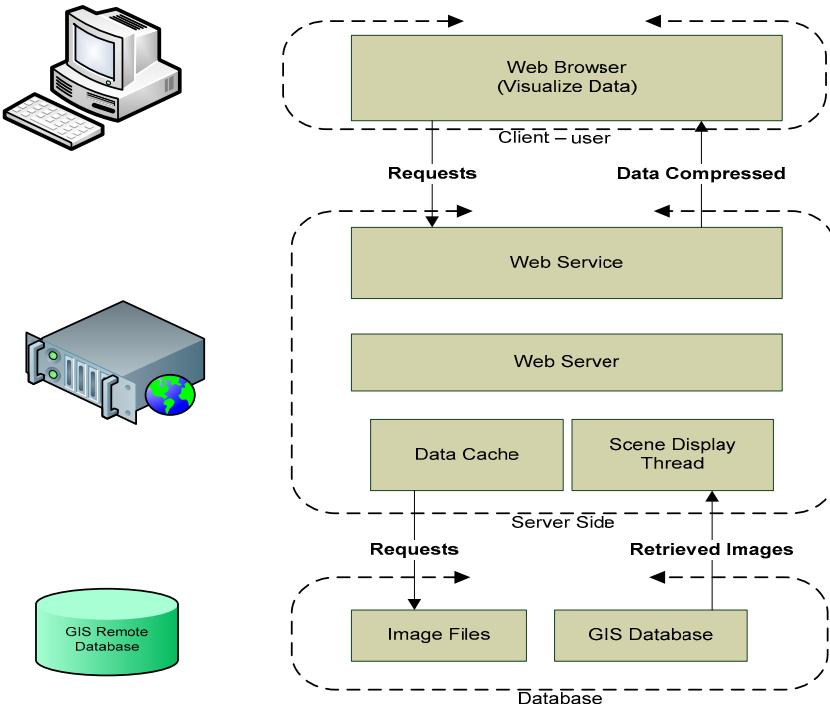


Fig. 1 Block Diagram of two dimensional GIS System

Requests from the users were submitted to the server. Communication between client and server was accomplished by using AJAX and XML technology. The user was requesting data depending on its position (longitude, latitude and altitude). The requested data were mined either from a temporary cache memory or by the georeferenced database. The retrieved information was sent back to the user's side in compress format and it was visualized on its side. The web server was used for the communication between the client's visualizing application and the georeferenced database.

Satellite or aerial photographs have been used to provide the necessary database content and the global positioning system has been used to calibrate the images (figures 1 – 2).

2.2 The Evolution of GIS Architecture

Google maps became known to general public. The quality of the services and the amount of information were increased by the time.. Initially, additional information was added to the database based on two dimensional aerial photographs. The names of the streets and the roads were the first data that were added to the

database. Afterwards, metadata such as text, external links, hypertext, and place-marks enriched the database.

Despite this evolution, the fundamental architecture has remained practically the same (figure 2). All the data were imported to databases and calibrated over the same geographical positioning system.

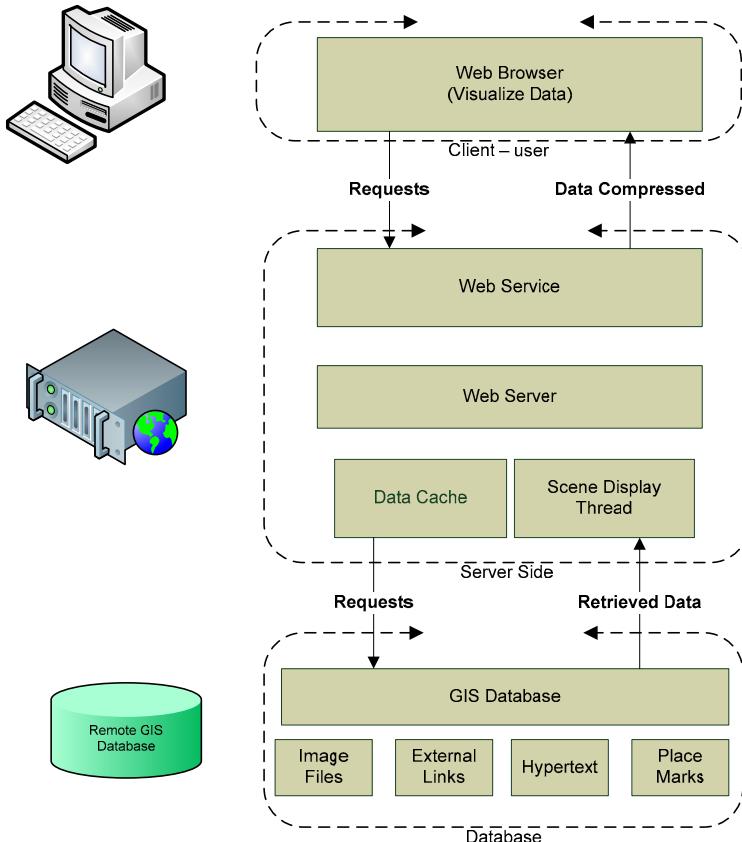


Fig. 2 Block Diagram of improved GIS system

The data repository of the system is comprised by four different resources, each of them potentially forming a separate database. All databases have to use the same geographic positioning system. This architecture allows the system to be very flexible and easily distributed to the extent that, for example, each database may be hosted by a different provider. Moreover, there is no limit on the number of physical databases that may constitute each of the aforementioned resources.

Beyond the two dimensional platforms, new browsers were presented with 3D exploration capabilities. Apart of images and hypertext, information about the morphology of the ground was added. In addition 3D models of buildings were enriched the database (figure 3). Initial models of buildings were limited and only specific cities were modeled.

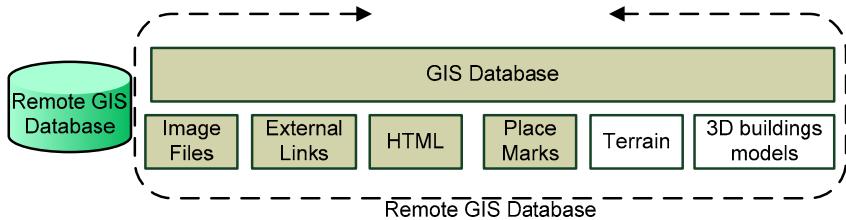


Fig. 3

3D models were designed and built in a 3D object description language called COLLABorative Design Activity (COLLADA). It has a language with XML schema and it has been adapted by many software and hardware developing companies.

On the user's side a standalone application or web browser with visualizing capabilities of COLLADA is required. Components have been developed by distributors for this purpose using OpenGL or DirectX graphics languages.

2.3 Recent GIS Platforms

Recent years many databases with georeferenced content have been released. The popularity of such systems and the advancements in computer technology have made possible the application of three dimensional environment on GIS systems, based on the above architecture. Information about the altitude and the morphology of the ground has been added to the database. In addition, 3D models of buildings are also being gradually added (figure 3). The goal is to create a virtual world that simulates the real one. There is no limit on the data that they can be added. There are GIS databases that are freely available for public use, whereas some others are not.

Data are requested based on user's position and preferences. Users can filter the data using the GIS platform interface. Data might be organized in dedicated databases. The user's application is presenting the data organized in layers. Each layer contains data from a different database. When all the data are being presented, a synthesized image is been created from the projection of the layers onto the same geographic position system. The result is a synthesized three dimensional scene.

The latest versions of user's application have the ability to cache the data, or even to load external data stored on local user's computer. The local data give the ability to create and use a custom local database. The user's client can be used as viewer or browser, also visualizing the local data.

3 State of the Art

In the field of geo-referenced and navigation systems, several methodologies have been introduced aiming to solve individual problems.

3.1 *Geo-referenced Systems*

One well-known approach is using a topic that represents the context of the image in order to classify geo-referenced images. The topic produced by a probabilistic latent semantic analysis (pLSA), a well-known technique in the text-mining community. Recently vocabulary was extended to describe, in a compact way, the contents of an image database (Bosch et al., 2006). The input of the probabilistic latent semantic analysis is a set of images and their location. The analysis classifies the images (Cristani et al., 2008).

In the field of geo-referenced databases, there are many distributed systems that have been developed. Most of the approaches are using a standard database schema with some extra geographical data for each object. On the other hand, the first implementations of spatial query languages were in 1980's. Borrmann Andre and Rank Ernst have developed a 3D spatial query language for building information models. The language is based on directional relationships among the buildings. The relationship between two buildings was calculated by the euclidean distance of the centers of the objects and the relative position of them, such as *northOf*, *southOf*, *eastOf*, *westOf*, *above* and *below* (Borrmann & Rank, 2008). Recently, the vocabulary of the language was extented and topological relationships were added such as *within*, *contain*, *touch*, *overlap*, *disjoint* and *equal* (Borrmann & Rank, 2009).

3.2 *Generation of 3D City Models*

There are several approaches on 3D city models generation. It can be accomplished eigher by reconstruction or by generation process based on rules.



Fig. 4 3D Large Scale Scene of 17.362 buildings (Larive & Gaidrat, 2006)

According to the first method, an efficient approach has been implemented using photogrammetry. A set of sensors are scanning the building and the scene (Koutsoudis, et al, 2007). The representation of the model is very accurate but experts, special equipment and heavy computational power are required.

Another approach is based on LIDAR (Light Detection And Ranging) (Lan, et al, 2009). It is an optical remote sensing technology that measures properties of scattered light to find range and/or other information of a distant target. The prevalent method to determine distance to an object or surface is to use laser pulses.

On the other hand the 3D model can be generated by an engine. A declarative engine for generating a 3D large scale scene was introduced by Larive & Gaidrat (Larive & Gaidrat, 2006) – (figure 4).

In addition several commercial applications have been presented in 3D reconstruction of georeferenced scenes. They can generate the model based on rules and on image recognition of satellite and aerial photographs.

3.3 Scene Understanding and Exploration

Several approaches have been implemented on scene understanding and exploration. The exploration of a virtual environment on behalf of the users can be classified in three distinct categories: (a) navigation, presenting data from the surface of the world, (b) presentation of an object or a path and (c) presentation of an object or item from specific views in order to reveal as much information as possible (Tab et al., 2001).

The most interesting methodologies are based on the last approach. Scene exploration can take place in two steps. In the first step, the best points of view are calculated and then in the second step, the path of the camera is designed based on the best views of the previous step.

An efficient approach for the calculation of the best view of a 3D scene has been presented by Sokolov and Plemenos. It is based on the coverage of the surface of the scene (Sokolov & Plemenos, 2005), (Sokolov & Plemenos, 2007), (Tab et al., 2001). This approach is considering the complexity and the total number of surfaces of the scene. Each view is evaluated individually and then the best views are selected for the trajectory of the camera.

In addition, Alcorer presented in 2003 a methodology for quality measurement of a certain view of a scene. It was based on information that can be extracted from a specific aspect of view of an object (Alcorer, 2003).

Another approach is trying to optimize the area of the visible surface comparing to the sum of the area of the surface of the whole scene (Sokolov & Plemenos, 2005), (Sokolov & Plemenos, 2006), (Sokolov & Plemenos, 2007), (Tab et al., 2001).



Fig. 5 Initial urban scene

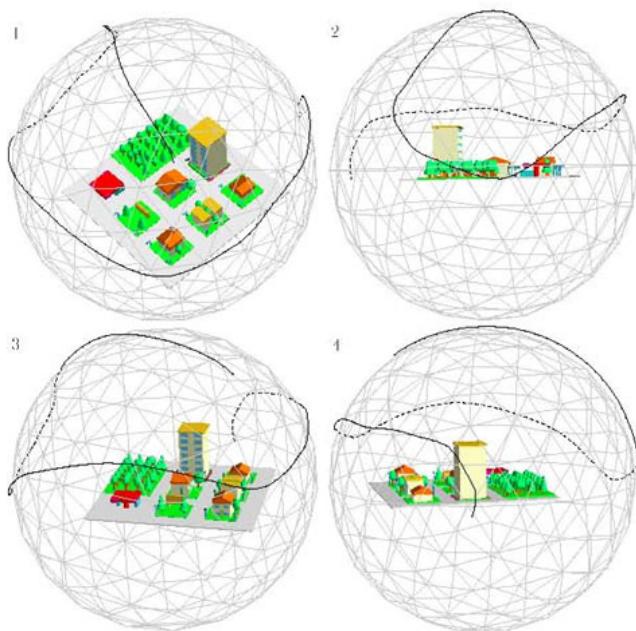


Fig. 6 Camera trajectory of an urban scene (Sokolov & Plemenos)

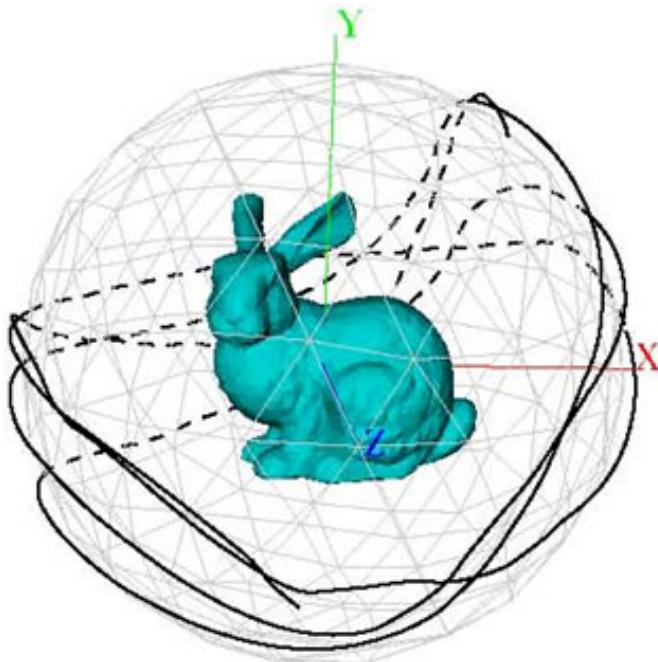


Fig. 7 Camera trajectory of rabbit (Sokolov & Plemenos)

Each object of the scene, either 2D or 3D can be presented from a different point of view. The angle of viewing an object in a three dimensional space requires five variables to be defined. The first three were used to define the position of the camera on an orthogonal 3D coordinate system and the other two were used for the horizontal and vertical orientation of the angle of view. Each point of view is evaluated individually. All possible points of views have been marked on the figures, where the points of the net of a virtual sphere are shown (Sokolov & Plemenos, 2007).

In the figure 5 the whole scene is presented. The best points were being considered for the trajectory of the camera as it was presented on the figure 6 from different aspects of view. In the figure 7 another example is presented using the best viewing algorith of Sokolov and Plemenos.

Moreover, Mackinlay introduced another approach of path planning and navigating in a 3D scene. It is based on points of interests (Mackinlay et al., 1990), (Ruger et al., 1996), (Wolf, 2000). The path is planned considering the user's desired points of interest. The path has to be calculated based on a set of parameters. Each parameter is defined by a weight or a degree of interest (Lucas & Desmontils, 2007). Measuring the various parameters and their weight, it is possible to optimize the information that can be gained following the designed path. The parameters may be pre-defined by the system or defined and evaluated by the

user. The evaluation can be done in several ways. A weight value or a degree of interest can define each point of visit, representing the importance of it, compared to the others (Bonnefoi & Plemenos, 2002).

In addition, methodologies of moving the camera on a virtual sphere have been published. In these approaches the object can be watched closer, either by user's intervention or by the system automatically (Tab et al., 2001).

Finally, the last approach is inspired by imagination and, in particular, the answer to the question: "how would it be if we could see the world from the eyes of the interesting object?" (avatars).

4 Proposing a GIS Platform for Automatic Navigation into a 3D Scene

4.1 Motivation

The massive use of simulators and virtual globe applications makes necessary the design and develop of an intelligent platform for evaluation and automatic navigation into the scene.

The complexity of the scene, the amount of information and the lack of knowledge of the surrounding makes difficult or impossible to navigate efficiently into a 3D large scale scene. Applying efficient techniques and methodologies that were presented in previous sections, our proposed intelligent system can be achieved. In the figure 4.2.1 the main architecture of our purposed platform is presented.

4.2 Functionality of the Proposed Intelligent System

Considering the above methodologies and approaches, it is possible to design *an intelligent platform that can evaluate a number of parameters, design a path and automatic navigate into a three dimensional environment presenting all the interesting points and important information based on user's preferences*.

The proposing GIS platform has to offer the following functionalities:

- Ability to extract knowledge from the scene. The features of 3D models have to be extracted such as location, shape, size, volume, texture, etc.).
- Support of multiple databases
- Automatic navigation into the scene based on user's preferences and criteria

In order to design such a system, focused on a 3D large scale space, it is necessary to be aware of the virtual environment and the objects it contains. The altitude of the ground can be known. The system must be able to calculate the volume and the size of the obstacles (3D Models) in the environment. Their shape and their

type could be important. In addition there are cases that properties of models or specifications have to be considered in order to plan a flight. Moreover, unique specifications will help the system to decide and better present an obstacle.

All these data have to be built or retrieved from an existing GIS system. Additionally, information and data that might be used will be stored in a local collaborative database. When the system has all of the above data and knowledge, it will be able to choose the points or objects to present and potentially the ones to avoid. These choices will take place based on the user's criteria, preferences and subsequently, the path will be planned according to them.

In the final processing stage of the system, the solution will be presented using a three dimensional viewer or browser. It has to give the opportunity to the user to interact with the environment and maybe change the parameters of the plan during the flight on the virtual world. The system has to be able reconsider the path based on the user's new preferences or criteria.

4.3 System Architecture

The above functionality is supported by the proposed architecture presented herein. A prototype application has been developed, called “GIS-Scene-Explorer”, implementing this architecture. Among the available GIS systems, Google Earth was selected due to the advantage of availability of many supported tools like Google Building Maker and Sketchup. The platform has been developed upon the Windows operating system using the Google Earth component within a web browser. AJAX and XML technology was used to communicate between the client and the public database. The communication was accomplished using Google Earth API. The interaction and the control of the data were limited by Google

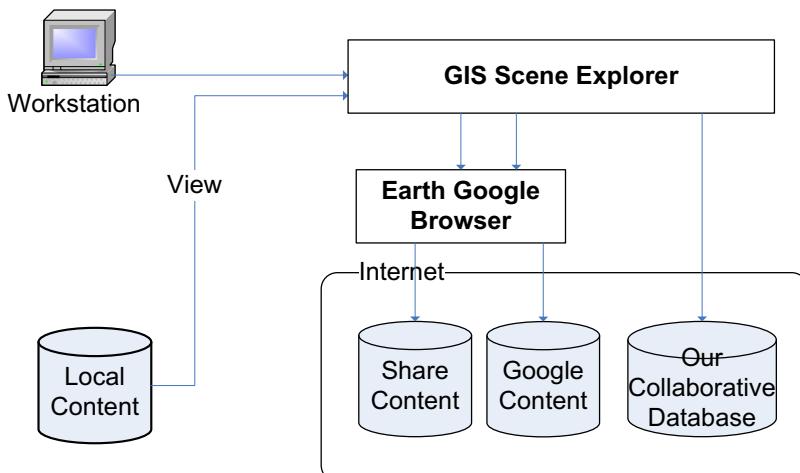


Fig. 8 Our GIS platform architecture

Earth API. Google Earth does not allow retrieval of the 3D models' information. The public GIS database was used only for retrieving data information for the morphology of the ground and for visualization purposes. Moreover, information about 3D models that exist on the public database is not useful. A collaborative GIS database was built for this purpose and all the 3D models of the scene were stored.

Google Earth component was used inside the GIS-Scene-Explorer only for visualization. The access and retrieval of the data was controlled by the GIS-Scene-Explorer. The morphology of the ground was retrieved by the public GIS database and the 3D models of buildings from the local, custom collaborative database (figure 8).

Multiple databases were used. With the aim of achieving our first requirement of extracting knowledge of the scene, all the required information and data of the whole scene must be stored on our own collaborative database. All the 3D models of the buildings were stored on it. Each model was containing information about its shape, size and location. Moreover the mapped images of texture of each individual side of the model were also contained on the database.

Google's public database was used for retrieving information about the morphology and texture of the ground (terrain). The morphology of the ground was not necessary to be known for the position of the buildings on the environment.

The GIS platform was able to use only the mined data for our own collaborative database. The rest data either form Google or from third party databases was used only for visualizing purposes on user's side.

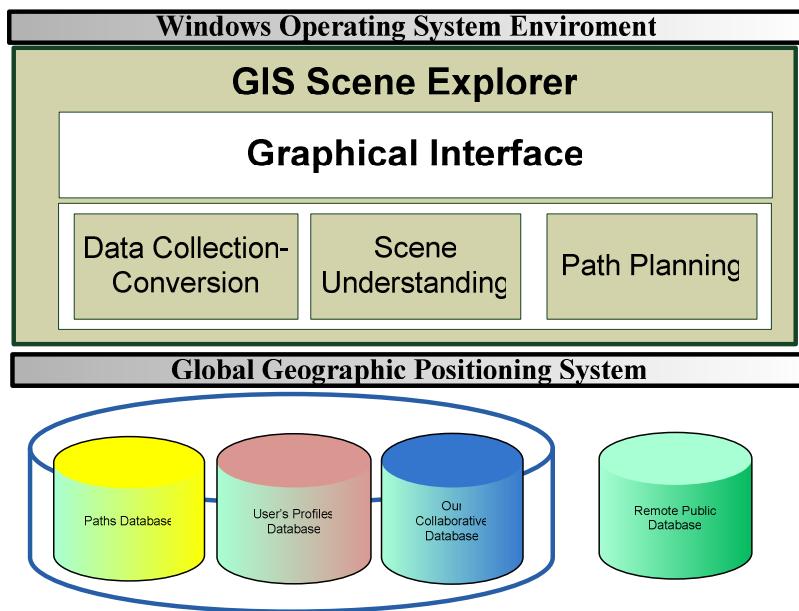


Fig. 9 Block Diagram

Regarding the knowledge of the models of the entire scene, it is possible to understand the scene and, afterwards, to plan a path, exploring the scene based on user's preferences or search criteria. This path is automatically calculated by the prototype application and visualized inside the GIS-Scene-Explorer using the Google Earth component (GIS viewer/browser) (figure 9).

GIS Scene Explorer (GIS-SE) has three modules. The first part of our approach was to design a collaborative database. The first module that it has already been completed is to gather data of 3D models. For this purpose a Collaborative web based platform was designed. 3D buildings were reconstructed using Google Building Maker. If it was required the models were improved by Google Sketchup. Apart of those models, data are retrieved from free online public databases. Likewise it was stored in our own collaborative database, in order fulfill the scene.

Afterwards the 3D models were mined from the collaborative database and they were analyzed. This module is still under development. Features of 3D models are extracted, such as location, size, shape and the mapped images of the texture of it. Considering these data, the whole scene is known. Only data from the collaborative database can lead to scene knowledge and understanding. Data that are retrieved from other databases can't be used on this step of the process.

The aim of our platform is the automatic navigation and exploration of the whole 3D scene, based on user's search criteria, user's profile and preferences.

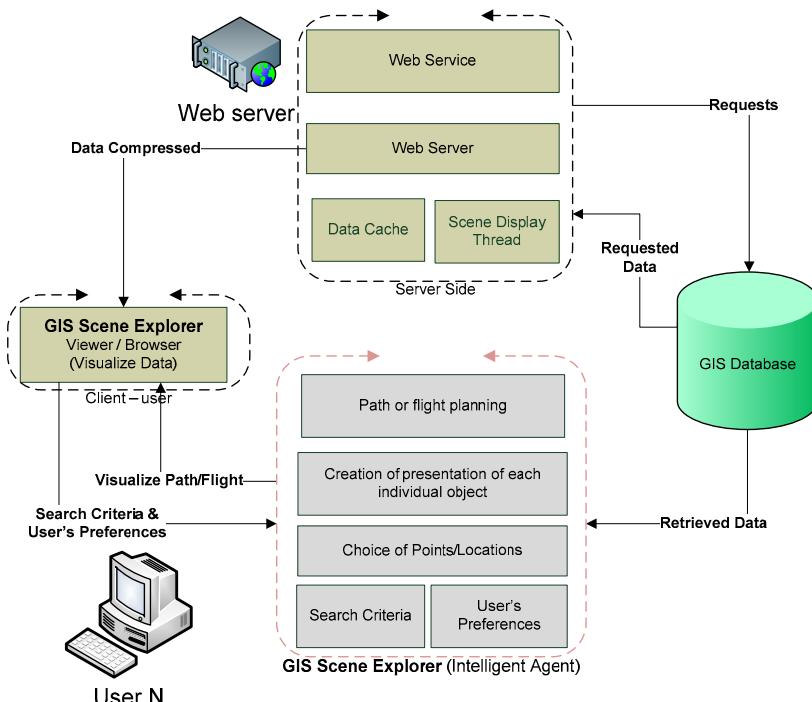


Fig. 10a Detailed Block Diagram of GIS-Scene-Explorer

The last module of our platform has to be built. It is responsible for the automatic navigation into the 3D scene.

A more detailed functional diagram of our platform is displayed below (figure 10). Each module is presented in details, analyzed in components.

Automatic navigation can take place into three steps. Taking in consideration the knowledge of the scene, the user's criteria, profile and preferences some points or buildings will be included or avoided from the calculated path. Secondly the presentation of each building model will be calculated using a best viewing algorithm. Finally the trajectory of the camera will be calculated in a way to include all the desired buildings and points of interests.

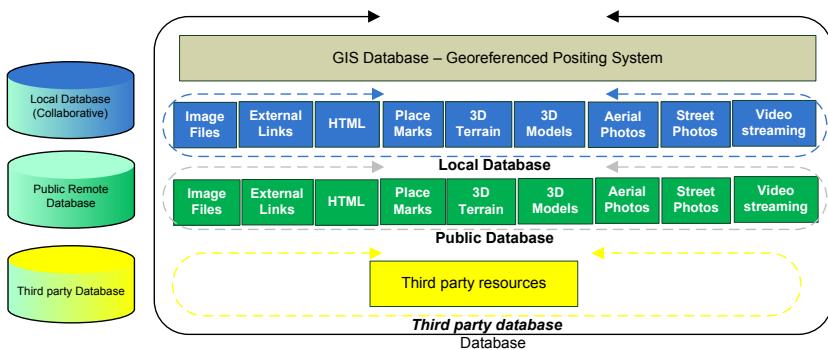


Fig. 10b Detailed Block Diagram of GIS-Scene-Explorer

3D models are mined from the collaborative one and terrain of the ground from Google Earth's one. Support for multiple third party databases have also taken into consideration. In addition it was used for the visualization of 3D environment of the scene using OpenGL-DirectX (figure 10). Apart of georeferenced data, user's profile, preferences and search criteria can stored on our database. The last is considered for the selection of the points of interests or buildings in the module of automatic navigation.

In a similar way, points or models can be avoided. Each model is presented individually based on its own specifications and shape. The best view is calculated for each model and finally the trajectory of the camera is created. The morphology of the ground does not affect the knowledge of the scene, as relative to the ground altitude can be used. The final trajectory – path of the camera contains only the interesting models, ensuring best viewing for each one of them. The path is transformed into the Google Earth's API language and is subsequently visualized by Google Earth component.

4.4 GIS Database

The existing GIS systems have two parts, the server side and the client side. In the client side there is an application that can visualize the data from the GIS

database. These applications are only viewers or browsers without any ability to interact with the virtual world. There is no ability to select an obstacle or measure it. The specifications of any object are not available. Alternatively, if all the models are stored on a local database, they can be accessible and easily measured.

In order to design a system that will be able to navigate, search and analyze 3D models, an alternative, fully accessible GIS database has to be used. For this purpose, a collaborative database was designed. The remote access of the database and the distribution of data were achieved through the http web service. A web site was built and the database was hosted in a SQL server. A Google earth plug-in component was used to visualize the data and create a 3D dimensional world with OpenGL-DirectX. Google has distributed in public an API for embedded Google Earth use. Using AJAX technology and the Google Earth API, the public database was filtered and the distributed database was loaded and visualized on client's application upon request.

The collaborative database is able to store and enhance the knowledge and the data of the GIS database. All three dimensional models and morphology of the ground is contained in the collaborative database. Afterwards, in the next module of path planning process, only the collaborative database is considered, ignoring the public one. All required information has to be on the database, i.e. the whole scene.

A team of student volunteers participated in the prototype evaluation. They constructed three dimensional models of buildings in the region of Athens, using Google Earth's tools. The three dimensional models had texture obtained from real satellite and aerial photographs. The models were eventually stored in the collaborative database.

4.5 Navigation to 3D Scene

Using the GIS-Scene-Explorer application, the user can navigate and freely “fly” among the 3D models of the scene (figure 4.5.1). Data are filtered from the collaborative or public GIS database and then they are visualized. The navigation process can easily be controlled using the interface of the Google Earth Plug-in window. The data exchange and communication between the client and the database are based on AJAX and XLM technology. In the next figure the GIS-Scene-Explorer is presented during the navigation of a scene with 3D models of buildings.

4.6 GIS Scene Explorer (GIS-SE)

A prototype “GIS Scene Explorer” application has been designed and implemented in order to fulfill our platform's objectives. The platform has taken advantage of an existing GIS platform. Google Earth plug in was used for visualizing purposes, providing a three dimensional georeferenced space. It was used inside our standalone application (figure 11).

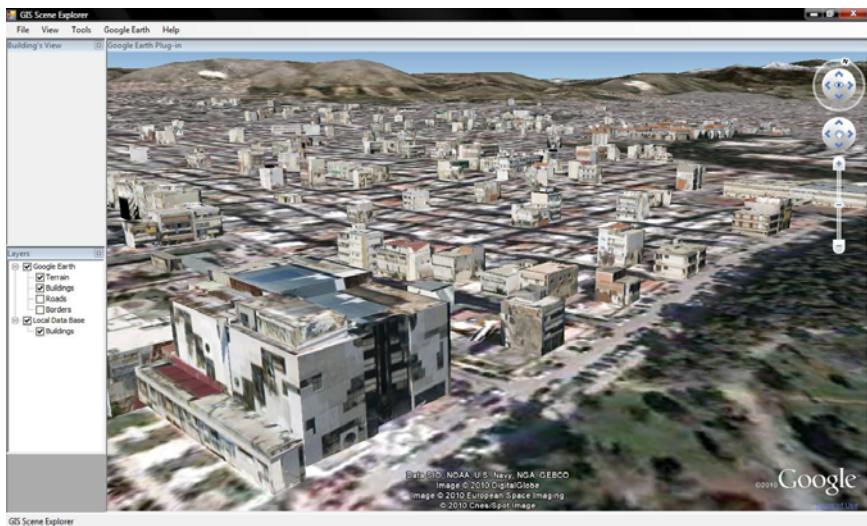


Fig. 11 Using GIS-Scene-Explorer for free navigation into georeferenced scenes containing 3D building models with texture

5 Conclusions

On this paper a prototype GIS platform was presented. Our approach was able to support multiple GIS databases. Data retrieved by our own collaborative databases were contained features of the 3D models. Knowledge of the scene was able to be extracted. The public GIS database of Google Earth was used for retrieving the morphology of the ground and a local collaborative one for the entire GIS scene.

The scene contains 3D building models with texture originating from satellite or aerial photographs. The user can fly and navigate freely in this three dimensional environment just like in Google Earth or any other GIS 3D application.

The purpose of GIS Scene Explorer is to provide a platform where all the content of the 3D environment is available. The public GIS systems do not allow gathering information about their content. They just use a visualizing module and allow the user to navigate into it freely. In our approach the whole scene is stored on a collaborative database. Each 3D model of the scene is accessible and editable by the platform itself. Moreover, all the benefits and advantages of the public GIS systems are still maintained. The user can still navigate freely into the 3D GIS scene using the mouse and the Google Earth interface.

The evolution of the proposed and implemented environment will allow the user to search into the scene based on criteria or preferences. The GIS Scene Explorer will automatically navigate the user into the scene, presenting only desired information, presented from optimal viewpoints, also taking into account the user's preferences.

6 Future Work

6.1 Semantic Knowledge

In this paper the architecture of a GIS platform. A prototype “GIS Scene Explorer” application was presented supports our approach. The content of the data was including only 3D models written in COLLADA. The database may also contain photographs, placemarks and any hypertext content that is supported by Geography Markup Language (GML), which has been adapted by the Google Earth. After the construction of a scene, a question arises: How we could navigate or fly inside the scene, watching all the interesting points while avoiding the rest of the data? The management of a large scale three dimensional scenes is one of the major problems of large city models. In addition to the information conveyed by the 3D models themselves, if semantic knowledge is added to the scene, the user could be guided by the system to see only the desired information of the scene.

6.2 Path or Flight Planning

This part of the process can be accomplished by an intelligent agent obtaining all required data for the process from the local database. Before the path planning takes place, the user’s profile containing his/her preferences must be known. The user will be able to describe, to choose or set criteria for the navigation. The criteria will suggest points of interest or points to avoid.

In the first part of the process each point of the path and the sequence of them will be calculated. Afterwards, for each point of the path, the optimal way of presenting the object will be calculated, based on its specifications and its type.

6.3 Scene – Path or Flight Viewer

This is the last part of the process. In this part the path calculated in the previous step is visualized. The role of the viewer module can be accomplished by the client of the GIS system. The user may either watch the path/flight without intervention or he/she may alter the initial plan. In this case the system will have to recalculate the path. Dynamic user intervention in the visualized trajectory increases the complexity of the task.

6.4 Possible Applications

Applications of city 3D modeling systems can be found in a variety of fields. Virtual tours can be achieved in places that one can or cannot visit. In the field of tourism, for example, a 3D city helps to develop and present a region to the rest of the world. Furthermore, simulations can be based on 3D city models, with respect to physical phenomena, evacuating processes, etc.

References

- [1] Alcorer, P.: On the selection of good views and its application to computer graphics (2003)
- [2] Bonnefoi, P.F., Plemenos, D.: Constraint satisfaction techniques for declarative scene modeling by hierarchical decomposition. In: 31A 2000 International Conference (2002)
- [3] Borrmann, A., Rank, E.: Specification and implementation of directional operators in a 3D spatial query language for building information models (2008)
- [4] Borrmann, A., Rank, E.: Topological analysis of 3D building models using a spatial query language (2009)
- [5] Bosch, A., Zisserman, A., Muoz, X.: Scene classification via plsa (2006)
- [6] Chen, C.: Information visualization and virtual environments. Springer, London (1999)
- [7] Cristani, M., Perina, A., Castellani, U., Murino, V.: Content Visualization and Management of Geolocated Image Databases (2008)
- [8] Jones, T.M.: Google's Geospatial Organizing Principle (2007)
- [9] Anestis, K., Fotis, A., Christodoulos, C.: On 3D reconstruction of the old city of Xanthi. A minimum budget approach to virtual touring based on photogrammetry. Journal of Cultural Heritage 8, 26–31 (2007)
- [10] Larive, M., Gaidrat, V.: Wall Grammar for Building Generation. In: Graphite 2006, Kuala Lumpur, Malaysia (September 2006)
- [11] Lucas, M., Desmontils, E.: Declarative modelers. Revue Internationale de CGAo et d' Infographice, 559-585 (2007)
- [12] Mackinlay, J., Card, S., Robertson, G.: Rapid Controlled Movement through a Virtual 3D Workspace (1990)
- [13] Ruger, M., Preim, B., Ritter, A.: Zoom Navigation Exploring Large Information and Application Spaces (1996)
- [14] Sokolov, D., Plemenos, D.: High level methods for scene exploration. Journal of Virtual Reality and Broadcasting 3 (2006)
- [15] Sokolov, D., Plemenos, D.: Intelligent scene display and exploration (2006)
- [16] Sokolov, D., Plemenos, D.: Viewpoint quality and scene understanding (2005)
- [17] Sokolov, D., Plemenos, D.: Virtual world explorations by using topological and semantic knowledge, pp. 173–185. Springer, Heidelberg (2007)
- [18] Sriphaisal, W., Pujari, A.K. (n.d.): A comparative Assessment of Giservices Architectures
- [19] Desney, T., Robertson, G., Mary, G.: Exploring 3D Navigation: Combining Speed-coupled Flying with Orbiting (2001)
- [20] Wikipedia (n.d.). Google Maps,
http://en.wikipedia.org/wiki/Google_maps
- [21] Wikipedia (n.d.). Virtual Globes,
http://en.wikipedia.org/wiki/Virtual_globes
- [22] Wolf, C.: Content based Image Retrieval using Interest Points and Texture Features, Technical Report, PRIP-TR-61 (2000)
- [23] Zhang, J., Gong, J., Lin, H., Wang, G., Huang, J., Zhu, J., et al.: Design and development of Distributed Virtual Geographic Environment system based on web services (2007)

A Level of Interaction Framework for Exploratory Learning with Characters in Virtual Environments

David Panzoli, Christopher Peters, Ian Dunwell,
Stéphane Sanchez, Panagiotis Petridis, Aristidis Protopsaltis,
Vincent Scesa, and Sara de Freitas

Abstract. This paper investigates a range of challenges faced in the design of a serious game aimed at teaching history *in situ*, through the use of an immersive, open virtual environment. In the context of this paper, such an environment is described as an exploratory, expansive virtual world within which a user may interact in a non-linear and situated fashion with the virtual characters that populate it. The main contribution of this paper consists of the introduction of the Levels of Interaction (LoI) framework, designed to assist in the creation of multiple forms of interaction between a user-driven avatar and synthetic characters. The LoI approach addresses the necessity for balancing computational efficiency with the need to provide believable and interactive virtual characters, allowing varying degrees of visual, interactive and behavioural fidelity. The *Roma Nova* project demonstrates a first implementation of the concept, showing in practice how the LoI are likely to foster more natural interactions between the player and the non-playing characters.

D. Panzoli · C. Peters · I. Dunwell · P. Petridis ·

A. Protopsaltis · S. de Freitas

Serious Games Institute,

Coventry University Technology Park, Cheetah Road, Coventry,
CV1 2TL, United Kingdom

e-mail: DPanzoli@cad.coventry.ac.uk, Sara.deFreitas@coventry.ac.uk

Stéphane Sanchez

IRIT-CNRS, UMR5505, Université de Toulouse,
2 rue du Doyen Gabriel Marty, 31042, Toulouse, France

e-mail: Stephane.Sanchez@univ-tlse1.fr

Vincent Scesa

Instinct[Maker], Toulouse, France

e-mail: v.scesa@instinct-maker.com

1 Introduction

The *Roma Nova* project aims at teaching history to 11 to 14 years old children, which corresponds to key stage 3 (KS3) of the English curriculum. *Roma Nova* is a serious game which enables exploratory learning [10] by immersing the learner/player inside a virtual environment where they learn different aspects of history through their interactions with a crowd of virtual yet authentic Roman characters. It is a single-player game, presented in English language.

The game is built upon *Rome Reborn* [12], the most high-fidelity model of Ancient Rome currently in existence, providing a 3D digital model which may be explored in real-time. *Rome Reborn* includes hundreds of buildings, 32 of which are highly detailed monuments reconstructed on the basis of reliable archaeological evidence, for example, as depicted in figure 1a. The rest of the 25 to 30 square kilometers model is filled with procedurally-generated buildings based on accurate historical knowledge (figure 1b).

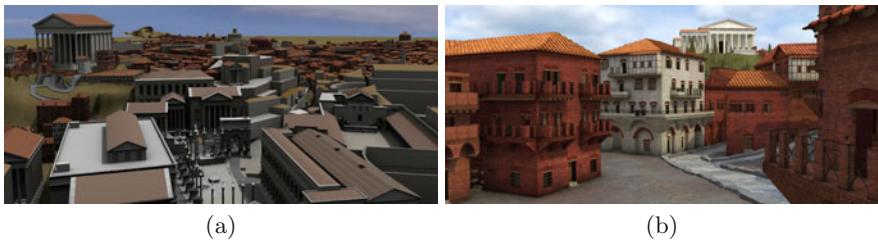


Fig. 1 The Rome Reborn project [36] is a 3D model of the A.D. 320 city of Rome. The model includes accurate models of monuments (a) as well as procedurally-generated areas (b).

The *Roma Nova* project addresses several challenges faced when seeking to design a serious game where the player is taught history by interacting with autonomous characters in a wide open environment. It builds on previous work at Coventry University [1] that demonstrates the potential of game techniques for cultural heritage experiences, outlining the problems encountered when integrating a substantial number of different state-of-the-art techniques. In [24], we have already suggested a step towards achieving this integration involves the design of a general model, to frame the interactions between the player and the characters, bringing together these different interaction modalities into a single multi-scale agent model.

The first critical aspect of the *Roma Nova* project considers the design of an intelligent tutoring system to explore issues related to pedagogic design, such as how desired learning outcomes may be broken down into different scenarios, how different classes of characters or personality may help learners explore different perspectives on events and ultimately how to assess the knowledge effectively learnt. The present paper focuses on the technical aspects of the game (see [25] for the pedagogical aspects), notably which techniques from behavioural animation or artificial life may be used to enable such a broad range of interactions with any character of the crowd, and how they can be integrated together into a single real-time agent controller, while still enabling many characters to be displayed simultaneously.

2 Related Work

A significant volume of literature exists around the study of projects aiming to populate a virtual environment with a crowd of characters. Crowds are desirable for a range of purposes, such as bringing immersion and presence to a historical place, accurately simulating the behaviour of human people, or application-driven scenarios, such as emergency evacuations [35]. The following projects are particularly worthy of mention since they focus upon simulating realistic and believable crowds.

The Pompeii project [21] “revives” the ancient city of Pompeii, buried during an eruption of the volcano Mount Vesuvius. One of the challenges of the project was to populate the 3D reconstruction of the city with a realistic crowd of Romans. To that end, the project places a particular attention on the diversity of the crowd (figure 2.a and .b). The virtual Romans are modelled using a variety of body, face and clothing models, in such a way that every character appears unique. The project also includes a differentiation between city areas and the subsequent impact on appearances, e.g. the variation between wealthy and poorer areas. Another interesting feature includes the implementation of “places of interest”, wherein certain salient city features can attract the attention of passers-by.

The Pennsylvania station project [34] aims to populate the historical reconstruction of the famous New York city railroad station by the same name. Although an emphasis is placed on the characters’ diversity, akin to the Pompeii project, Shao and Terzopoulos predominantly focus upon emulating the rich variety in behaviour of the characters (figure 2.c). The characters walk toward goals in the station based on individual perceptions of it. They can rest, sit on public benches, talk with each other, queue in front of ticket desks, and so on, so that they appear to have a purpose and intent. To further accentuate the realism, live events take place randomly, such as attracting the attention of passing characters.

Other projects consider the role of the human viewer in more detail when synthesising virtual crowds. The Metropolis project [22] investigates visual

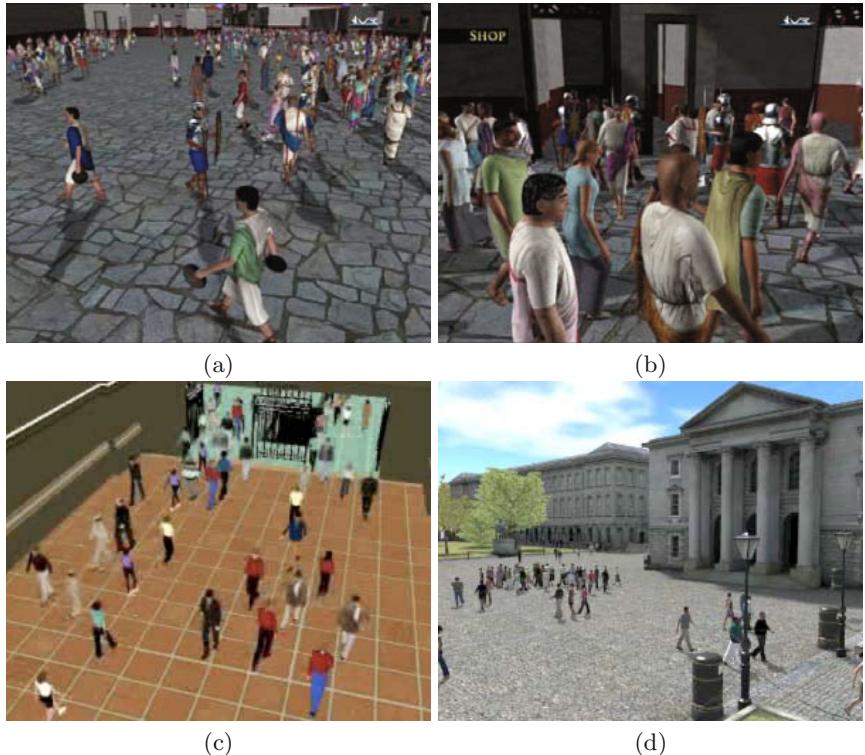


Fig. 2 (a-b) The Pompeii project aims to populate a virtual replica of the buried city of Pompeii, Italy. More than 4,000 virtual Romans populate the city, each unique in their combination of shape, size and colour. (c) The Pennsylvania station is populated with autonomous pedestrians that show different behaviours like queuing, resting, talking, etc. (d) The Metropolis project pays a particular attention to the role of user visual and auditory perception when interacting within the environment.

and auditory perception of crowds in city environments (figure 2d). It introduces an important methodology for evaluating crowd scenes based on viewer perceptions of realism. For example, at far distances from the camera, it may not be necessary to animate or display models at the same degree of detail as when closer, since the viewer may not be capable of noticing differences. Another aspect of realism investigated is viewer perception assessed with regard to the formation of groups [28].

Although the previous projects explore interesting aspects of a crowd of characters, they do not consider interactions between the player and the crowd. Commercial games are thus worthy of interest, since interactions between the user and automated ‘non-player’ characters (NPCs) have a critical impact on the game-play and the overall experience of the player. Owing

to significant progress in computer hardware over the few last years, recent games allow the player to wander freely in large virtual cities or populated areas. Although such games vary in nature and context, the approach used to manage the interactions between the player and the NPCs is often similar. Traditionally, the player is not allowed to interact with every character in the game. There is a distinction between characters that have been placed at a specific location for interaction and other characters that are simply bringing more life to the area.

A crowd of characters may enhance the perception of a coherent world and the player's feeling of being part of it can even be enhanced if the crowd is able to react to them. The following games introduce interesting crowd behaviours. *Assassin's Creed*, from Ubisoft focus on creating an immersive experience for game players. The crowd of characters the player is navigating through can be considered as a fully-fledged character in the game. Indeed, although the NPCs in the crowd attend their own business, regardless of the player, they are still likely to mind the player, and protest when the player collides with them. Similarly in *Grand Theft Auto*, from Rockstar Games, passing NPCs can adopt a specific behaviour depending on the player's actions or the surrounding atmosphere. For instance, they run away shouting when a gunfight takes place or defend themselves when attacked by the player. *Dead Rising*, from Capcom, uses the crowd as an element of game-play. The crowd of zombie NPCs flocks to converge on the player or circulate around a large area.

Different means are used to signify to the player the characters' ability for engaging in more detailed interaction. In *Fable*, from Lionhead studios, halos are used to highlight such characters (figure 3(a)). Additionally, when the player enters a range of proximity to the character, a sign appears above them to specify the kind of interaction expected. Although Blizzard's *World of Warcraft* is a massive multi-player game, mainly populated with human



Fig. 3 (a) In the game *Fable*, the interaction abilities are represented by a coloured glowing halo around the NPC. (b) In *World of Warcraft*, an exclamation mark is used similarly.

players, exclamation marks are used to identify non-playing characters responsible for allocating quests or trading with the player (figure 3b). This game also demonstrates that, although human interaction may be a predominant component of the game design, a need persists for artificially-driven characters to perform mundane or unrewarding tasks. Characters performing these tasks are central to creating a believable, immersive world; however, these games are typically designed to empower the player as a hero, and to achieve this goal for all human players, the artificial intelligence must serve as the ‘ordinary’ people whom players rise above, rescue or defeat. Therefore, even in massively multiplayer scenarios, a need persists for interactive, believable non-player characters.

In both exploratory and narrative-driven serious games, interaction between the player and artificial non-player characters can form a fundamental part of both game-play and instructional design. A substantial volume of research exists around the theme of populating virtual environments with perceptually and functionally realistic characters - the crowd modelling techniques described in Section 3.1, for example, commonly seek to implement realistic crowd behaviours to enhance immersion by imparting background characters with the necessary intelligence to form groups and navigate through highly populated virtual spaces. However, implementing an effective serious game or instructional environment within a virtual world requires the creation of characters that are not only visually perceived as behaving realistically on a large scale, but that are also capable of interacting with the user in a coherent and consistent manner, to maintain flow and immersion and thus effective learning transfer. For evidence of this relationship, see the works of Csikszentmihalyi and colleagues [6].

To enable and sustain this level of immersion and interactivity, we need to firstly define a framework capable of supporting such interactions and subsequently design a system which effectively implements these notions in a computationally and resource-efficient form. Our proposed approach integrates crowd modelling, intelligent characters and communicating agents into a composable and generalisable approach with potential application in a wide range of virtual environments and training simulations.

3 Levels of Interaction

In order to integrate the interactions between the player and the NPCs, we propose a novel framework called the Levels of Interaction (LoI). The LoI conceptually simplifies the interactions between the player and the NPCs. Graphically, the LoI can be represented as auras [11] of increasing complexity centred on the player’s avatar (figure 4) and based on a simple social space metric.

The first level aims to populate the area with an authentic crowd of characters. It provides a *living background* to facilitate the immersion of the player

in the environment. Characters evolving in closer surroundings of the player belong to the *interaction* level. These characters pay attention to the player and provide some basic interactions, comparatively to the next level of interaction. Finally, a character inside the *dialogue* level interacts with the player in a natural way, ultimately using speech recognition and synthesis.

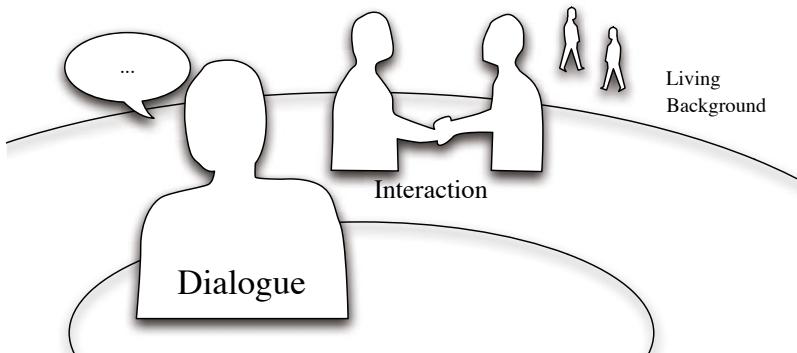


Fig. 4 The levels of interaction (LoI) technique allows different levels of complexity in the interactions between the player and the other characters, from an educational dialogue to a simple living background.

Characters are not assigned a specific level of interaction *a priori*. By default, they all belong to the background. But as the player moves and/or they move freely in the environment going about their business, they happen to get closer or away from the player and thus enter or exit the interaction or dialogue levels.

3.1 Living Background

The living background layer contributes predominantly to the believability of the environment. Kim and colleagues [16] utilise the term *believability* to define a two-layered model in an attempt to classify the various aspects of virtual environments of greatest concern when creating a real-world simulation. In this model, immersion plays a key role in creating a believable environment, in terms of the relationship between human sensory input and system output. The further human senses are exposed to external stimuli, such as background noise, Kim and colleagues argue, the lower the level of immersion and hence the less believable the environment. However, this is refuted to some extent by Robertson [32], who claims that “immersion should not be equated with the use of head-mounted displays: mental and emotional immersion does take place, independent of visual or perceptual immersion”, an opinion reinforced by Csikszentmihalyi and Kubey [6].

A simple out-of-place texture or inappropriate sound can have catastrophic effects for believability, as metrics of immersion such as the performance indications and cognitive surveys applied by Pausch and colleagues [26] demonstrate. With respect to the behaviour of avatars, this is similarly true – for example, if a character is perceived to be undertaking behaviour which is without context or motivation. The approach addresses these issues through not only the support of the living background level, but also by enabling the system to adapt to more sophisticated interaction requirements dynamically by elevating an interaction through the layers of the model. However, at the single crowd level, the believability challenge is addressed through two mechanisms. Firstly, by allowing the characters to adaptively evolve in an informed environment, they may respond and adapt to the behaviour of the player. Secondly, by endowing the characters with motivations and goals, the coherence of their behaviour may be assessed.

An informed or annotated environment contains semantic information, helping the characters to trigger the right behaviour regarding the context. Such information is useful for the general navigation of the crowd [9], defining routes, building entrances and associating areas with classes of characters to avoid incongruities – like a slave wandering in the *curia*, where the senators were meeting. Annotations may also be used for triggering specific behaviours related to specific places, like shops or street happenings. Although their behaviour is guided by these annotations, it is very important – for diversity's sake but also for believability – that the characters maintain their own autonomy. Artificial Life (AL) provides techniques for enhancing their adaptivity. Indeed, increasingly ambitious approaches use AL-driven agents to simulate the varied and complex behaviours one can observe in a crowd, however relying on lightweight and simple systems. Such techniques, mostly inspired by Reynolds flocks [31], use physically-based systems [13] or cellular automata [17] to let properties emerge instead of scripting them manually. AL-crowds have revealed the ability to model coherent flowing and emergence of lanes, jamming behaviours at constricted points (e.g. doorways) and formation of groups. Finally, adaptive characters by definition have the ability to tailor their behaviour to a given situation, such that it is always coherent with regard to their context. This makes the characters opportunistic, as they can change their objective(s), for example going to the temple, depending on their mood or interest – attending a street exhibition or listening to an orator haranguing the crowd.

To maintain this coherency in time, characters also need to be motivationally-driven. As such, they are guided by internal motivations including internal needs, duties or a diary, so that everyone of them seems to have a reason to be part of the scene. If the player follows a character, he will witness a citizen living a coherent life. Furthermore, every character has a role and a psycho-sociologic profile [19]. The first defines strong relations between each other – a husband and wife, a master and slave(s), a centurion and soldiers, a merchant and customers. The latter defines soft relations, that

mostly depend on the situation – two friends encountering and waving at each other, or stopping their activity to chat, mendicants begging for money as a rich person passes by.

Although from the player's point of view, the characters composing the background are merely able to walk or stop, getting closer will reveal in detail the interactions these characters are engaged in. This is explained in the next section, as part of the interaction level.

3.2 *Interactions*

Characters inside the interaction level are no longer part of the background since they provide a first degree of interaction with the player. This is revealed by a more complex behaviour providing a visual way to teach history, and the ability to dynamically respond to the player or engage them.

Inside the interaction level, the characters are rendered in more detail and animated with smoother motion captures in order to enhance their believability. Moreover, visually and historically more accurate motion captures can be used to depict scenes of craftsmanship, trading, art or rites, thus providing the learner with a first-hand perspective when learning history. Figure 5a shows for example how virtual scenes of acting or dancing, taking place in the Roman theatre of Aspendos, might promote a better understanding of the contextualised social role such places occupied in the ancient societies [20]. The same technique could be applied to provide accurate depictions of the know-how of a baker, of the meticulous craft-work of a blacksmith or the social rules involved in a transaction between a merchant and a customer.

The interactive context of the game makes the use of a single lengthy motion-capture inappropriate to represent a complex behaviour. Rather, such a behaviour ought to be considered as an adaptive series of actions that the



Fig. 5 (a) Accurate motion captures allow for learning transfer through interactive observation. (b) The VIBES architecture uses a classifier system to allow the automatic creation of adaptive complex behaviours from sequences of simple procedural actions. In this case, the character learns to cook a meal using basic ingredients and household appliances.

player would be able to interrupt dynamically. Agent technology provides a wide range of controllers for managing the complex behaviour of an animated character like a virtual human. ‘Hybrid’ controllers are well suited for this kind of task. They combine a set of basic behaviours to a database of knowledge representations within the same architecture, for the agent to be reactive enough to act naturally yet cognitive enough to act coherently. The ViBes architecture [33] (figure 5b) has relevant features. It is firstly a hierarchical controller, so that every complex behaviour is built from the organisation of lower-complexity behaviours operating recursively until the lowest level of complexity, which consist in basic moves that can easily be rendered by motion captures. Additionally, the ViBes architecture uses AL techniques – namely, a classifier system – to ensure the adaptive sequencing of each behaviour, in such a way that any behaviour, no matter how complex, can be seamlessly interrupted by another, without leading to an unknown and hence non-displayable situation. As a result, a Roman character processing a series of actions towards a motivated goal can be interrupted by another character and start a new interaction scheme, at the end of which it would be able to return to their occupation.

The interaction level also provides the first degree of integration between the player and the crowd, by allowing proactive and non-verbal interactions with the other characters (figure 4). For instance, passing characters stare at the player, so that he does not feel like a ghost in the crowd. They could also greet the player, if for example the player has met them in the past. More importantly, these interactions provide a way to initiate a dialogue or give the user the incentive to initiate a dialogue. Indeed, non-verbal communication, such as waving or gaze behaviours, can play an important role in the initiation of a verbal interaction [27], for example by attracting the player’s attention. The player would then be free to step towards the character and thus enter the dialogue level.

3.3 Dialogue

No matter how many characters are included in the scene, the dialogue level, which involves a close-up interaction with a character, contains only one at a time. Since more details of this character can be potentially seen, this in turn suggests the requirement for a highly detailed representation, including for example, facial expressions, gestures or body stance [5].

When a character enters the dialogue level, a dialogue is triggered with the player. At this level, the role and capabilities of the character are well described by research into Embodied Conversational Agents [4], or ECAs. ECAs are 3D (figure 6a) or cartoon-like (figure 6b) characters that attempt to demonstrate many of the same capabilities as humans in face-to-face conversation, including the ability to produce and respond to verbal and non-verbal communication. ECAs may be endowed with a multimodal interface

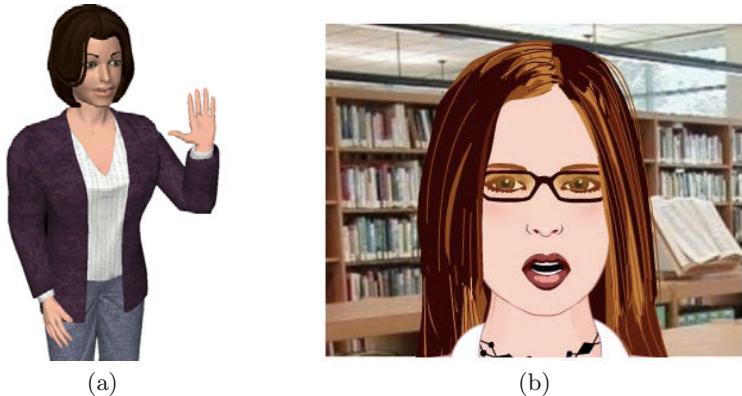


Fig. 6 Depiction of (a) Greta, an Embodied Conversational Agent [4], or ECA, and (b) Lillian the virtual librarian chatbot (©Daden Ltd).

that parallels their verbal output to other modalities, matching the emotional content of the dialogue or provide appropriate feedback. For example, the parameterised action representation [2] coincides specific expressions and gestures with speech, in order to bridge natural language and embodied action.

This level particularly supports situated, experiential learning. Therefore, the gaming aspect is largely supported by this level of interaction, and brought to the player via several quests, wherein each scenario is not explicitly defined, but rather built through the course of interactions with a variety of characters. By talking to different characters, the learner can interact with and reflect upon different aspects of the life in Rome and history and the differing roles and jobs of individuals within the society. As an example, the player may learn about military history while speaking to a praetorian or about diet when talking with a grocer.

These dialogic interactions are the very core of this history-teaching serious game, as known pedagogic strategies and techniques can be applied. Namely, the game could benefit from the experience of pedagogical ECAs in order to improve human learning performance. These specific ECAs are virtual instructors that autonomously train a human learner by applying appropriate pedagogical techniques during instruction [7] through scaffolded learning.

To summarise, the LoI framework proposes an elegant and convenient way to represent the types of interactions offered to the player through a single integrated system (figure 7). The next section presents a first implementation of the concept in the *Roma Nova* project, showing how technical issues have been overcome.

Level of Interaction	Living background	Interaction	Dialogue
Distance	Far	Mid-range	Near
LoD	Low	Medium	High
LoS	Low	Medium	High
Granularity	Crowd	Group	Individual
Agent properties	Reactive, Motivational, Opportunistic	Pro-active, Adaptive	Intelligent, Social, Conversational
Example capabilities	Navigation	Complex behaviour, Social attention	Facial expressions, Speech synthesis, Dialogue management
Computational resources	Low	High	Low

Fig. 7 This table summarises the different features of the characters, for each level of interaction. Note that in spite of higher realistic model and behaviour, the dialogue level is not the most costly in terms of computational resources.

4 Implementation

Although the levels of interaction are merely a conceptual architecture to frame the interactions between the player and the characters, their implementation relies on existing techniques, some of which are already implemented in game engines.

In order to prove both the relevance of the framework and its technical efficiency, a first implementation of the levels of interaction has been provided for the *Roma Nova* project. The implementation is based on the Unity game development tools [15]. Based on a recent internal study [20], Unity has shown to offer the best compromise between functionality and accessibility.

4.1 Computational Efficiency

A fundamental challenge addressed by the project is the effective modelling of interactions between the player and characters in a large environment, wherein a substantial number of entities are likely to interact in a non-linear and emergent way. Interestingly, this problem is very similar to modelling the visual and behavioural fidelity of the crowd. Many state-of-the-art techniques allow for the creation of visually realistic crowds. However, each technique is individually demanding in terms of computational resources and hence simplifications must be done in order to avoid disrupting graphical flow and fidelity and impeding perceived realism. Such techniques for ameliorating the impact of high-fidelity content on frame rate are known as *levels of detail* and *levels of simulation*.

The Level of Detail approach [18, 23] (LoD) involves creating, as part of a pre-process or at runtime, 3D representations of each object, capable of being resolved to varying levels of triangle resolution. There are currently four different frameworks for LoD: discrete, continuous, view dependent and hierarchical. Discrete LoD is a traditional approach that pre-processes several meshes of different complexity for each object. Continuous LoD adjusts the detail gradually. View-dependent LoD renders the portion of the object that is closer in higher detail. The Hierarchical LoD treats the scene as a single object and it simplifies the object according to the position.

The classical discrete implementation of the LoD has been preferred for *Roma Nova*. Although it requires more work from the modeller, we found at least one advantage using this technique as most game engines already include an implementation (like Unity for instance). The original mesh for the Roman character used in the game has been reworked using a modelling software in order to offer 3 levels of detail. A lower one, using a polygon simplification, was designed for distant crowd characters. Conversely, a higher mesh has also been designed for the dialogic interactions. Parts of the anatomy have been smoothed to avoid unpleasing angles whereas others have been re-created from scratch like eyes or lips, in order to fit the higher visual requirements of a close-up interaction. During the game, these meshes are switched seamlessly during the transitions between the levels of interaction, as explained further in this section.

The levels of simulation technique [30] (LoS) allows the computation time to be distributed according to the spectator's viewpoint, causing the quality to be enhanced in areas where attention is focused, e.g. around the viewpoint. The quality is decreased towards occluded or inconspicuous areas. For instance, the path-finding routine for some characters can be approximated

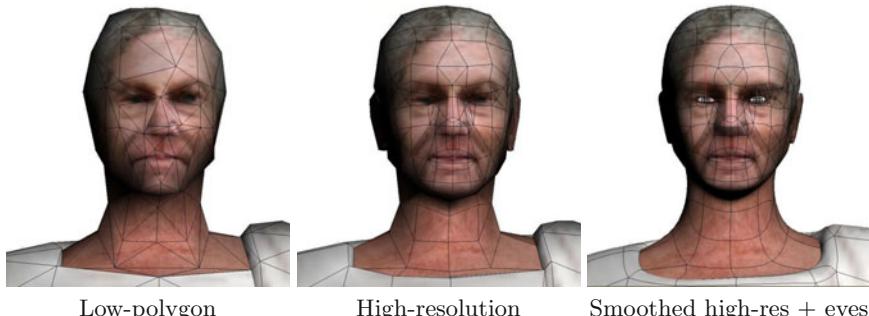


Fig. 8 Illustration of how levels of detail have been implemented for the *Roma Nova* project. Meshes of different complexity have been created on the basis of the original high polygon mesh. A polygon simplification has been applied to reduce the mesh complexity by 60% for the low-polygon version. The mesh has been smoothed and parts have been manually enhanced (eyes, lips, etc.) for the ultra high-resolution version.

and collision detection reduced in accuracy or removed. Whereas LoS are usually tricky to implement, as the behaviours have to be parameterised, the LoI framework provides a convenient simplification, allowing for one or more specific behavioural components to be attached to each level of interaction. Switching between two behaviours is managed exactly like switching the models, as explained below.

When modelling a crowd, the LoD and LoS techniques may be used in concert, such that NPCs closest to the camera are rendered and animated with the highest degree of fidelity (see figure 9). By comparison, NPCs in the far distance are typically rendered using lower resolution models and animations. Since the scene is intended as dynamic, LoD and LoS also address issues related to transitions between different levels. The key to implementing such an approach effectively is enabling seamless and imperceptible transitions between levels to the user and central to this is the seamless modification of the mesh [14] or the animation of the characters as it approaches or moves away from the observer.

Implementing the levels of interaction was quite straightforward as Unity allows to use boundary spheres associated with triggers (figure 10). The implementation relies on these spheres to trigger a change of the 3D model and/or the behavioural controller, therefore providing an implementation of LoD and LoS. Different controllers are loaded in real-time as characters enter or exit the levels of interaction. These controllers are then responsible for the degree of interaction allowed by the NPC.

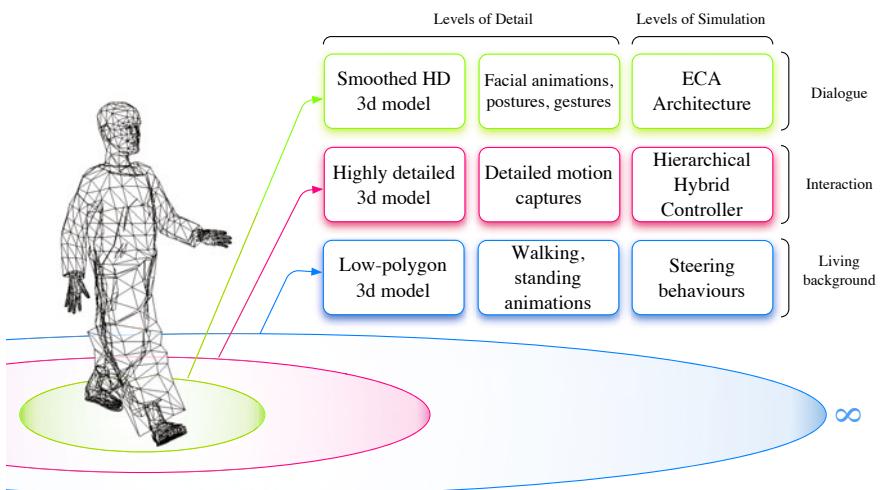


Fig. 9 Technically speaking, the LoI are implemented together with LoD and LoS, so that each level of interaction is related to a level of detail – one or several 3D models along with animations – and a level of simulation – one or several behavioural controllers.

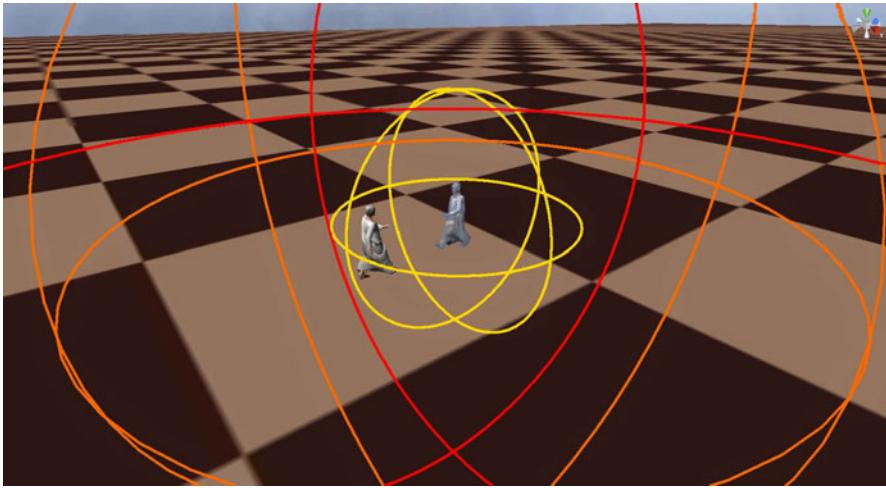


Fig. 10 Unity allows to use boundary spheres, on which triggers can be attached. The implementation of the LoI takes full advantage of this feature. Any NPC entering a sphere is triggered by the system and sent a message in order to update both its 3D model and behavioural controller. When leaving the sphere, a similar operation happen and the controller is dropped.

4.2 Boundaries

Whereas a critical aspect of the framework (as can be seen on figure 4) concerns the boundaries between each level of interaction, no indication of distance has been given so far. Although the boundaries have been considered as fixed radii centered on the player's avatar, the first implementation tends to suggest that it may only be a representation simplification and that a refinement of the definition of these concentric auras is required to accommodate a better understanding of human perceptual traits. Distance does not necessarily imply salience [8, 3] and the LoI approach may integrate with models for determining salience to provide a more perceptually-driven allocation of resources.

For instance, adjustable boundaries would provide dynamicism with respect to the graphical flow of the game, in addition to LoD and LoS techniques. Since the LoI integrates these techniques, as explained in section 4.1, they are directly responsible for the number of characters inhabiting each level of interaction. Assuming that a NPC showing a complex behaviour requires more computational resources than a background or a conversational character, as expressed in table 7, the LoI are responsible for the overall CPU and memory consumption of the game. To illustrate this, one can imagine for example that the number of characters allowed in the interaction level can be tweaked in real time by shrinking or expanding the corresponding

boundary in order to guarantee a constant framerate while maximising the player's immersion.

The dialogue level benefits from a specific rule. Although it is schematised as a radius, it is restricted to contain only one character at once, as current research on ECAs merely considers the verbal interaction with one agent only.

Another important aspect of the boundaries concerns the design of how they are meant to be crossed by the NPCs. Although switching the mesh and the controller is technically speaking an almost trivial operation, as explained in the previous section, the transition visually is likely to be perceived by the player and therefore act as an impediment to the immersion. Oscillatory phenomena and hysteresis must be foreseen as well, for instance when a NPC walks in the same direction as the player for a while, in and out of the interaction level. As a result, special care must be taken to ensure the visual and behavioural continuity between crowd and interacting characters.

Again, for the dialogue level, things are different given the specific "one character" rule. Also, the transition to a dialogue should not be seamless and graphically invisible but on the opposite explicitly notified to the player. The next section elaborates on how the interactions have been managed in the first implementation of *Roma Nova*. This includes an elegant solution for the dialogue transition.

4.3 Interactions

For the first implementation of *Roma Nova* we decided to temporarily defer the interaction level, focusing on the two remaining ones, and place specific emphasis on the transition from a crowd behaviour to the dialogue phase. In the current version, a crowd of Roman characters is introduced in the Forum, a highly detailed set of buildings that belong to the *Rome Reborn* model. The NPCs are wandering in the 3D environment between predefined points of interest, while the player is able to move freely (figure 11).

When clicked on by the player, a NPC changes its current target to the position of the player, and hence starts walking towards the player. When the NPC is close enough to enter the dialogue level, a series of actions is triggered by the engine:

- An animation is triggered to change the camera from a wide angle to a close-up perspective.
- The smoothed highly detailed version of the Roman character mesh (figure 8) is loaded to replace the low polygon version, along with the corresponding animations.
- The steering controller attributed to every background character is dropped and replaced by a simple ECA engine developed on purpose to play the scenario (figure 12).



Fig. 11 A crowd of Romans walk through the Forum to reach randomly distributed points of interest. The player, through his avatar, is free to summon any of them to start a dialogue.

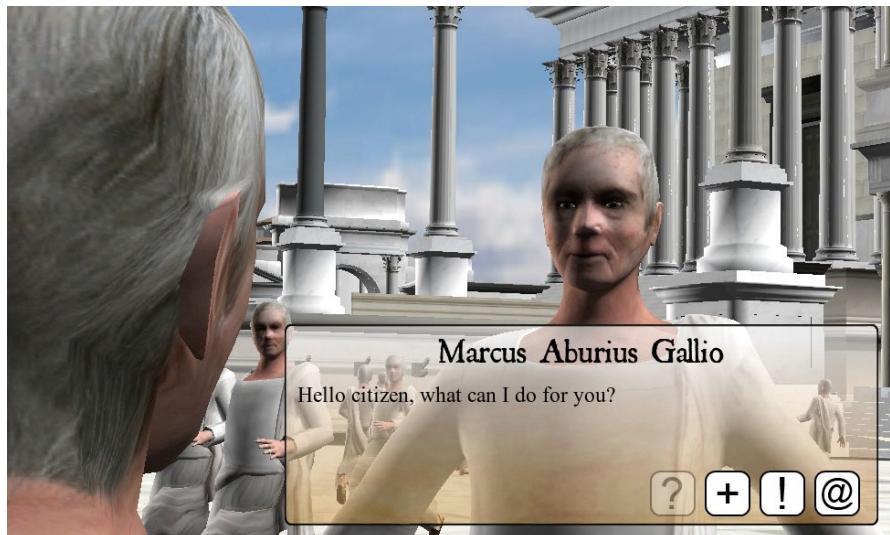


Fig. 12 At the moment, the dialogues in *Roma Nova* consist of a simple textual interaction between the player and any character of the game. Future work intends to provide the conversational Romans with a fully-fledge ECA engine and content-rich dialogues.

At the end of the dialogue, the reverse set of operations is performed. The simple 3D model and the steering behaviours are re-applied to the NPC. It carries on its navigation to its previous point of interest as the computer memory is freed from the ECA engine and the high definition 3D model.

The next milestone will intend to introduce a first attempt of interactive behaviour. The characters from the crowd, as they get closer and pass by the player and thus enter the interaction level, will be endowed with a more realistic 3D model and gaze animations in order to visible respond to the player and thus acknowledge his presence. Characters already met by the player will be able to wave at him, hopefully encouraging the player to start a new conversation or to resume the previous one.

5 Conclusion

As well as a challenging opportunity to study how history could be taught *in situ* by means of a serious game, the *Roma Nova* project is an opportunity to explore the technical issues that must be solved to allow such a scope of interactions with a large number of believable virtual characters in an open and immersive environment. In this paper, our intention was to show that although this kind of environment has never been seen before in a simulation or a commercial game, it is less a question of computational efficiency than the lack of an adequate framework. Not only do the Levels of Interaction apply a divide-and-conquer strategy on the broad range of interactions, ultimately relying on current techniques to enable the implementation, but they also tailor the graphical, animation and behavioural levels to the interaction, so that the immersion of the player is always ensured.

Future work consists of the continuation of development work in order to explore and further refine the framework in the context of the *Roma Nova* case study. In parallel, further research intends to extend the framework to multiplayer games, where several players are immersed in the same environment, along with a crowd of NPCs and a human tutor.

Acknowledgements. The authors wish to express their thanks to the *Roma Nova* development team. All credits regarding the implementation of the project and the illustrations provided in the paper go to Kévin Sauvourel, Alexandre Spyropoulos and Guillaume Patinaud.

References

1. Anderson, E.F., McLoughlin, L., Liarokapis, F., Peters, C., Petridis, P., Freitas, S.d.: Serious games in cultural heritage. In: 10th VAST International Symposium on Virtual Reality, Archaeology and Cultural Heritage (VAST 2009), VAST-STAR, Short and Project Proceedings, Malta, pp. 29–48 (2009)

2. Badler, N., Bindiganavale, R., Allbeck, J., Schuler, W., Zhao, L., Palmer, M.: Parameterized action representation for virtual human agents. In: *Embodied Conversational Agents*, pp. 256–284 (2000)
3. Beeharee, A.K., West, A.J., Hubbold, R.: Visual attention based information culling for distributed virtual environments. In: *VRST 2003: Proceedings of the ACM symposium on Virtual reality software and technology*, pp. 213–222. ACM, New York (2003)
4. Bevacqua, E., Mancini, M., Niewiadomski, R., Pelachaud, C.: An expressive eca showing complex emotions. In: *AISB 2007 Annual convention, workshop Language, Speech and Gesture for Expressive Characters*, Newcastle, UK, pp. 208–216 (2007)
5. Bevacqua, E., Mancini, M., Pelachaud, C.: A listening agent exhibiting variable behaviour. In: Prendinger, H., Lester, J.C., Ishizuka, M. (eds.) *IVA 2008. LNCS (LNAI)*, vol. 5208, pp. 262–269. Springer, Heidelberg (2008)
6. Csikszentmihalyi, M., Kubey, R.: Television and the rest of life: A systematic comparison of subjective experience. *Public Opinion Quarterly* 45(3), 317–328 (1981)
7. Doswell, J.T.: Pedagogical embodied conversational agent. In: *IEEE International Conference on Advanced Learning Technologies*, pp. 774–776 (2004)
8. Dunwell, I., Whelan, J.C.: Spotlight interest management for distributed virtual environments. In: *Proceedings of 14th Eurographics Symposium on Virtual Environments*, Eindhoven, NL (2008)
9. Farenc, N., Boulic, R., Thalmann, D.: An informed environment dedicated to the simulation of virtual humans in urban context. In: *Eurographics*, pp. 309–318 (1999)
10. de Freitas, S., Neumann, T.: The use of 'exploratory learning' for supporting immersive learning in virtual environments. *Computers and Education* 52(2), 343–352 (2009)
11. Greenhalgh, C., Benford, S.: MASSIVE: a collaborative virtual environment for teleconferencing. *ACM Trans. Comput.-Hum. Interact.* 2(3), 239–261 (1995)
12. Guidi, G., Frischer, B.: Virtualizing ancient rome: 3d acquisition and modeling of a large plaster-of-paris model of imperial rome. In: *Videometrics VIII*, San Jose, California, USA, pp. 119–133 (2005)
13. Hegeas, L., Luciani, A., Thollot, J., Castagné, N.: A physically-based particle model of emergent crowd behaviours. In: *Proceedings of Graphicon* (2003)
14. Hoppe, H.: Progressive meshes. In: *SIGGRAPH 1996: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pp. 99–108. ACM, New York (1996)
15. <http://unity3d.com/>
16. Kim, H., Di Giacomo, T., Egges, A., Lyard, E., Garchery, S., Magnenat-Thalmann, N.: Believable virtual environment: Sensory and perceptual believability. *Believability in Virtual Environments and Emotion Simulation* (2000)
17. Lightfoot, T., Milne, G.: Modelling emergent crowd behaviour. In: *The Australian Conference on Artificial Life (ACAL)*, pp. 159–169. University of New South Wales, Canberra (2003)
18. Luebke, D., Watson, B., Cohen, J.D., Reddy, M., Varshney, A.: *Level of Detail for 3D Graphics*. Elsevier Science Inc., New York (2002)

19. MacNamee, B., Dobbyn, S., Cunningham, P., O'Sullivan, C.: Men behaving appropriately: Integrating the role passing technique into the aloha system. In: Proceedings of the AISB 2002, pp. 59–62 (2002)
20. Magnenat-Thalmann, N., Foni, A.E., Cadi-Yazli, N.: Real-time animation of ancient roman sites. In: GRAPHITE 2006: Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia, pp. 19–30. ACM, New York (2006)
21. Mam, J., Haegler, S., Yersin, B., Müller, P., Thalmann, D., Van Gool, L.: Populating Ancient Pompeii with Crowds of Virtual Romans. In: Proceedings of the 8th International Symposium on Virtual Reality, Archeology and Cultural Heritage - VAST (2007)
22. McDonnell, R., Larkin, M., Dobbyn, S., Collins, S., O'Sullivan, C.: Clone attack! perception of crowd variety. In: SIGGRAPH 2008: ACM SIGGRAPH 2008 papers, pp. 1–8. ACM, New York (2008)
23. O'Sullivan, C., Cassell, J., Vilhjalmsson, H., Dingliana, J., Dobbyn, S., MacNamee, B., Peters, C., Giang, T.: Levels of detail for crowds and groups. Computer Graphics Forum 21(4), 733–742 (2002)
24. Panzoli, D., Peters, C., Dunwell, I., Sanchez, S., Petridis, P., Protopsaltis, A., Scesa, V., de Freitas, S.: Levels of interaction: A user-guided experience in large-scale virtual environments. In: IEEE 2nd International Conference in Games and Virtual Worlds for Serious Applications (VS GAMES 2010), Braga, Portugal, pp. 87–90 (2010)
25. Panzoli, D., Qureshi, A., Dunwell, I., Petridis, P., de Freitas, S., Rebolledo-Mendez, G.: Levels of interaction (loi): a model for scaffolding learner engagement in an immersive environment. In: Tenth International Conference on Intelligent Tutoring Systems: Bridges to Learning, Pittsburgh, pp. 821–823 (2010)
26. Pausch, R., Proffitt, D., Williams, G.: Quantifying immersion in virtual reality. In: Proceedings of the 24th annual conference on Computer graphics and interactive techniques, pp. 13–18. ACM Press/Addison-Wesley Publishing Co. (1997)
27. Peters, C.: Direction of attention perception for conversation initiation in virtual environments. LNCS, pp. 215–228 (2005)
28. Peters, C., Ennis, C.: Modeling groups of plausible virtual pedestrians. IEEE Computer Graphics and Applications 29(4), 54–63 (2009)
29. Petridis, P., Dunwell, I., de Freitas, S., Panzoli, D.: An engine selection framework for high fidelity serious games. In: The 2nd International Conference on Games and Virtual Worlds for-Serious-Applications (VSgames 2010), Braga, Portugal (2010)
30. Pettré, J., Ciechomski, P.d.H., Maïm, J., Yersin, B., Laumond, J.P., Thalmann, D.: Real-time navigating crowds: scalable simulation and rendering: Research articles. Comput. Animat. Virtual Worlds 17(3-4), 445–455 (2006)
31. Reynolds, C.W.: Flock, herds ans schools: a distributed behavioural model. In: SIGGRAPH 1987. Computer Graphics, vol. 21(4), pp. 25–34. ACM Press, Anaheim (1987)
32. Robertson, G., Czerwinski, M., Van Dantzig, M.: Immersion in desktop virtual reality. In: Proceedings of the 10th annual ACM symposium on User interface software and technology, pp. 11–19. ACM Press, New York (1997)

33. Sanchez, S., Luga, H., Duthen, Y., Balet, O.: Bringing autonomy to virtual characters. In: Ramos, F.F., Unger, H., Larios, V. (eds.) ISSADS 2004. LNCS, vol. 3061, Springer, Heidelberg (2004)
34. Shao, W., Terzopoulos, D.: Autonomous pedestrians. Graph. Models 69(5-6), 246–274 (2007)
35. Waldau, N., Gattermann, P., Knoflacher, H., Schreckenberg, M. (eds.): Pedestrian and Evacuation Dynamics. Springer, Heidelberg (2007)
36. <http://www.romereborn.virginia.edu>

A Comparison of Two Machine Learning Approaches for Photometric Solids Compression

Samuel Delepoulle, François Rousselle, Christophe Renaud, and Philippe Preux

Abstract. The use of photometric solids into both real time and photorealistic rendering allows designers and computer artists to enhance easily the quality of their images. Lots of such data are available from lighting societies since they allow these societies to easily present the luminance distribution of their often complex lighting systems. When accuracy is required the amount of discretized luminance directions and the number of photometric solids that have to be used increase considerably the memory requirements and reduce the algorithm efficiency. In this paper we describe and compare two machine learning approaches used for approximating any photometric solid: an artificial neural network and ECON (Equi-Correlated Network Algorithm). By applying these two approaches on a large set of real photometric distribution data, we were able to show that one of them provides generally a better approximation of the original distribution.

Keywords: approximation, image render, machine learning, photometric solids, light distribution.

Samuel Delepoulle · François Rousselle · Christophe Renaud

Université Lille- Nord de France ULCO

LISIC - BP 719 - 62228 Calais cedex

e-mail: delepoule@lisic.univ-littoral.fr,

rousselle@lisic.univ-littoral.fr, renaud@lisic.univ-littoral.fr

Philippe Preux

Université de Lille

Centre de Recherche INRIA-Lille Nord Europe & Laboratoire d'Informatique

Fondamentale de Lille (LIFL, UMR CNRS 8022)

Parc Scientifique de la Haute Borne 40, avenue Halley,

59650 Villeneuve d'Ascq

e-mail: philippe.preux@inria.fr

1 Introduction

Computing realistic lighting effects for image synthesis requires to model accurately the light propagation and the interaction of light with the object materials. Another important feature is an accurate model of the source light distribution since it determines all the lighting simulation. This can be done by modeling in depth the numerous parts of any lighting system and then to use this model in the lighting simulation. The main drawback of such an approach is that modeling is often complex and simulation times increase according to the details the source is modeled with. this approach is obviously not really usable for any real-time rendering software.

One way to approximate the main features of the light distribution of any lighting system is to use a discretized representation of the photometric distribution of this system: measurements are performed for a set of directions around the center of the lighting system which is approximated as a point light source. Assuming a large distance between any illuminated point and the source, interpolating between discrete photometric data can then provide a useful approximation of the source light distribution. This light distribution is known as a photometric solid (PS) (see figure II for examples) and is generally stored in a standardized representation known as the IES file format [IES95].

When accuracy is required in a rendering software, a large number of measurements have to be performed and the size of the IES file increases considerably. Then when large light sources are approximated or for close to source illuminated points, the point light source approximation becomes false and results are far from being accurate. A workaround is to split the source area into small sub-sources, each one having its own PS [Den02]. Obviously, this increases both memory requirements and computation times. Using a large number of accurate photometric solids thus involves finding a way of reducing their memory consumption and their computation time overhead. Up to our knowledge, compressing PS has not yet been much studied. Lots of works have been done in some related problems such as compressing Bidirectionnal Reflectance Distribution functions (BRDF) [SAWG91][CPB06][NP00], and for approximating 3D shapes out of large point clouds [WSC06]. In the latter, the goal is not the same as ours since the purpose is to allow the shape to be rendered through its new representation using potentially cost effective algorithms. For the former, Deniel [Den02] showed that the classical approaches for its resolution are not well-suited for photometric solids compression. He then proposed a hierarchical adaptive compression model that efficiently compress the PS and provides a fast access to the directionnal luminance data. In previous works, we studied the use of machine learning methods [DRP09][LPDR09] for PS compression. The goal was to use these methods in order to learn the “shape” of any PS and then to be able to approximate it with very few basis function. The outcome of this work was to compress efficiently the PS and to be able to efficiently generate the luminance emitted from the source along any direction. In this paper, we are

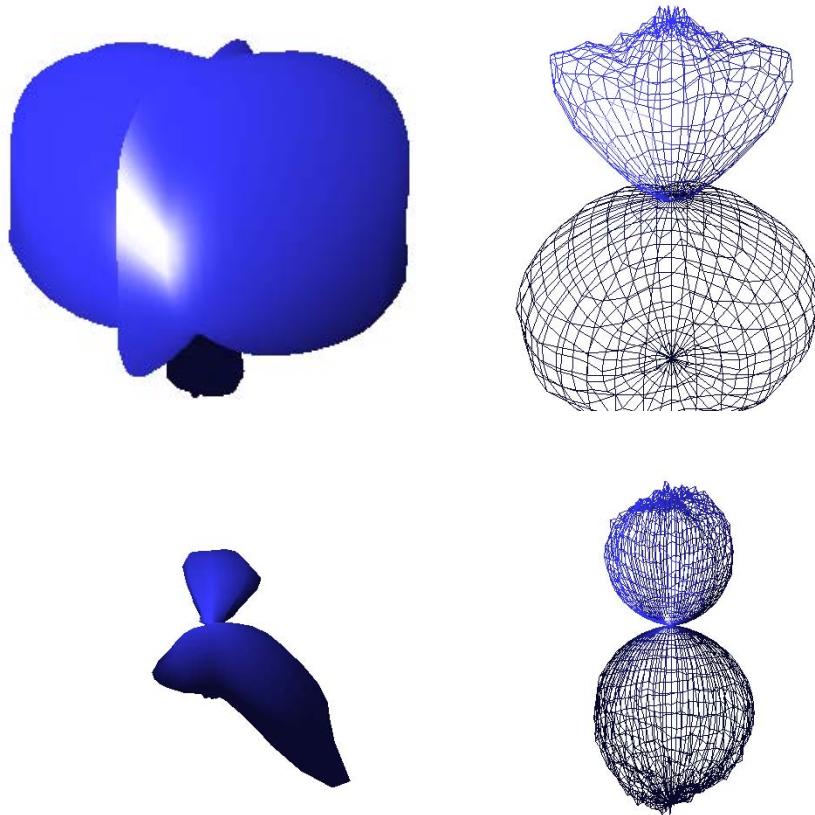


Fig. 1 A sketch of four photometric solids built from data extracted from the Metalumen photometric data files library [Met]

interested in comparing our two previous approaches on a large data set of PS. This allows us to provide a better analysis of their advantages and drawbacks.

In the next section, we provide a brief overview of the machine learning techniques that are suitable to deal with the problem at hand, and describe more precisely the two methods we compare. Our methodology is detailed in section 3 and our results are presented in section 4.

2 Machine Learning Approach

The general idea is to use supervised learning to produce a regression model. The regression problem may be summarized as follows: we have a set of N examples (x_i, y_i) with $i \in \{1, \dots, N\}$, $x_i \in \mathcal{D} \subset \mathbb{R}^P$, $y_i \in \mathbb{R}$; y_i is assumed to be a noisy realization of an unknown function y , that is: $y_i \equiv y(x_i) + \epsilon$

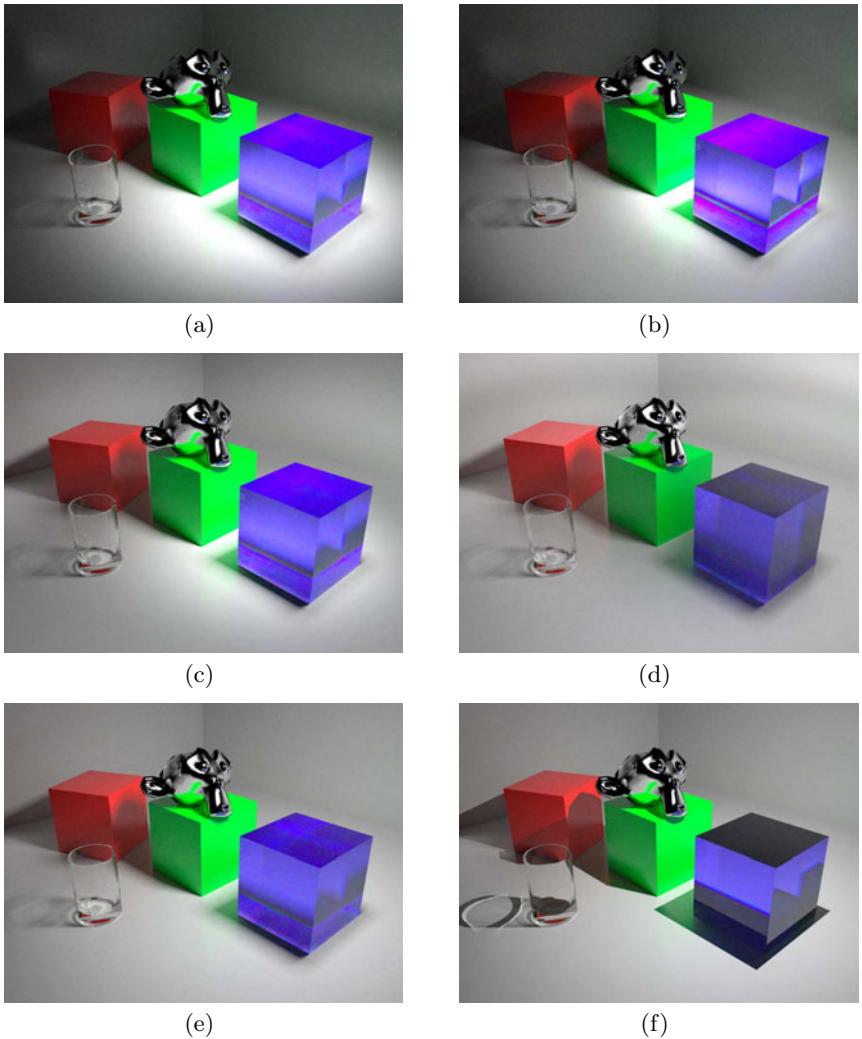


Fig. 2 Views of the same scene using different light distributions

where ϵ denotes some noise. The goal is to learn, or estimate, a function \hat{y} that approximates as well as possible y , and so that \hat{y} may be further used to predict the value associated to any data $x \in \mathcal{D}$. \hat{y} may also be considered as a model of the data, a model that is learned in order that the function y may subsequently be estimated for any data x in the domain. The model is obtained by induction, that is, from data, one derives a general rule. The regression problem has been studied for at least two centuries, and tens of thousands publications, and hundreds techniques and algorithms are known. The study of this problem is still a very active research area, with very

significant challenges still pending. There has been very significant advances since the 1980's, with the development of multi-layer perceptrons, and more recently, the development of statistical learning which is deeply rooted in statistics, the theory of function approximation, and functional analysis.

2.1 Artificial Neural Networks

In this work, to learn a \hat{y} , we use a multi-layer perceptron (MLP), and we restrict ourselves to 1 hidden layer perceptrons [Hay08]. In this case, \hat{y} has the form:

$$\hat{y}(x.) = S_o \left(\sum_{k=1}^{K} w_k S_h(a_{j,k} x_{.,j}) \right) \quad (1)$$

where S_o (resp. S_h) is the so-called activation function of the output (resp. hidden) unit, the w_k and $a_{j,k}$ are real weights, $x_{.,j}$ denotes the j^{th} component of the data $x.$, and K is the number of hidden units. Here, both S_o , and S_h are sigmoid functions of the form $S_o(x) \equiv \frac{1}{(1+e^{-x})}$. Training an MLP means finding the “best” value for the K weights w_k and the $P \times K$ weights $a_{j,k}$. Please note that K is fixed *a priori*.

To learn these weights, we initially set them to some value (arbitrary or not), so that \hat{y} may be computed for any data $x \in \mathcal{D}$. Basically, iteratively, one computes $\hat{y}(x_i)$ for some example x_i , and modifies the value of the weights according to the discrepancy between the expected value y_i , and the predicted value $\hat{y}(x_i)$. Various methods exist to achieve this, the most well-known being the backpropagation of error, which is a simple gradient descent algorithm.

2.2 ECON

Introduced in [LPDR09], the Equi-Correlated Network Algorithm (ECON for short) is considering models of the form:

$$\hat{y}(x) \equiv w_0 + \sum_{k=1}^{K} w_k \phi_k(x). \quad (2)$$

For each k , $\phi_k : \mathcal{D} \rightarrow \mathbb{R}$ is a feature function that maps a data to a real value; it may be any such function. Here again, the w_0 and the set of w_k are real weights. There are two points that come in sharp contrast with the MLP approach introduced above:

- K is not fixed: it is learned by ECON,
- generally, each feature function ϕ_k has some parameters; these parameters are also learned by ECON. For the sake of illustration, one may

consider that ϕ_k is drawn from the family of multivariate Gaussian functions $\phi_k(x) = e^{-(x-\mu_k)^T C_k(x-\mu_k)}$, and the parameters for such a function are the center $\mu_k \in \mathcal{D}$, and the covariance matrix $C_k \in \mathbb{R}^{P \times P}$.

These two points add a great flexibility to ECON with regards to previous algorithms.

Thus, ECON learns a lot of parameters: the very number of parameters itself (K), $K + 1$ weights, and the parameters of each feature function. As such this is an ill-posed problem (in Hadamard's sense). To solve it, we adopt a principled way based on a parsimonious approach, which tends to make K as small as possible; particularly, we look for the optimal parameters for the smallest value of K . Probably quite surprisingly, there is an exact algorithm to obtain the solution of this problem, and this algorithm is very efficient, and effective. To this end, ECON is a generalization of the LARS algorithm introduced in [OPT00], and named by [EHJT04]. The idea is to iteratively compute the optimal \hat{y} for each value of K , starting with $K = 0$, and until some stopping criterion is met. For each value of K , the weights are computed exactly, and the feature function parameters are obtained either analytically if the form of the feature function has such an analytical solution, or via numerical optimization in the general case.

2.3 ECON vs. MLP

MLP and ECON both learn an approximation of a real function given a set of samples of this function. The result is a weighted sum of feature functions. But some differences arise that should provide interesting advantages to ECON:

- a MLP has a fixed number of units whereas ECON learns the number of units that should be used;
- all the parameters of a MLP unit are the same for all units whereas ECON selects the parameters of each single feature function;
- ECON seeks a parsimonious representation, which means that it compresses the information.

In the next sections of the paper, we thus investigate the use of ECON for photometric solids approximation.

3 Methodology

3.1 The Learning Stage

In order to compare the results provided by the two learning methods (MLP and ECON), we apply a methodology that is sketched in figure 3. Each IES file used in our experiments is first of all submitted to both learning methods.

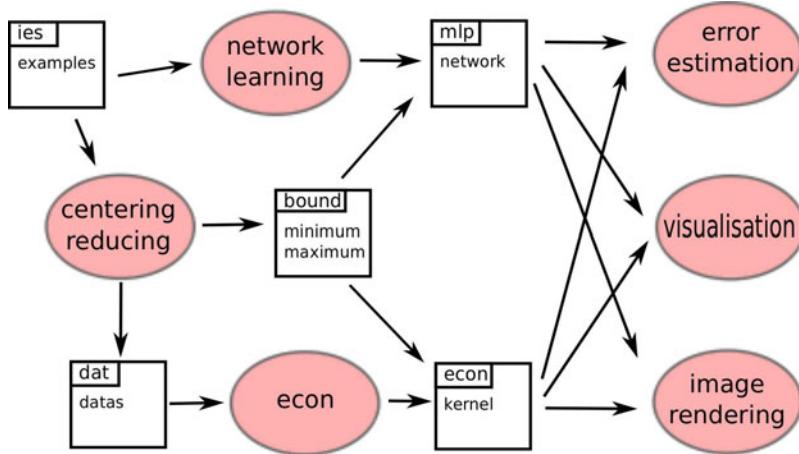


Fig. 3 The methodology of the approach used in this paper to compare MLP and ECON

Both methods use a subset of the input data: we use the rest of the data to control the accuracy of the model. This way, we avoid overfitting, that is, the model is too fit to training data, and no longer good to predict accurately other data in the domain. For practical purposes 80% of the input data are used for learning and 20% for evaluating the resulting approximator. This split is made at random.

After each learning algorithm has been run, its output (the model it induced) is written into a new file according to a specific file format. MLP outputs the weights, while ECON outputs the weights and the function parameters. According to the input data, the learning computation time can range from a few seconds to a few minutes. Note however that learning is performed only once and that once learned, the use of the model is very fast.

3.2 Using the Resulting Approximators

Once learning has been performed, MLP and ECON provide an approximator of a photometric solid. These approximators can then be inserted into any global illumination or realtime rendering method using a simple interface where an emission direction is provided to the approximator and a luminance value is returned.

The main use of the output of the approximators in this paper is the analysis of the discrepancy between the IES data and the approximated values. These outputs are thus used either in for error analysis (see section 3.3), or in a 3D visualization tool that can help us in locating the directions where large errors occur.

3.3 Computing the Error

In any rendering method, evaluating the accuracy of the new representation during its use requires the definition of a relevant error criterion. From a rigorous point of view, we should be able to compare the luminance emitted by the photometric solid along any direction to a measure of the luminance received along the same direction for the corresponding real lighting system. But in addition to the complexity of such measures, it is obviously difficult to access the numerous lighting systems for which photometric solids have been analyzed. We thus have to deal with this problem only with the IES available samples. One way to measure the error could be to compute a visual distance between two images rendered respectively with the original IES data and the new representation. But two problems arise with such an approach. The first one is concerned with the fact that images are generally computed with a classical camera model that allows only a part of the scene to be rendered. Some kind of fish eye camera could be used but it requires dealing with distorted images. The second problem is concerned with the choice of the interpolation method that has to be used with the IES data, since samples represent a small subset of the emission directions set. Because interpolation is used, it can introduce artefacts into the rendered images that would not be present in real images (see for example figure 4). It is thus difficult to get a relevant measure of the error in this case.

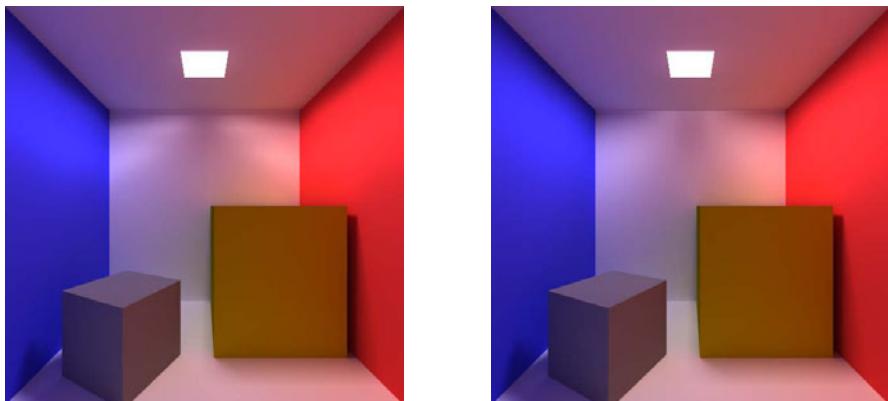


Fig. 4 Global illumination with IES data using linear interpolation (left) and with an MLP model (right). Lighting discontinuities due do the linear interpolation are clearly visible onto the back wall while they do not exist when using the MLP model.

Another way to measure the error is to consider the numerical distance between the available data and the corresponding values provided by the new approximation models.

Being unable to make a global comparison of two images, we have to use a local comparison based on a set of sampled directions, for which we compare the luminance obtained from the IES file, and from the MLP, or ECON. From this local error, we obtain a global error measure using the normalized root mean square error, and the normalized mean absolute error.

We used the normalized root mean squared error (RMSE) which is given by equation (3), where n is the number of directions used for the error computation, y the measured value of luminance along any direction and \hat{y} the value provided by the model along the same direction. Because IES file provides only discrete values, a linear interpolation is computed to find y .

$$RMSE = \sqrt{\frac{1}{n} \cdot \sum_{i=1}^n (y - \hat{y})^2} \quad (3)$$

Normalization is performed using equation (4), and provides the normalized root mean squared error (NRMSE). $\max(y)$ (resp. $\min(y)$) is the maximum value (resp. minimum value) of y over the n sampled directions.

$$NRMSE = \frac{RMSE}{\max(y) - \min(y)} \quad (4)$$

Mean square error has the disadvantage to over-represent the large error terms in the result: since differences are squared, the contribution of large error is exaggerated. That is why a Mean Absolute Error (MAE) and a Normalised Mean Absolute Error (NMAE) are computed (equations (5) and (6)).

$$MAE = \frac{1}{n} \cdot |\hat{y} - y| \quad (5)$$

$$NMAE = \frac{MAE}{\max(y) - \min(y)} \quad (6)$$

Another way to normalize the error that has been suggested in [Den02] is to normalise the error with respect to the statistical variance of y . We thus can define an error function according to equation (8).

$$ErrFunct = \sqrt{\frac{MSE}{\sigma_y^2}} \quad (7)$$

$$= \sqrt{\frac{\frac{\sum (\hat{y}-y)^2}{n}}{\sum \frac{y^2}{n} - \left(\frac{\sum y}{n}\right)^2}} \quad (8)$$

4 Results

4.1 Experimentation Data

To overcome the unavailability of a formal proof that one method outperforms the other, we compare MLP and ECON on a large set of IES data files. For this purpose we download the photometric data files libraries of two lighting societies: Ledalite [Led] (613 data files) and Metalumen [Met] (1310 data files). This provides us with 1.923 IES files that have all been used during our experimentations¹.

Regarding MLP and ECON, we have implemented them both in C. The MLP used here is a 1 hidden layer perceptron, with 10 units in the hidden layer (that is $K = 10$ in eq. (1)). ECON uses multivariate Gaussian feature functions with a diagonal covariance matrix.

Data are provided both to ECON and to an MLP. The median learning time is 28s for ECON and 16s for the MLP². Let us recall that learning is performed only once during the IES data compression.

4.2 Error

Table II presents statistical results for the two photometric data libraries that have been used during our experimentations. It highlights that ECON provides generally better results than the MLP approach for the two libraries. With ECON the median NRMSE is distinctly smaller: 0.06615 for Ledalite and 0.07119 for Metalumen (as compared to 0.1537 and 0.1718 for the MLP).

Table 1 Statistical results for the approximators provided by ECON and a Multi-layer Perceptron applied on the Ledalite and Metalumen photometric data libraries. For each, mean and median are shown. The median is less sensitive to abnormal values than the mean.

error mesure	Ledalite				Metalumen			
	ECON		MLP		ECON		MLP	
	mean	median	mean	median	mean	median	mean	median
RMSE	0.1357	0.1065	0.2283	0.2245	0.1631	0.09554	0.2234	0.2207
NRMSE	0.1164	0.06614	0.1791	0.1537	0.1639	0.07119	0.2073	0.1718
MAE	0.09481	0.07387	0.1786	0.1807	0.1181	0.06511	0.1732	0.1733
NMAE	0.08107	0.04472	0.1388	0.1217	0.1167	0.05020	0.1596	0.1317
ErrFunct	0.1540	0.1172	0.2283	0.2245	0.2491	0.2253	0.1736	0.1056

¹ Statistical analysis and graphical representation are made with the R [RD09].

² Learning is performed on an Intel®Core™2 Duo CPU T8300@2.40GHz running Ubuntu 9.10; programs are build using GNU g++ 4.4.1 compiler with -O3 optimization flags.

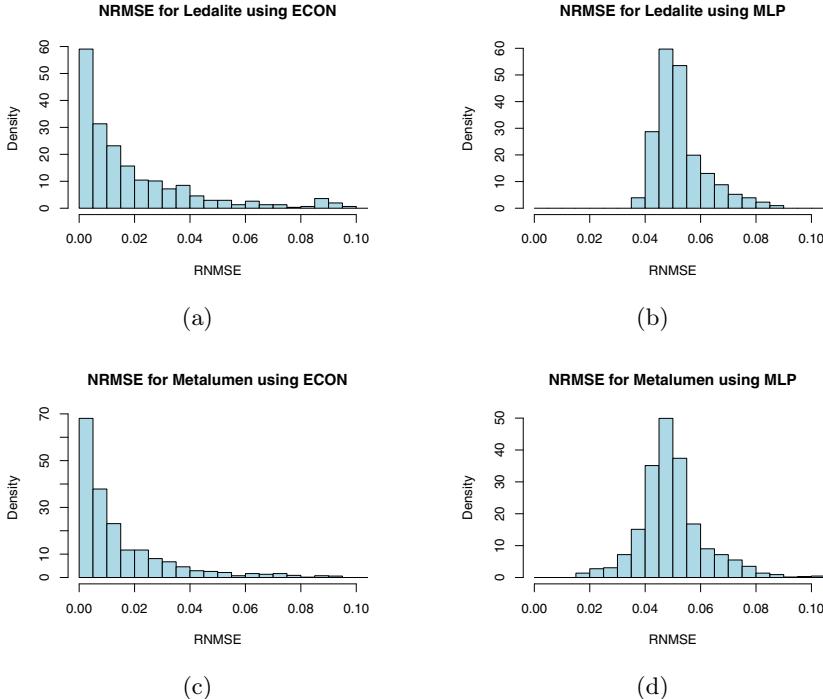


Fig. 5 Histograms of the NRMSE of the approximations provided by ECON and the MLP, measured on the set of 1923 IES files (Ledalite database on the left, Metalumen on the right). The shape of these distributions does not really depend on the library, but the dependence on the algorithm is striking.

In figure 5, for each photometric database and for each approximator, we report the distribution of the number of IES file concerned with some specific NRMSE error. The diagrams show clearly that most of the approximators provided by ECON are more accurate than those provided by the MLP.

Even if the NRMSE is relatively low in the two cases, the luminance distribution can be visually different. This is highlighted in the images of figure 6 where these distributions are visualized on a wall: the MLP approximator provides clearly a less accurate distribution especially in the grazing directions (additional comparisons are given in figures 10, 11, 12 and 13 for complex photometric solids and in figures 8 and 9).

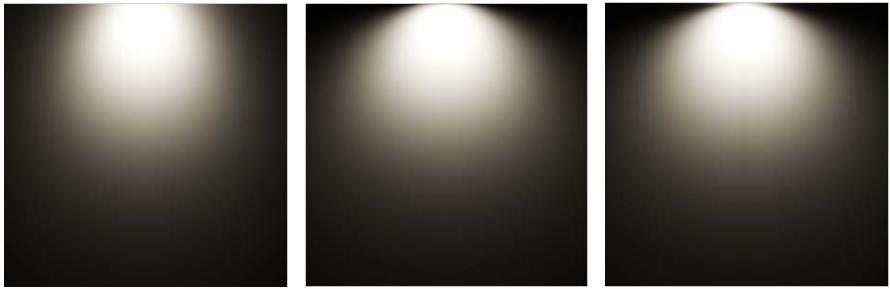


Fig. 6 We illustrate the visual impact of using the different methods on the Ledalite 9414D1H254 IES file. The 3 subfigures are the distribution of the luminance obtained by linear interpolation of the original IES file (center), the one obtained from the MPL approximator (left) and the one obtained from the ECON approximation (right). RNMSE(MLP) = 0.204, RNMSE(ECON) = 0.0369 for ECON.

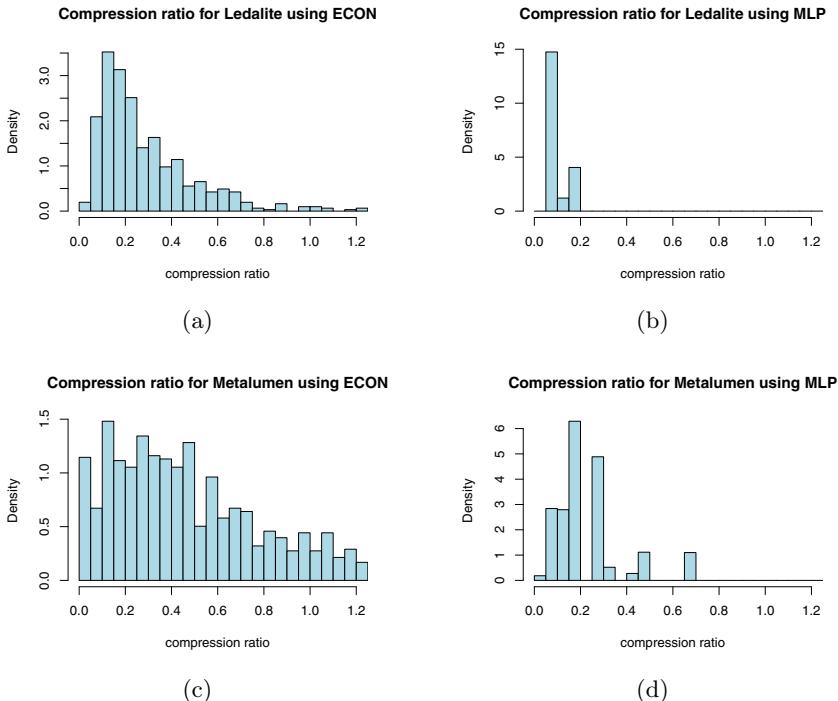


Fig. 7 Histograms of the compression rate C_r for ECON and the MLP, obtained on the set of 1923 IES files (Ledalite database on the upper row, Metalumen on the lower row). The shape of these distributions slightly depends on the library, while the dependence on the algorithm is striking.

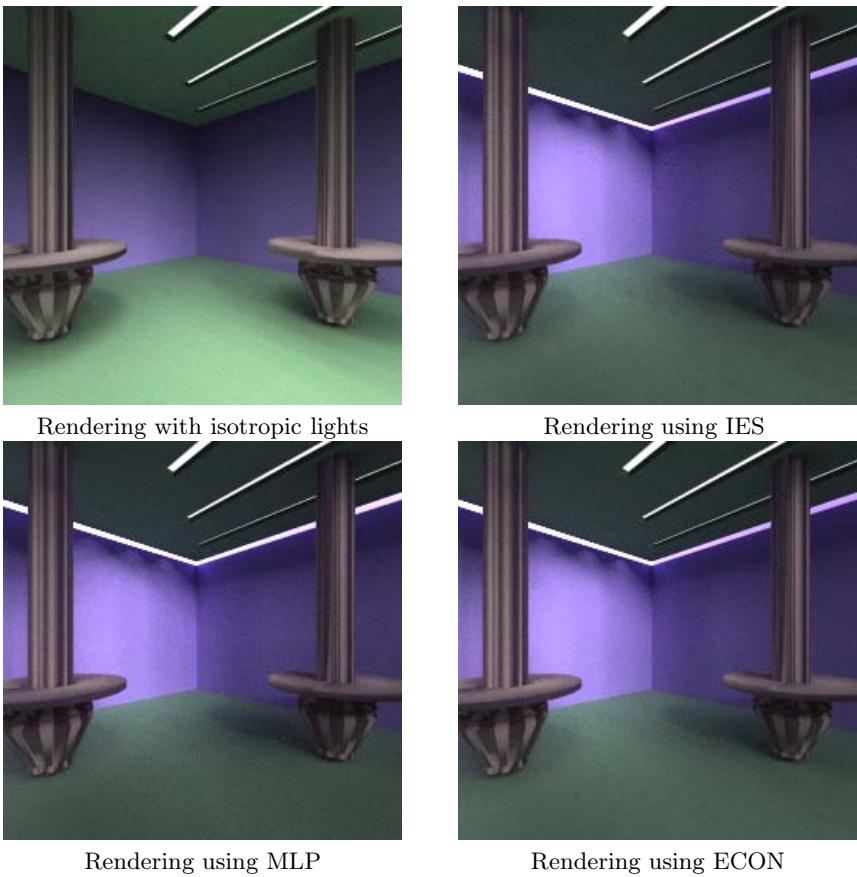


Fig. 8 Example of rendering using MLP or ECON as approximator. The same scene is used to compare with a linear interpolation of IES file and with isotropic light sources.

4.3 Compression

Another important way of evaluating the approximators is to measure the size of the data they generate as compared to the original data size. For this purpose we compute a compression ratio C_r that is defined as the ratio between the number n_{IA} of floating point values required for storing the data of the approximator and n_{IES} , the one of the corresponding IES file. Those values represent the weights for the MLP, the weights and feature parameters for ECON, and the angle and luminance values for the IES files.



Fig. 9 Example of rendering using a standard IES (center light), MLP (right light) or ECON (left light) as approximator in the same scene. Rendering is performed using Indigo Renderer.

$$C_r = \frac{n_{IA}}{n_{IES}}$$

According to this criterion, the MLP has generally better compression performances than ECON. The C_r median value for MLP is 0.0895 for the Ledalite library, and 0.1752 for the Metalumen library, whereas these values are respectively 0.2249 and 0.4342 for ECON. Here again, we can note that the dispersion of results is much more important with ECON (see figure 7). Furthermore we found that the compression ratio is greater than 1 for a few IES files using ECON. This is really surprising and must be investigated in future works.

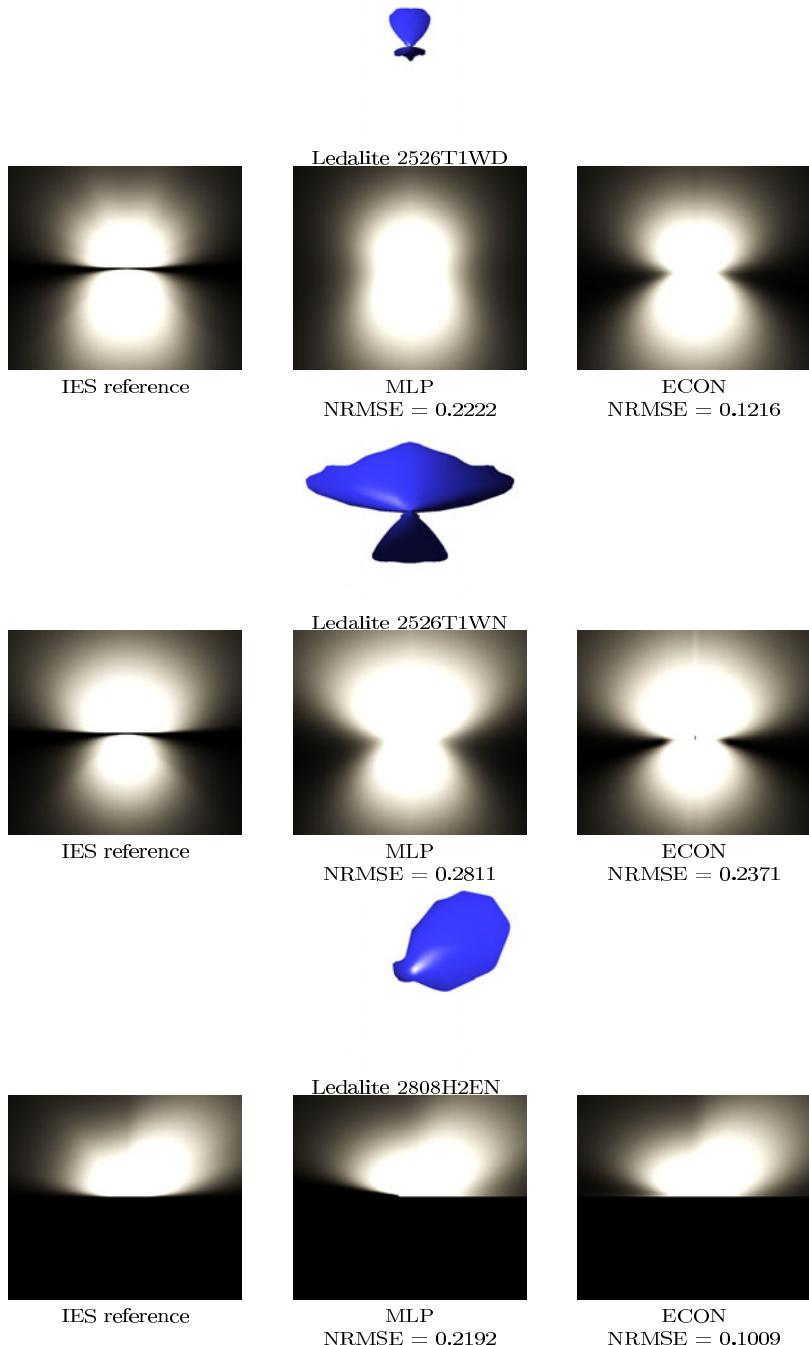


Fig. 10 Examples of rendering using MLP or ECON as approximator

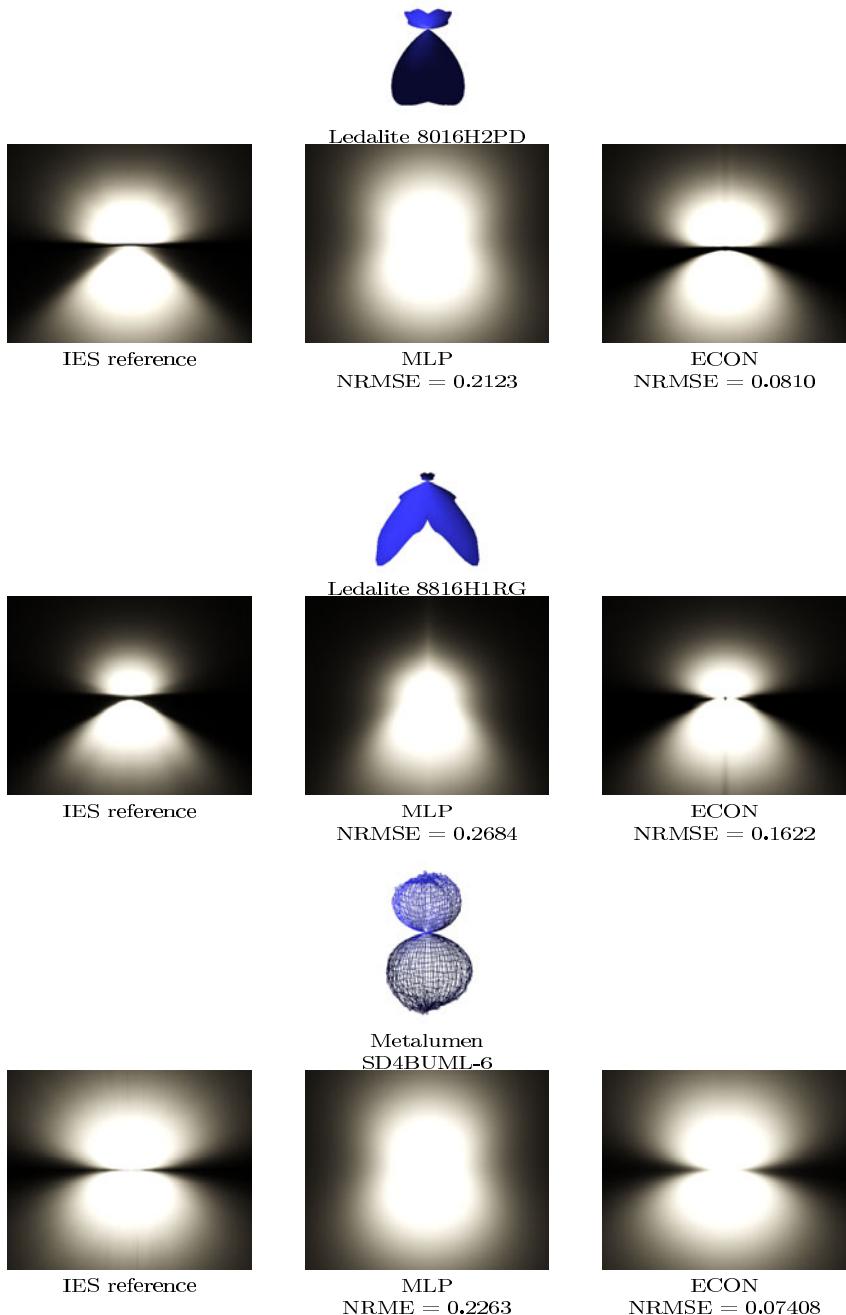


Fig. 11 Examples of rendering using MLP or ECON as approximator

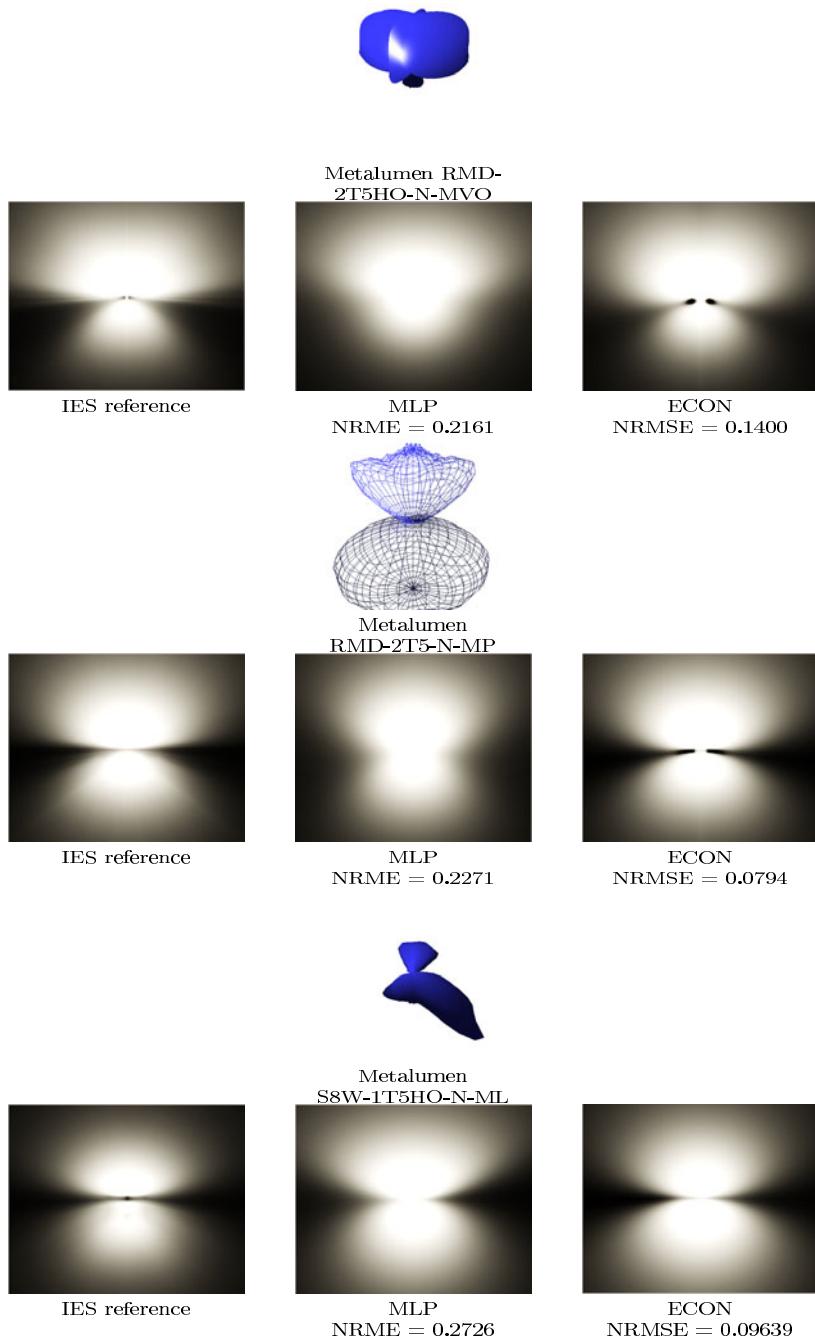


Fig. 12 Examples of rendering using MLP or ECON as approximator

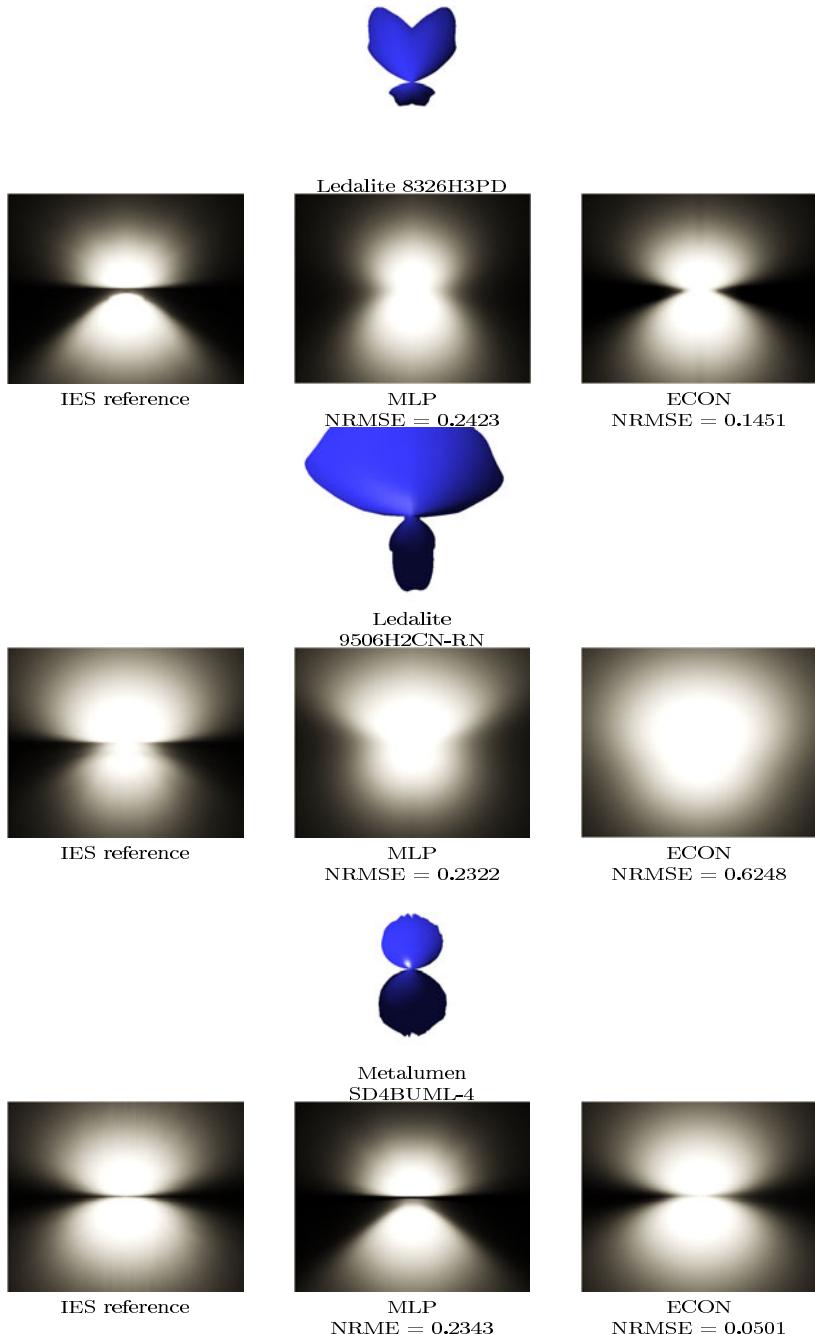


Fig. 13 Examples of rendering using MLP or ECON as approximator

5 Conclusion

In this paper we studied and compared the use of two machine learning approaches for photometric solids compression. By using a large number of photometric data from two lighting societies we were able to show that this kind of approach could be of great value for photometric approximation. In most cases the ECON algorithm appears to give the best approximator but in some few cases the MPL approach surpass its results. Reciprocally the compression ratio is generally better when using a MLP. These two features should be investigated deeply in a future work. Nonetheless one way to take advantage of the two approaches could be to use them simultaneously inside a multi-modeling approach: both approaches could be run onto the input file and their results would be compared in order to choose which one should be finally used according to a given criterion (error, compression rate, ...). This kind of approach should then be extended by studying the use of other feature functions in ECON. Finally, we will investigate the few IES files where either a large error, or a bad compression rate, has been found in order to be able to generalize our approach.

References

- CPB06. Claustres, L., Paulin, M., Boucher, Y.: A Wavelet-Based Framework for Acquired Radiometric Quantity Representation and Accurate Physical Rendering. *The Visual Computer* 22(4), 221–237 (2006)
- Den02. Deniel, J.M.: Modélisation des luminaires et des BRDF: réalisation, mesure et compression. PhD thesis, Université de Rennes 1 (2002)
- DRP09. Delepoule, S., Renaud, C., Preux, P.: Light source storage and interpolation for global illumination: a neural solution. In: Plemenos, D., Miaoulis, G. (eds.) *Artificial Intelligence Techniques for Computer Graphics. Studies in Computational Intelligence*, vol. 159, pp. 87–104. Springer, Heidelberg (2009)
- EHJT04. Efron, B., Hastie, T., Johnstone, I.M., Tibshirani, R.: Least angle regression. *Annals of Statistics* 32(2), 407–499 (2004)
- Hay08. Haykin, S.: *Neural Networks: A Comprehensive Foundation*, 3rd edn. Prentice Hall, Englewood Cliffs (2008)
- IES95. IESNA. *IESNA Standard File Format for Electronic Transfer of Photometric Data*. Illuminating Engineering Society of North America, Im-63-1995 edn. (1995)
- Led. Ledalite. Electronic online photometric data files library, <http://www.ledalite.com/>
- LPDR09. Loth, L., Preux, P., Delepoule, S., Renaud, C.: ECON: a Kernel Basis Pursuit Algorithm with Automatic Feature Parameter Tuning, and its Application to Photometric Solids Approximation. In: Wani, M.A., Kantardzic, M., Palade, V., Kurgan, L., Qi, Y. (eds.) *Proc. International Conference on Machine Learning and Applications (ICML-A)*, Miami, USA, pp. 162–169. IEEE Press, Los Alamitos (2009)

- Met. Metalumen. Electronic online photometric data files library, <http://www.metalumen.com>
- NP00. Noé, N., Péroche, B.: Hierarchical reconstruction of brdfs using locally supported functions. Computer Graphics Forum 19(2), 173–183 (2000)
- OPT00. Osborne, M.R., Presnell, B., Turlach, B.A.: A new approach to variable selection in least squares problems. IMA Journal of Numerical Analysis 20(3), 389–403 (2000)
- R D09. R Development Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2009) ISBN 3-900051-07-0
- SAWG91. Sillion, F., Arvo, J., Westin, S., Greenberg, D.: A global illumination solution for general reflectance distributions. Computers Graphics 25(4), 187–196 (1991)
- WSC06. Walder, C., Schölkopf, B., Chapelle, O.: Implicit surface modelling with a globally regularised basis of compact support. Computer Graphics Forum 25(3), 635–644 (2006)

A Collaborative Framework for Virtual Globe Based 3D City Modeling

Christos P. Yiakoumettis, Georgios Bardis, Georgios Miaoulis,
Dimitri Plemenos, and Djamchid Ghazanfarpour

Abstract. Urban planning and city modeling have gradually shifted, during the last decade, from highly demanding, in terms of required skills and supporting information, to tasks now supported by efficient, widely available applications, thus becoming popular and largely accessible. There are many GIS systems nowadays that offer a freely navigable three dimensional environment. This evolution of GIS systems has, in turn, led to the requirement for and creation of virtual 3D models of the ground and buildings. Online communities have created and distributed over the Internet libraries of georeferenced 3D models. The public is encouraged to participate in the design of 3D scenes and many companies offer free tools to facilitate the design of 3D models, specialized in buildings. In this paper, we present a collaborative approach for the construction of a city model, and its implementation through a prototype environment, employing freely available design tools. The prototype system comprises a collaborative database, supported by a web-based interface. Users are able to create and upload their models to the common database over the web, thus constructing a realistic 3D city model in a given area in a collaborative manner.

Keywords: virtual globe, GIS systems, collaborative city modelling, visualisation, public participation, scene modelling, urban planning.

1 Introduction

A number of Geographic Information Systems (GIS) applications have been released during the last years, one of the first Geographic Information Systems

Christos P. Yiakoumettis Dimitri Plemenos · Djamchid Ghazanfarpour
Université de Limoges, Laboratoire XLIM, CNRS, UMR 6172
83, rue d'Isle, 87000 Limoges France
e-mail: {djamchid.ghazanfarpour@unilim.fr}

Christos P. Yiakoumettis · Georgios Bardis · Georgios Miaoulis
Technological Education Institute of Athens
Department of Informatics
Ag.Spyridonos St., 122 10 Egaleo, Greece
Tel.: (+30) 2 10 53 85 826; Fax: (+30) 2 10 59 10 975
e-mail: {chgiak@teiath.gr, gbardis@teiath.gr, gmiaoul@teiath.gr}

being the one introduced by Microsoft in 1998 (Wikipedia, Virtual Globes) and included in the Encarta digital encyclopedia. Nevertheless, the first three dimensional geographic information system to become widely known was released by Google in 2004. It was the commencement of a series of applications, also known as *virtual globe* applications, that would follow in the subsequent years. All these GIS applications have tried to simulate real world based on a three dimensional virtual environment. Three dimensional scenes in such an environment are typically enriched with multimedia data which can be three dimensional models of buildings, photographs taken from an airplane or at street level, real time video, and more.

2 Virtual Globes

2.1 Simple Two Dimensional GIS System Architecture

Google Earth, released in 2004, was the first virtual globe application to break the limits of the experts' community and become widely known to the general public. It comprises a three dimensional virtual environment, freely navigable and explorable by the user. It is based on a server-client GIS system architecture where the typical scenario is initiated by the client requesting data, based on the user's current position in the virtual scene. In turn, the client retrieves the requested data either from a temporary cache memory or from the remote GIS database. Subsequently, the retrieved data are visualized on the client's side with the aid of a specialized application. The first applications which served this purpose were limited to two dimensional environments (Wikipedia, Google Maps), (Wikipedia, Virtual Globes). Google Earth started with publishing static maps, evolved to static web mapping applications and then to GIS distributed systems (Sripaisal & Pujari). Satellite or aerial photographs have been used to provide the necessary database content (Zhang, et al., 2007). Besides the images, systems of this category contain information about ground elevation, roads, points of interest as well as three dimensional building models (Jones, 2007). The global positioning system is used to integrate and calibrate the georeferenced content.

2.2 Evolution of the Architecture

The evolution of the GIS systems has led to the improvement of the services and the amount of available information. In the beginning, additional information was added to the database, mainly based on aerial photographs. Names for the streets and roads network were among the first data to enhance the database content. Afterwards, metadata such as text, external links, hypertext and placemarks also found their place the database.

Despite this evolution through additional material, the fundamental functional block diagram has remained practically the same (figure 1). All data is imported to databases and calibrated over the same geographical positioning system.

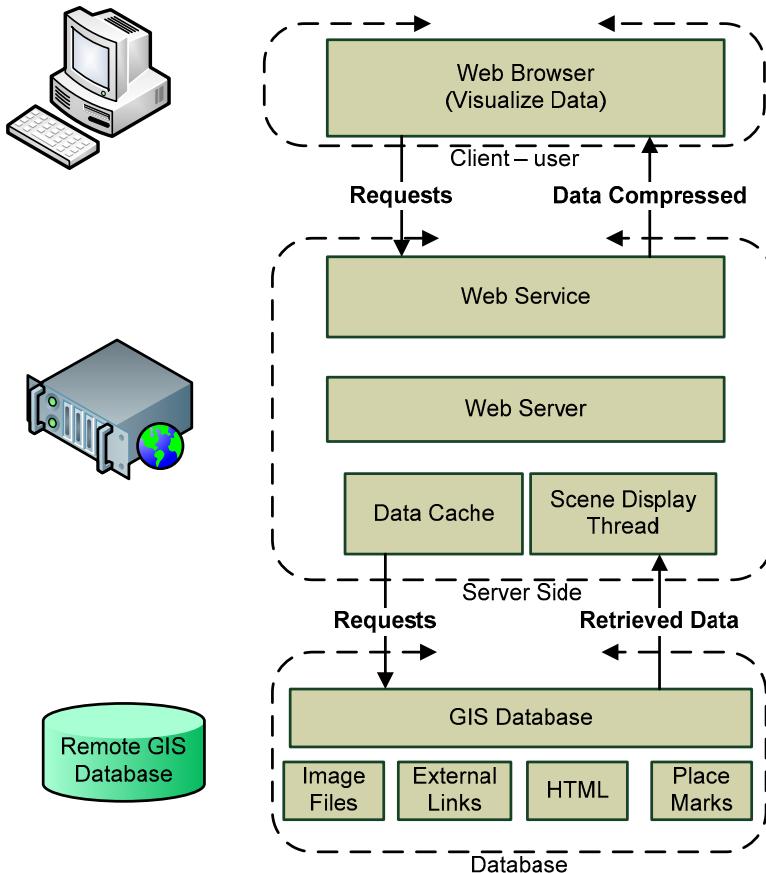


Fig. 1 Block Diagram of improved GIS system

In this case the data repository of the system is comprised of four different resources, each of them potentially forming a separate database. Nevertheless, all databases have to use the geographic positioning system. This architecture allows for system flexibility and easy content distribution to the extent that, for example, each database may be hosted by a different provider in physically remote locations from the others. Moreover, there is no theoretical limit on the number of physical databases that may constitute even one of the aforementioned resources. The end-user may choose which information to retrieve from the GIS system through the client application, which is presenting the data organized in intuitive layers, completely transparent with respect to the actual data organization. Each layer typically contains data from one or more different databases. When all the data are being presented, a synthesized image is created from the projection of the superimposed layers on the same geographic area.

Evolution led to three dimensional GIS platforms. Evolved platforms offer browsers with 3D capabilities and freely exploration of the scene. Beyond the images and hypertext, new browser can visualize the morphology of the ground. Moreover 3D models of buildings are also included in some browsers. (figure 2). The result is a synthesized three dimensional scene that can represent a three dimensional model of an area or city with realistic or reality-enhancing attributes. Initially, models of buildings were limited and only specific cities were modeled.

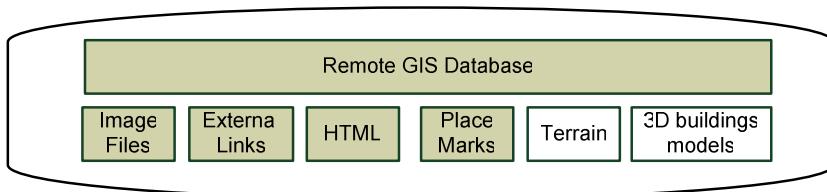


Fig. 2

3D models were designed and built in a 3D object description language called COLLABorative Design Activity (COLLADA). It has a language with XML schema and it has been adapted by many software and hardware developing companies.

On the user's side a standalone application or web browser with visualizing capabilities of COLLADA is required. Components have been developed by distributors for this purpose using OpenGL or DirectX graphics languages.

Today, there are numerous databases that can provide information and data for such a GIS system. The popularity of such systems and the advancements in computing and telecommunications have made possible the creation of powerful three dimensional GIS systems, based on the aforementioned architecture. Information about the altitude and the morphology of the ground has been incorporated to them whereas 3D models of buildings are also being gradually added. The goal is to create a virtual world that closely simulates the real one. Recently, many 3D models of buildings have started to appear on most popular capitals and cities around the world on the virtual globe systems. Each virtual globe application typically features its own independent database and is usually supported by its own developing team. Many cities have been modeled but most of them are still incomplete or not modeled at all.

2.3 Server – Client Functional Model

The virtual globe applications of interest in the context of this work are internet based and implement the server – client model. They are using a client application which is responsible for requesting data according to user criteria and location. A web server is submitting the required data. The communication between server and client is achieved by web services. Real time data transmission can be accomplished by AJAX and SOAP technology (Huayi, et al, 2010). Many users can request the

same data or different ones, but all of them are accessing the same database. The latency between the retrieved models and data and the real-world information mainly depends on the update rate of the database but may also represent different levels of accessibility offered by the data provider where high latency characterizes free access but almost real-time access is only commercially available.

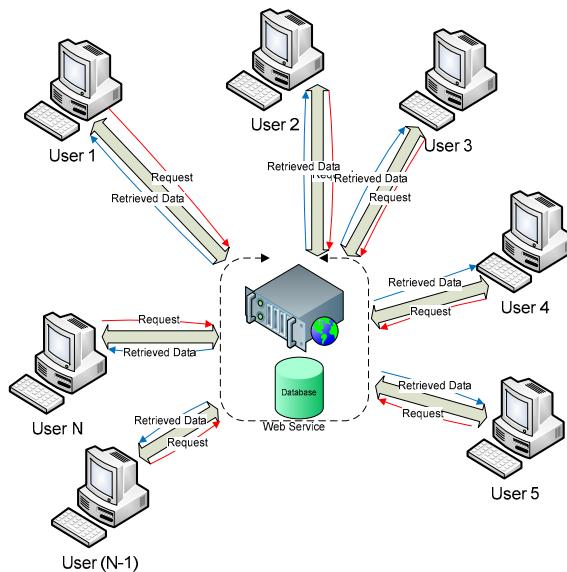


Fig. 3 Server – Client model of virtual globe systems

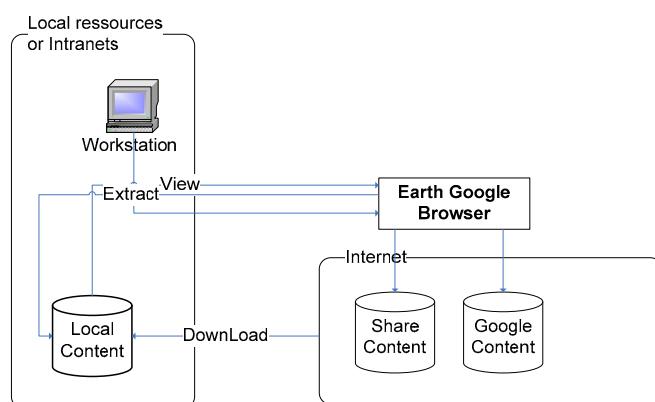


Fig. 4 Functional Block of a virtual globe (Google Earth)

According to the above functional block, the path of the information is only from the server to the clients, the latter only submitting requests for data to be retrieved from the server.

The latest versions of user's side client application have the ability to cache data, or even to load external data, locally stored on user's system. The local data offer the ability to create and maintain a local database thus enhancing through customization the experience of the virtual globe system. However, this local database can neither be shared nor it can be used to update the public database. Each user's client can only visualize his/her own data in combination with the publicly available data.

2.4 3D City Modeling

The modeling of the majority of cities of the world is incomplete or not available at all. The update of the GIS databases is taking place continuously but at low rate, considering the vast amount of information and the considerable computation power that is required to reconstruct a three dimensional model of the whole scene of a city. In this work we propose an alternative model of a collaborative GIS system, based on data distribution and gathering from multiple users, taking advantage of internet technologies and communication.

3 Collaborative Systems

3.1 Principles of Web-Based Collaborative Platforms

Collaboration is a process that two or more people work together in a project or to achieve a common goal. There are tasks that can't be accomplished individually. Nevertheless collaboration can help to carry out the task faster and more efficient (Dragonas & Doulamis, 2007).

Several methodologies have been presented in the literature. According to the most popular methodologies the main requirements of a collaborative platform can be reviewed on the followings topics:

- Information Systems of georeferenced scenes. Google Earth has been chosen for this purpose.
- Communication Tools. It is allowed communication between group members.
- Geometrical Representations. A software application that facilitates the visualization of a 3D building model among all the group members.
- Knowledge Representation Methods. Generic rules that are stored in an information database and describes conceptual 3D models.

Considering the above typical principles of a collaborative platform, a web based platform will be presented in the next sections.

4 State of the Art

In the field of geo-referenced systems and three dimensional modeling several approaches have been presented.

4.1 *Geo-referenced Systems*

The steep increase in processing capabilities of modern day computers has made possible the efficient manipulation of huge amounts of data in the form of pictures, videos and three dimensional models. It is beyond any doubt that, the internet has made widely known a great number of new applications and services. The last decade many geo-referenced systems have been introduced. In the beginning, the GIS applications were two dimensional, the first being included in the 1998 version of Microsoft Encarta. Afterwards, three dimensional data were added to GIS systems. In 1999 Chen introduced a three dimensional urban planning system for Los Angeles (Chen, 1999) which was a combination of VR technology and GIS systems. Web services have also been part of with GIS systems for urban planning in (Drummond & French, 2008).

In 2003, a generic framework for dynamic environment of three dimensional modeling was presented using java (Huang, 2003). An interactive framework for public participation in urban planning was presented by Huayi, Zhengwei and Jinya (Huayi, et al. 2010). Their approach is based on AJAX and XML technology. In addition, several approaches have been introduced by commercial companies. Google presented in 2004 Google Earth, which became widely popular and, subsequently, other companies have followed, including Microsoft and others. Some of the applications were designed for public use. In the field of urban planning commercial applications have also been introduced but their use is targeting expert people. The reconstruction and modeling are based on variety of parameters, like height, map of street, aerial, satellite photos and more.

4.2 *3D City Modeling Systems*

Modeling a city or scene requires a lot of information to be gathered. Data for 3D reconstruction process can be gathered by the photogrammetry process, based on geo-referenced photos captured by cameras on the ground. A set of sensors are scanning the building and the scene (Koutsoudis, et al, 2007). The result of the process is very accurate but its main disadvantage is that it requires special equipment and numerous measurements.

An alternative methodology of 3D reconstruction is based on LIDAR (Light Detection And Ranging) (Lan, et al, 2009). It is an optical remote sensing technology that measures properties of scattered light to find range and/or other information of a distant target. The prevalent method to determine distance to an object or surface is to use laser pulses (Wikipedia, LIDAR).

On the other hand the 3D model can be generated by an engine. A declarative engine for generating a 3D large scale scene was introduced by Larive & Gaidrat (Larive & Gaidrat, 2006) – (figure 5).



Fig. 5 3D Large Scale Scene of 17.362 buildings (Larive & Gaidrat, 2006)

In addition several commercial applications have been presented in 3D reconstruction of georeferenced scenes. They can generate the model based on rules and on image recognition of satellite and aerial photographs.

Georeference in the global positioning system is only one of many approaches used. Regardless of the georeference approach used, there are plenty of applications that can be used for urban planning. Most of them are stand-alone applications and have a visualizing browser. The user can build his/her own scene and navigate in it. These applications are using a visualizing module typically based on OpenGL. With the aid of such applications, the user can build an urban plan by selecting and modifying existing models, thus constructing virtual worlds with their own rules.

4.3 *Wiki-Mapia Platforms*

Besides the three dimensional GIS systems, there are vulnerable of collaborative GIS systems using two dimensional environment. These approaches are known as wiki-mapia platforms. They permit the user to interact with a two dimensional map. Consequently, all the users of the platform are using the same resources, i.e. the same map. The user can add georeferenced content on the map which is instantly shared to the whole community of the platform. Some examples which support these platforms can be multimedia, place markers, or hypertext content. The wiki-mapia platforms function over web services and typically use AJAX and XML technologies to communicate with the user. The majority of the most known platforms are using the API of known GIS systems for the visualizing process.

5 Large Scale 3D City Modelling Using Google Earth

Google Earth, introduced in 2004, was the first virtual globe based system and large amounts of multimedia data were added in a small period of time to its database. Other companies like Microsoft, followed by releasing their own applications. Each one of the prominent GIS systems created its own community and an online public library of georeferenced content. These libraries were containing mainly three dimensional models of buildings. The models were built by the users with appropriate software tools and subsequently placed on the GIS system. The inaccuracy of the model and the inability of massive generation of three dimensional models of real buildings made the reconstruction of scenes or cities a processing task with high overhead. During the end of 2009, Google introduced a new tool for reconstructing a 3D model of a building. It was the Google Building Maker, a simple web based tool that allowed the user to reconstruct the model of a building from multiple satellite and aerial images within minutes. In our approach, we adopt Google Building Maker as the initial tool to design and build the 3D model of a building.

5.1 Our Approach

The current work presents the proposal and implementation of a *collaborative web-based environment for georeferenced 3D city modeling*. Our approach is using Google Earth platform. The environment offers the following functionalities:

- Construction of custom 3D building models.
- Refinement of the custom 3D building models to improve realism.
- Placement of custom 3D building models in a georeferenced scene.
- Navigation among the building models as well as in the corresponding georeferenced 3D scene as a whole.
- Publication and sharing of the custom models and scenes with the other users of the environment.

In order to achieve the aforementioned functionalities a series of tasks had to be accomplished. In particular, initially, a custom GIS collaborative database had to be built to store and maintain all custom constructed models. Next, the database was populated by building models comprising 3D scenes. These constructed models were contributed by a team of student volunteers.

A working region was set and each student was assigned the construction of the model of a building within the region. Google Building Maker was used for the initial design of the models and Sketchup was used to correct possible errors with respect to the corresponding building.

Moreover, the texture of 3D models can be improved with higher quality photographs or aerial images from the area.

By the completion of each building model, it was uploaded to the collaborative database and all the community was able to access it, visualize it and navigate to it. For the navigation process the Google Earth plug-in component was used.

The communication between the local server and the clients but also between clients and Google's server was achieved by AJAX technology. Google Earth API was used in order to manage the collaborative database and filter the data. In the figure 6 the steps of the whole process are been presented.

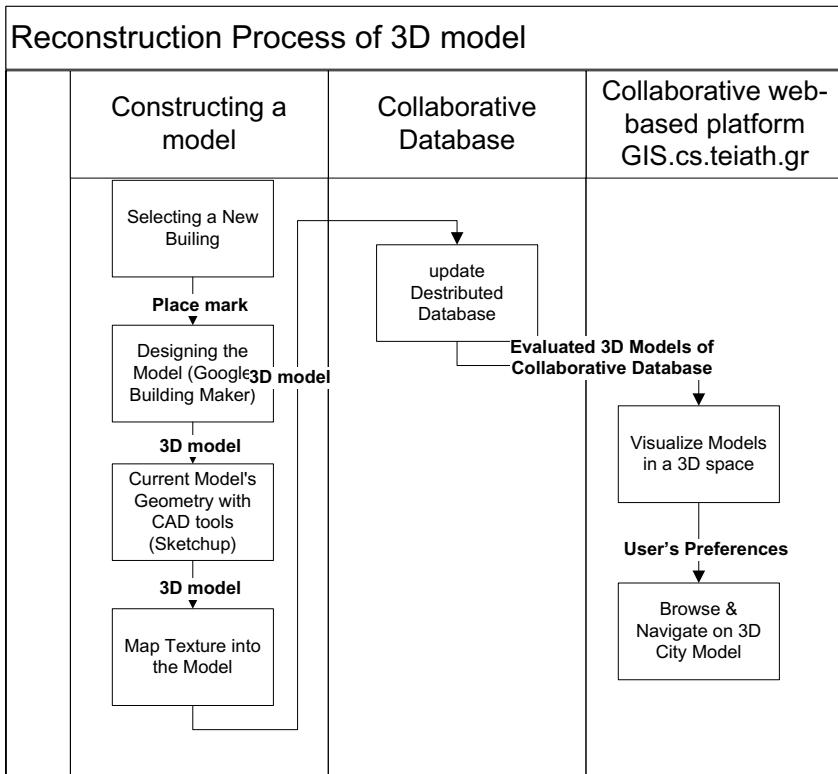


Fig. 6 Collaborative 3D city development process

Users were called to design 3D models of building in a desired specified area. Our web platform interface was visualizing the region in a 3D environment. All the available models were visible to all users. Users could freely navigate into the scene and select a desired free building to be constructed. Building was reserved by placing a placemark over the building. The placemark was announcing to the community that the building has been chosen for reconstruction. The model of building was reconstructed by Google Earth Building Maker. If it was necessary modifications and improvements was taking placing in order to improve realism. Afterwards the 3D model was uploaded on the collaborative web-platform and became visible to all the members of the community.

Our approach has been based on the server – client architecture, using a virtual globe system. On the side of the local server a collaborative database was built.

The users could access the collaborative database through a web based interface. A more detailed functional diagram is displayed below. Each module is presented in the figure 7.

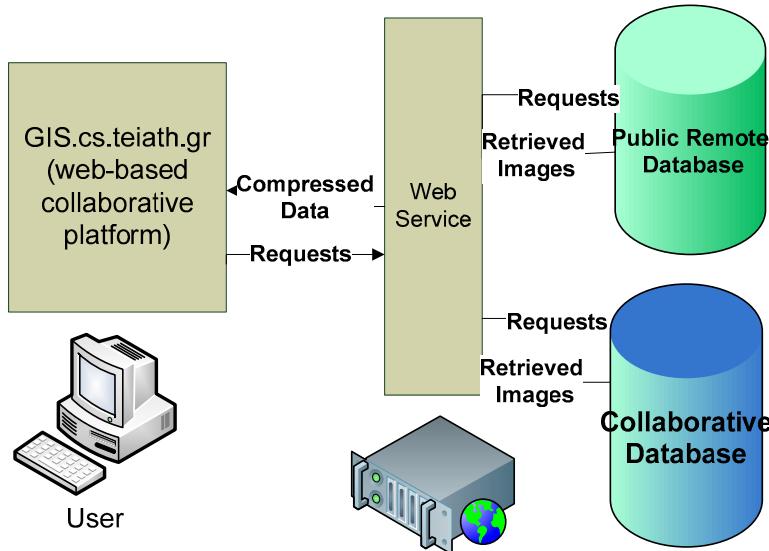


Fig. 7 Block Diagram

A collaborative local database was used. It was accessible by all the users of the platform. A programming component released by Google was used to browse and visualize the content of the GIS databases. The users were able to see their team's models and all the content of the public database through a web site. Any item that was added on the database was instantly visible by anyone.

For the reconstruction process of the city we have restricted the working area in a small region, by marking it on the database. The models and any other geo-referenced context had to be built on the user's side.

5.2 Reconstruction of 3D Building Model

Comparing all the existing tools for construction of a 3D building model, Google Building Maker was selected due to its user's interface and its reconstruction speed. With this tool a model can be designed within a few minutes. In addition texture can be added on the sides of the building model from the real satellite or aerial photographs. Other tools such as Sketchup or CAD tools present certain advantages but they lack the ability to automatically match the texture to each side of the building model.

Google Building Maker was not available for all regions in the world. In particular, it was available in cities with incomplete models or in ones having only a

few 3D models. For our purpose we chose Athens (figure 8) and a particular region nearby Technological Educational Institute of Athens.

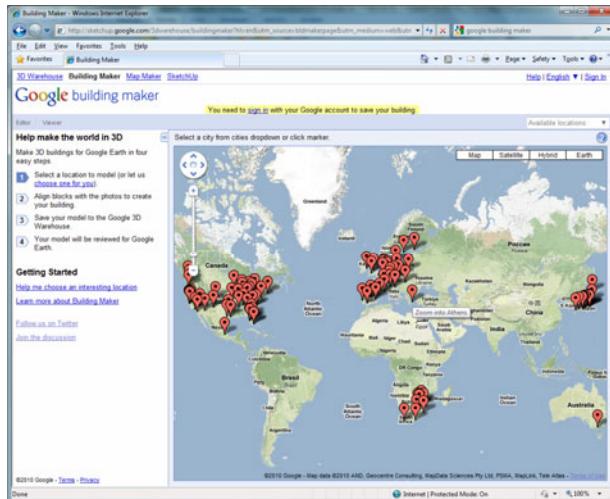


Fig. 8 Choosing a region

Users could choose a building to construct by placing a red placemark on it. With a blue circle existing models of public Google's Earth database buildings were marked (figure 9).

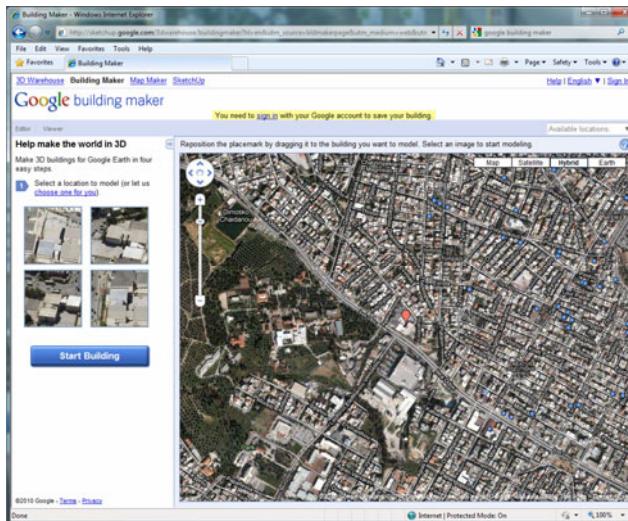


Fig. 9 Selecting a building for reconstruction

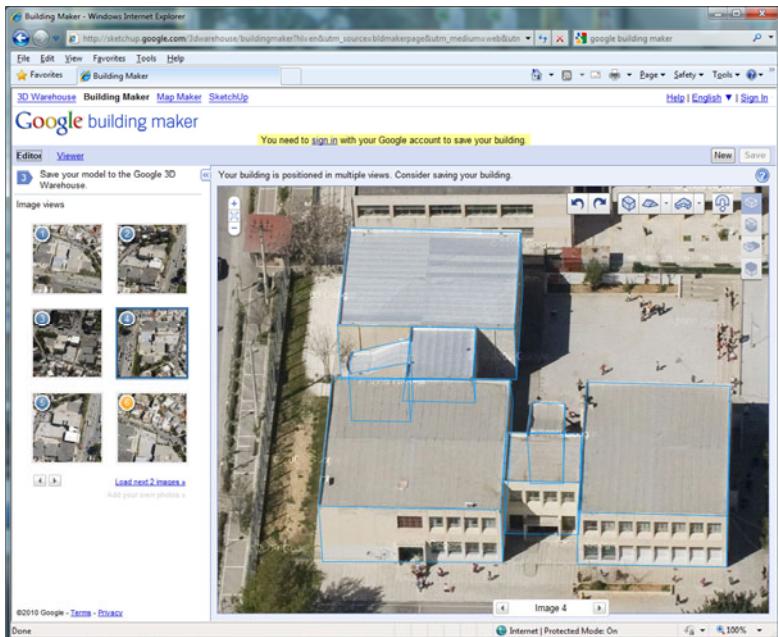


Fig. 10 Constructing a model with “Google Building Maker”

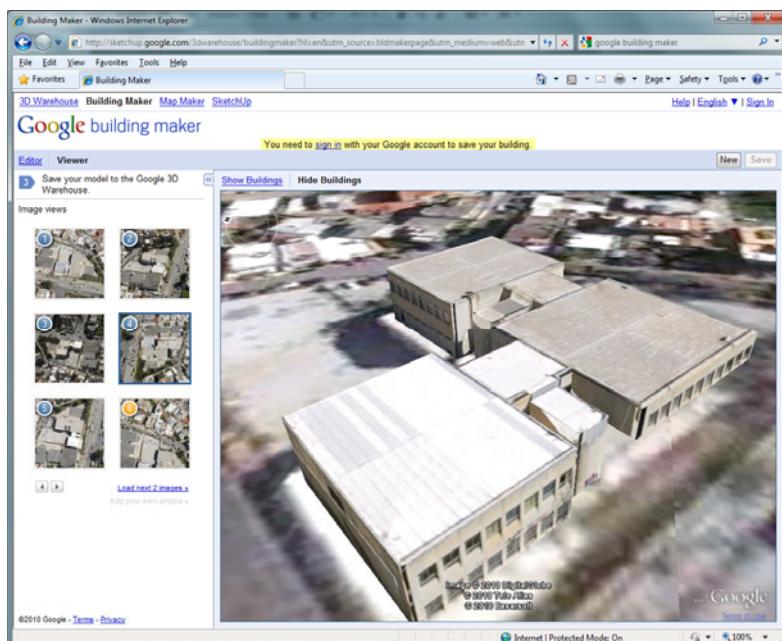


Fig. 11 3D model of building (“Building Maker” process)

The process of designing a model was based on bounding geometrical objects. Bounding boxes and bounding pyramids were used to construct the model. The bounding boxes had to cover and shape the building with best possible way. The GIS platform was providing the necessary photographs for a wide variety of points of view of the desired building. An arbitrary number of photos could be used for the design. An increased number of photos led to increased accuracy. The building was designed from the image of the first point of view and corrections were made by the next views. The 3D reconstruction process was done by the GIS system and a three dimensional geometrical model was created. Texture images were applied automatically on the sides of the model, from the satellite or aerial photographs. Each model that was created by the above process was placed on the public internet library of Google Earth and was accessible by anyone.

5.3 City modeling

Exploiting the possibilities of the above tool, a collaborative environment was created, based on an appropriate scene database, in order to gather many models of buildings and create a model of a localized scene or a whole city. The collaborative GIS system that was designed and used is shown and analyzed in processes in the diagram of figure 12.

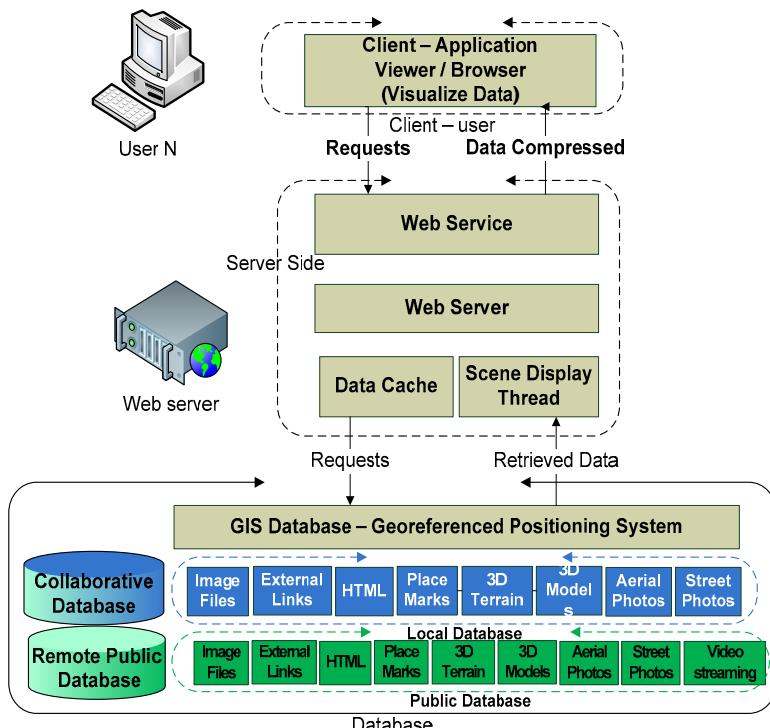


Fig. 12 “Gis.cs.teiath” Detailed Block Diagram

The remote access of the database and the distribution of data were achieved through http web service. In particular, a web site was built and the database was hosted in a SQL server. A Google Earth plug-in component was used to visualize the data and create a 3D dimensional world with OpenGL-DirectX. Google has made public an API for Google Earth use. Using AJAX technology and the Google Earth API, the public database was filtered and the shared database was loaded and visualized on Google Earth window.

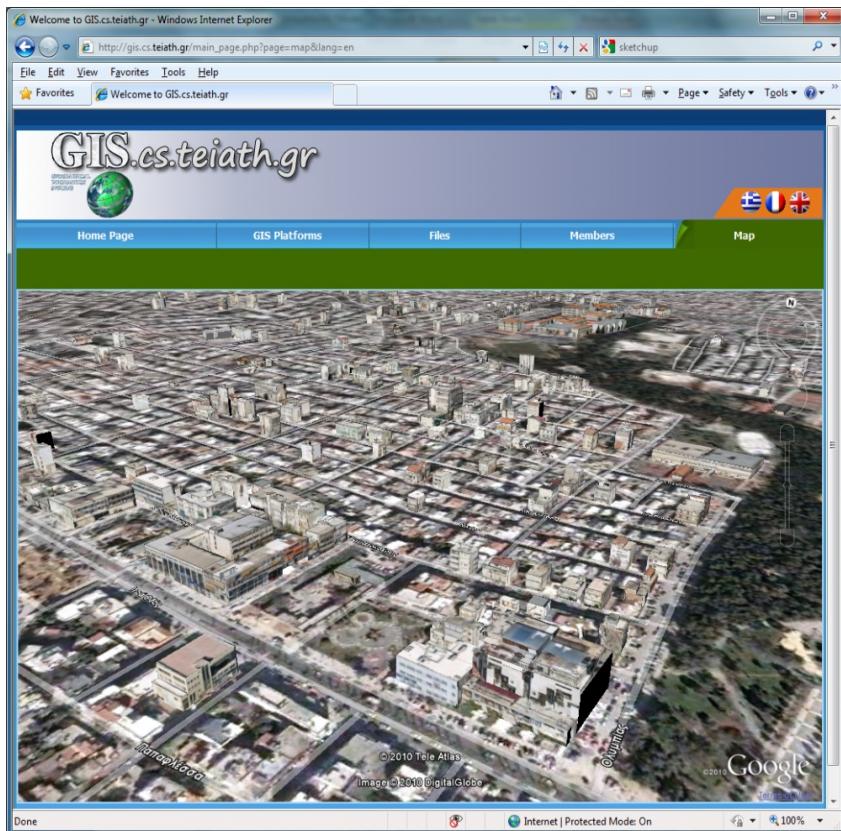


Fig. 13 “Gis.cs.teiath.gr” environment – 3D models of buildings in a collaborative

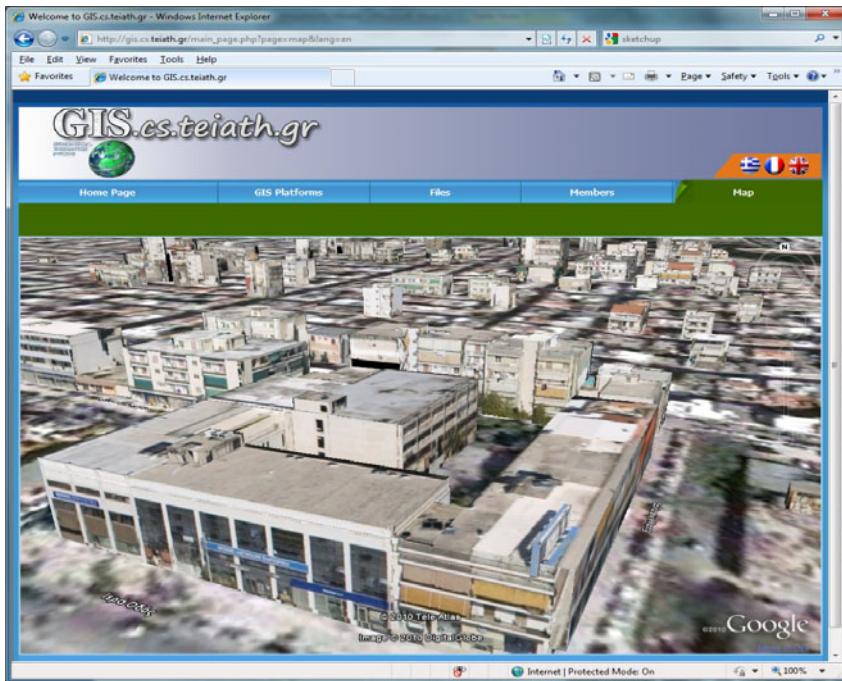


Fig. 14 Experimental set of 3D models of building

Each user could access other members' buildings. All members of the team were sharing the same map, working on the same region. They could navigate in a three dimensional space, by zooming in or out and by flying around the buildings. A mark could be placed on a building that was under construction, by the user, announcing that the specific building was in designing process. Each model was constructed using the Google Building Maker. All members of the team collaborated in order to create the 3D model of a scene like the one in figures 13-14.

5.4 Evaluating Process

As soon as a building model has been constructed, it was placed on the map using the global positioning system. Some building models were good copies of the original buildings whereas some others were poor or incomplete. Each building that was designed had to be evaluated before it was selected by the visualizing module.

Google Earth and Virtual Earth both have a public online library with multimedia content, including 3D models of buildings. Each user was free to build and upload models on these databases. However, each model had to be evaluated before it was placed on the virtual globes system. Only the detailed models and the ones that bear correct geo-reference was allowed to be visible on the GIS systems.

In our approach, each building could be evaluated in a scale from 0 to 10. The evaluation was taking place by a human expert, considering the satellite images of different point of views and 3D model of the buildings. In future versions of our platform, users will be able to evaluate the building of others. The buildings featuring correct outline and accurate placement were allowed to be visible in the Google Earth component.

6 Discussion

6.1 Our Approach

In this paper a collaborative web-based platform for city modelling was presented. Our initial requirements and goals were accomplished.

- A common working environment was designed. It was based on a custom GIS platform. All the members of team were using the same tools and share the same data. The working region had been specified and all the models of the scene were access by anyone.
- Common CAD tools and a Web-based application “building maker” was used from all the users to reconstruct the models
- A tree dimensional environment was freely available for navigation and exploration

The goal of the team was to develop the 3D models of a bordered area next to Technological Education Institute of Athens. In a small period of time about 440 models of building were reconstructed.

6.2 General Discussion

By 2004 and for a long period of time there were many regions around the world without any 3D building models or with only a few ones. Once the Google Building Maker was introduced by Google in the midst of October 2009, hundreds of three dimensional models of buildings appeared on the GIS system. Many of them have been evaluated by the team of Google and they appear on Google Earth too. A collaborative database of a GIS system, made possible the modeling of a geo-referenced scene or a city. The participants of the experiment could choose a place to build, or add material hypertext, data or image on the map. The result of the collaboration was to build a three dimensional scene with about 440 buildings in a very short period of time. Each model was completed within minutes. The whole city could be modeled in this way distributing the work to all the participants of the team.

6.3 Buildings and Obstacles Overlapping

On the other hand, building a model with such a tool, although it is a relatively easy task, the result is not always the desired one. The accuracy of the model and the detail of the texture are not always the expected one. There are cases that the satellite or aerial photographs are not clean, or there are obstacles in front of the building. The obstacles can be anything, like trees, big cars, etc., the most common problem being the partial overlap of the buildings. This problem is very common in cities, where one building is close to the other. In the figure 14 there is a green part at the bottom of the image texture of some buildings. This occurred due to tall trees in front of the building.

If there is no spare space between the buildings, one solution is to create the 3D model of the whole complex of buildings. Hence, the whole building complex has to be modeled up to the point a clear area is reached. Likewise, if there is a small area between the buildings, then the texture among the nearby buildings will be erroneously applied on the sides of them.

Problems with low quality of images or texture images with noise (obstacles mapped on them) can only be corrected by replacing the image with a real photograph from the proper aspect of view. All the models are written in COLLADA graphics language. It can be changed either manually or using a CAD tool compatible with COLLADA. Google Sketchup is one famous tool for that purpose. Moreover, a faulty model may occur if the bounding boxes or pyramids do not fit correctly on the satellite or aerial photographs. In that case the produced 3D model is poor in quality and typically rejected during the evaluation process.

7 Conclusions

In this paper we have proposed and implemented a prototype system for a collaborative GIS system. The data content comprised 3D models, hypertext and all the material supported by Geography Markup Language (GML) as it has been adapted by Google Earth. The initial content was constructed by volunteer students and was subsequently carried out and evaluated by the instructor. Despite the fact the refinement and evaluation process has revealed numerous discrepancies, the overall experiment has highlighted the value of collaborative model construction. The amount of effort required for the three dimensional modeling of the region that was covered by the experiment would be prohibitive for a single user. The implementation has shown the efficiency of the proposed design, under relatively demanding loads of concurrent access to the local database, as well as the more than adequate coverage of all desired functionalities. Last but not least, the prototype system and its use by the team of students have contributed a number of high quality 3D building models to the Google Earth community.

8 Future Work

Once the construction of a scene is completed, a question arises: how could we navigate or fly inside the scene, watching all interesting points while avoiding the rest of the information. The management and informative presentation of a large scale three dimensional scene is an issue which generates a great deal of heated debate one of the major problems of large city models. Beside the 3D models, if semantic knowledge is added on the scene, the user could be guided by the system to see only the desired information of the scene.

Moreover algorithms of best view can be applied on such a system to automatically guide navigation in the three dimensional environment, following a camera trajectory based on user's preferences.

Applications for a city 3D modeling system can be found in a variety of fields. Virtual tours can be achieved in places that one has not or cannot visit. In the field of tourism, a 3D city helps to develop and present a region to the rest of the world. Furthermore, simulations can be based on 3D city models, with respect to physical phenomena, evacuating processes, etc.

References

- [1] Chen, C.: Information visualization and virtual environments. Springer, London (1999)
- [2] Doulamis, N., Dragonas, J., Doulamis, A., Miaoulis, G., Plemenos, D.: Performance Evaluation of User's Profile Estimation Methods in a Declarative Collaborative Framework. In: International Conference 3IA 2009, pp. 129–138 (2009)
- [3] John, D., Nikolaos, D.: Web-Based Collaborative System for Scene Modelling. Studies in Computational Intelligence, vol. 181. (Print)1860-9503. Springer, Berlin (2009) ISSN 1860-949X
- [4] Drummond, W.J., French, S.P.F.: The future of GIS in planning: Converging technologies and diverging interests. Journal of the American Planning Association 74(2), 161–174 (2008)
- [5] Huang, B.: Web-based dynamic and interactive environmental visualization Computers. Environment and Urban Systems 27(6), 623–636 (2003)
- [6] Huayi, W., Zhengwei, H., Jianya, G.: A virtual globe-based 3D visualization and interactive framework for public participation in urban planning processes (2010)
- [7] Anestis, K., Fotis, A., Christodoulos, C.: On 3D reconstruction of the old city of Xanthi. A minimum budget approach to virtual touring based on photogrammetry. Journal of Cultural Heritage 8, 26–31 (2007)
- [8] Hengxing, L., Derek, M.C., Chenghu, Z., Ho, L.C.: Rockfall hazard analysis using LiDAR and spatial modeling (2009)
- [9] Larive, M., Gaidrat, V.: Wall Grammar for Building Generation. In: Graphite 2006, Kuala Lumpur, Malaysia (September 2006)
- [10] Michael, J.T.: Google's Geospatial Organizing Principle. IEEE Computer Graphics 27(4), 8–13 (2007)

- [11] Wilaiport, S., Arun, P.K.: A comparative Assessment of Giservices Architectures (2006)
- [12] Wikipedia (n.d.). Google Maps,
http://en.wikipedia.org/wiki/Google_maps
- [13] Wikipedia (n.d.). LIDAR, <http://en.wikipedia.org/wiki/LIDAR>
- [14] Wikipedia (n.d.). Virtual Globes,
http://en.wikipedia.org/wiki/Virtual_globes
- [15] Zhang, J., Gong, J., Lin, H., Wang, G., Huang, J., Zhu, J., Xu, B., Teng, J.: Design and development of Distributed Virtual Geographic Enviroment system based on web services. *Information Sciences* 177(19), 3968–3980 (2007)

A Computerized Method to Generate Complex Symmetric and Geometric Tiling Patterns

Ali Izadi, Mahdi Rezaei (Ghahroudi), and Azam Bastanfard

Abstract. This article contributes to a novel computerized and heuristic algorithm to make pretty geometrics used in traditional building patterns and tiling. Starting with some basic sketch or an ornamental pattern, it depicts how a designer may analyze the sketch structure and approaches to an analytical representation. In return, this analytical representation could be used to produce complicated pictures and schemes. The presented algorithm is capable of producing various complicated symmetric patterns as well as ancient Islamic geometric patterns that are called Star and Bracelet. Utilizing and expanding such patterns, the algorithm would be able to generate wide variety of divergent sketches for both monochrome and coloured patterns. The algorithm has been successfully performed and tested on the implemented software-GeoPat. Finally, the method has been applied on different types of sample patterns and the interesting results have nicely been depicted.

G
Geometric Patterns, 1
I
Islamic Patterns, 1
O
Ornamental Sketches, 1

S
Star and Rosettes, 1
Symmetry Patterns, 1
T
Tiling Methods, 1

Ali Izadi
Islamic Azad University, Maragheh Branch
e-mail: izadi@iau-maragheh.ac.ir

Mahdi Rezaei (Ghahroudi)
Qazvin Islamic Azad University (QIAU)
e-mail: rezaei@QIAU.ac.ir

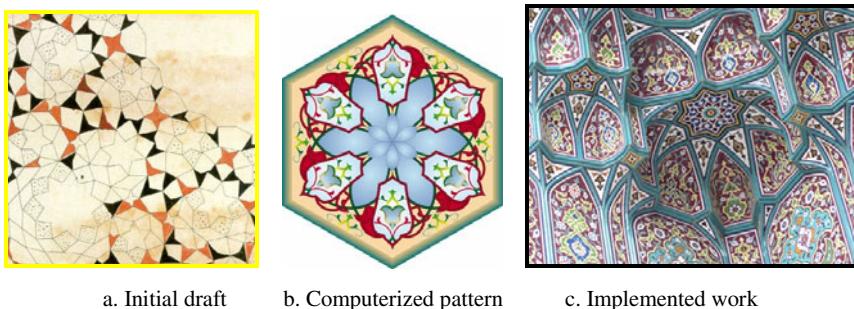
Azam Bastanfard
Karaj Branch, Iran
e-mail: bastanfard03@gmail.com

1 Introduction

Innovative merging of geometric shapes and patterns is the most impressive feature of traditional buildings, especially in ancient Islamic and Greek era. All master artists in the field aver that having a good sense of analytical geometrics is an essential to gain a successful design so to reach a grand tiling pattern. Recent studies prove the important role of mathematicians who have taught analytical geometric topics to architect designers; it reveals that they virtually played a significant role to develop overall visage of both ancient and new buildings. In this regard, medieval Islam literature is the major source to study and find the relations between mathematics and architecture.

Actually, geometric patterns consist of the combination of some basic diagrams such as triangles, tetragons, pentagons, octagons and stars. Repetition, rotation and combination of such basic shapes along with specific patterns such as herbal and animalistic patterns can create beautiful forms and lattices.

These forms can numerously be found in majority of Islamic artistic works as in architectural ornamentation, consisting of tiling, plaster work, mirror work, Muqarnas¹ or stalactite work (Fig. 1.) and brick works, as well as traditional arts like metal work, lattice work, textiles, and wooden windows.



a. Initial draft b. Computerized pattern c. Implemented work

Fig. 1 Some samples of Muqarnas (stalactite work)

Islamic Geometric Patterns, especially star patterns, have stunning artistic geometric forms. These patterns are known as one of the most beautiful regular figures, have ever been made by human and that is why they are still applying in handmade buildings. As a whole, the existing Islamic patterns are taken from natural elements and phenomena. Consequently, such patterns are nominated and classified by emphasizing the elements or natural phenomena from which they are taken.

Among Islamic patterns, “star patterns” are one of the most famous traditional ornamental designs in the world. These patterns are a concord combination of art, mathematics, spirituality, symmetry, balance, and creativity [6, 9] which some

¹ Muqarnas is a type of corbel used as a decorative device in traditional Islamic and Persian architecture.

researchers have attended and developed ideas for the production of these patterns by computer modeling.

With the exact look to star patterns, some precise and complicated mathematical secrets can be revealed; however, as the time has passed, most of the designing techniques been gone. A complicated ancient architectural is underlying such amazing geometrics and the original designers have kept their techniques secret. On the other hand, the final implemented works just give a small amount of information about such complex patterns as well as the hidden ideas behind the layout.

Now re-finding these innovative techniques is a concern for today's researchers in the field; we hope a gradual search and process among ancient artists and mathematicians' notes and works lead us to some improvements. In this regard, some of active scientists have found some parts of these lost techniques [11, 19, 20]; however, there are still a lot to do.

In this article, we will develop a robust algorithm to gradually make several samples of the mentioned patterns which can provide several Islamic geometric patterns. In the created sketches by the algorithm, some patterns like "*bracelet*", different kinds of "*Shamseh*" and "*Star*" will come into view. Starting with simple implementations and then expanding to complex patterns along with the possibility to create new figures are the superior characteristics of the proposed algorithm and software.

2 Related Works

In Grunbaum and Shephard's method [13], a deep and complete program has been used on group theory to study periodic star patterns. They derived a strong set of mathematical tools to analyze patterns in symmetric groups and periodic groups, while patterns are manipulated on the surface, skillfully.

Ostromoukhov [6] has extended Grunbaum and Shephard's analysis in 17 wallpaper group styles and some stapes are contributed for artists and designers to make symmetric ornaments. In Abas and Salman [10] method, the group theory is transferred to a library covered by historic patterns. In another work [12], they are contributed the increasing distinct patterns from available mathematical tools for industrialists who made them. In all above cases, the research is extensively analytical and the proposed methods often describe how the new star patterns could be made.

Kaplan and Salesin [4] are made new sketches by filling the tiles with some pieces of star patterns on the surface of tiling. They give a surface of elements which can accept the parameters, drawn in historic samples. After that, using a deductive algorithm, they try to complete the provided sketch by filling the irregular polygons. Jay Bonner [8], an architect who has extensively studied on Islamic Star Patterns, extends a symmetric performance of star patterns on an extended surface of tiling. He is also the creator of Geodazzlers [3] which is a set of foldable papers decorated with star patterns. For a better illustration, several techniques are shown in [7]. Bonner's work is considered as a reference for designers, but it is not appropriate for software designers. Dewdney offers a complete method for "*constructing designs*" based on the reflection of lines through a regular order of

circles [11]; however, this technique just could be used to construct some famous but limited patterns. Dewdney realizes that his method needs several natural leaps to reach the last pattern. Similar to Dewdney's method, Arabeske software [14] allows the users to construct star patterns.

In [5], Lee shows that a pattern should be extended by drawing pieces of lines from the middle of tiling edges. He illustrates polygons-in-contact technique. On the basis of Lee and Hankin works [9], Kaplan develops some software tools for tiling [1]. M. Valiente et al. [15] performs a calculation model to analyze samples of periodic patterns. This model firstly finds the contributed objects in the primary digital picture, and then compares them with each other, and finally calculates their symmetries relations.

Lu and Steinhardt [18] found out a wide usage of subdivisions in traditional Islamic design systems e.g. overlaid Penrose kites and darts on Iranian designs. P. Rasouli et al. [19] presented two algorithms to produce computerized Islamic geometrical patterns, called 8-ZOHREH and 8-SILI. Provided though their algorithm is pretty simple, the diversity of outputted patterns is rather limited. And recently, Cromwell [20] has described a tiling-based method to construct Islamic geometric designs.

Diving all top mentioned methods, it can be inducted that the majority of them are developed based on custom tiling technique, and in order to create an especific pattern, relying on the experience of a tile worker is needed. In contrast, our recommended method does not suffer from such limitations and can perform various kinds of heuristic patterns, individually.

3 Proposed Method: Creation of Key Symmetry Patterns

As can be seen in the following pictures, Islamic patterns are some extended forms of geometric patterns, whether simple or complicated. In this method, we determine "*fundamental area*" repeated in the entire sketch so we create the entire sketch by drawing the fundamental area in a simple way, using geometric technique. Finding the fundamental area or the "*fundamental region*" is not a new problem. Lalvani [16] mentioned this problem in 1989; however, the important problem is the way of finding the fundamental area.

The initial and basic step of the suggested method produces the figures just by a independent variable parameter. First, this independent variable determines an angle of a tetragon in which the fundamental area can be settled. Then the method is expanded for tiling techniques that will be discussed later in the paper. Regarding tiling there are some interesting works has been done before by Grunbaum and Shephard in their book: "Tiling and Patterns" [21].

There are different ways to extend and transfer a pattern on a surface. In order to extend a pattern in a symmetric manner, sometimes the pattern may change or not, depending on the type of symmetry which is used. As a whole, various kinds of symmetry can be divided into four groups, including *transfer symmetry*, *reflective symmetry (mirror symmetry)*, *periodic symmetry*, and *compound symmetry*.

Different kinds of symmetries are discussed in [2], and we have used them to produce the entire sketch as well. Symmetry means balance or repetition of a part or a similar form. But, symmetry extension is much more than using it in architecture and beautiful patterns [12]. Group theory in one-dimensional symmetry indicates that periodic patterns can be analyzed in seven different ways. More information to determine the sort of practical symmetries could be found in [10].

Similarly, in two-dimensional symmetrical periodic patterns, there are seventeen sorts of different patterns can be determined and formed [2].

3.1 Sketch Analysis within the Fundamental Region

Considering the sample sketch in fig. 2, it can be seen that there is a pattern in this sketch, the entire of which is made by the repetition of this pattern. This pattern can be put in a tetragon with the length of ' x '.

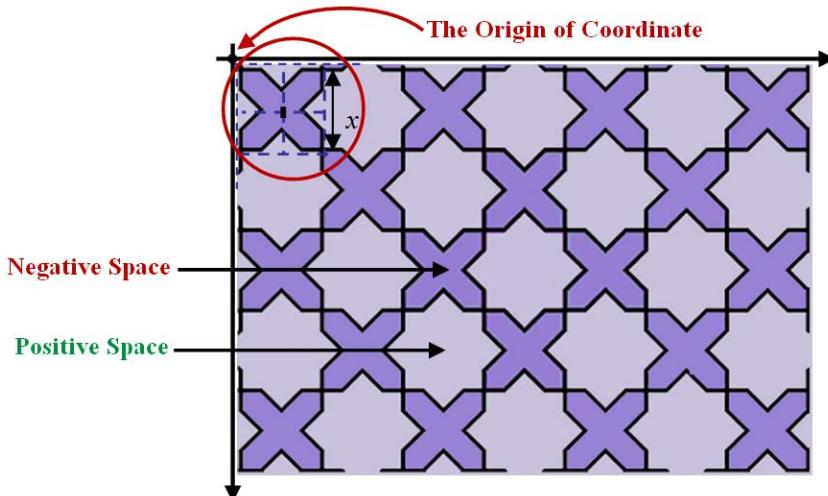


Fig. 2 Complete sketch made of by sample pattern repetition.

If two or more patterns join each other in some ways, they are called *Node*. The determined pattern in fig. 3 is called “*Bracelet*” which forms positive space of the sketch in fig. 2. The negation space of the sketch in fig. 2 is called “*Octa-Shamseh*”.

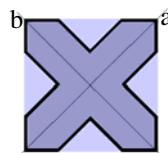


Fig. 3 The initial “*Bracelet*” pattern to create fig. 2 (a and b are the symmetry axes)

Noticing the pattern utilized in fig. 3, it can be understood that the pattern is made of a smaller parts which is shown in figure 4-a. and called fundamental region.

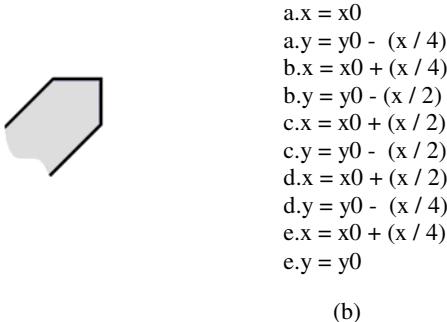


Fig. 4 Sketch Analysis via the fundamental region

This figure has five vertexes (a, b, c, d, e), the coordinate of which can be calculated easily in terms of the x as figure 3-*b*. The parameter x is the *side length* of the tetragon which is determined in fig. 2, the second and third parameters ($x0, y0$) are tetragon centre coordinates which depend on the origin of the coordinate and parameter x .

Fig. 5 shows the steps to find vertexes of fig. 4. This is done by procedure *FindVertex*. As a note it is not needed to determine all five vertex; in fact, if the vertex a and b are available, so vertexes d and e could be determined via “reflective symmetry” to the line oc in fig. 4.

```

Procedure FindVertex (x , x0 , y0);
Begin
  VertexA0.X := x0
  VertexA0.Y := y0 - (x div 4);
  VertexB0.X := x0 + (x div 4);
  VertexB0.Y := y0 - (x div 2);
  VertexC0.X := x0 + (x div 2);
  VertexC0.Y := y0 - (x div 2);
  VertexD0.X := x0 + (x div 2);
  VertexD0.Y := y0 - (x div 4);
  VertexE0.X := x0 + (x div 4);
  VertexE0.Y := y0;
End;

```

Fig. 5 Vertex Calculating Procedure

There are three input parameters for procedure *FindVertex*. The first parameter x is the length of sub-tetragon in fig. 2 which is extracted in fig. 3. The second and third parameters are x_0 and y_0 respectively which are center of tetragon based on original coordinate axes.

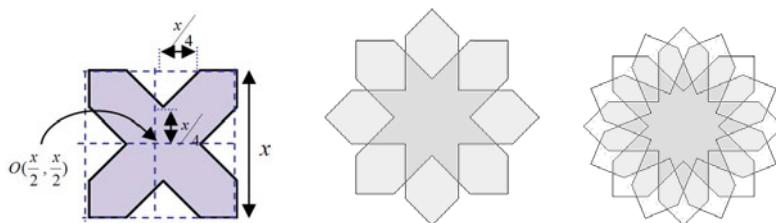
In general, in order to create an Islamic Geometric Pattern, there are three main steps that should be taken in to account.

1. Finding the basic sketch of which the whole pattern is made of.
2. Calculating relevant coordinate vertex for all basic sketches found in step 1.
3. Determining an appropriate arrangement system to create a complex geometric pattern, using basic sketches, found in step 1.

As a sample, we are going to introduce two most famous patterns called “Star” and “Rosettes”.

3.2 Rosettes

Assuming fig. 4-a as the fundamental region and performing the top three steps, fig. 6-a will be provided, with a graphical representation that shows how a more sophisticated patterns could be approached. Each individual region could be expanded by rotating and repetition of first sketch with different step angles of 90° , 45° and 22.5° . After obtaining coordinates of vertices for fig. 4 and drawing its sides, then rotating each vertex equal to 90° degree around the centre of quadrangle and repeating it, we can reach figure 6-a. This is called *rotational symmetry*. The procedure related to rotational symmetry is given in fig. 7.



- (a) Vertex rotation and repetition based on fig. 2-a in a surrounding tetragon and 90° step angle.
- (b) Vertex rotation and repetition with a 45° step angle
- (c) Vertex rotation and repetition with a 22.5° step angle

Fig. 6 Complicated patterns using rotational symmetry

```

Procedure Rotation_O (var p : TPoint ; const teta : Real );
var
  xx , yy : Variant ;
Begin
  xx := p.X × Cos(teta) - p.Y × Sin(teta) ;
  yy := p.Y × Cos(teta) + p.X × Sin(teta) ;
  p.X := xx ;
  p.Y := yy ;
End;
Procedure Rotation (var p : TPoint ; const teta : Real ; x , y : integer);
var
  temp : TPoint ;
Begin
  temp.X := p.X - x ;
  temp.Y := - p.Y + y ;
  Rotation_O (temp , teta) ;
  p.X := temp.x + x ;
  p.Y := - (temp.Y - y) ;
End;

```

Fig. 7 Rotational symmetry procedure around the centre point of (x,y)

Procedure *Rotation_O* in fig. 7, rotates *p* equal to *teta* degrees around source coordinate axes. Here parameter *p* could be each of vertex points in fundamental region e.g. *a*, *b*, *c*, *d*, *e* in fig. 4. In this procedure, *teta* and *p* are input parameters. Then second procedure-*Rotation*, rotates point *p* around (x,y) for *teta* degrees and uses procedure *Rotation_O* to perform it. In this procedure *p*, *teta*, *x* and *y* are input parameters.

If we transfer vertexes of fig. 3 around the centre of a $2x$ length tetragon, then rotate them with a 90° angle, so fig. 4-a will be obtained. Performing a similar manner with different repeating angles could be applied to reach the desired patterns. Looking back on fig. 2 it can be seen that the figure is provided by the transferring fig. 6-a with transfer vector of $(2x, 0)$. The negative and positive spaces in the existing sketch are determined in fig. 2. If fig. 6-a transferred with another transfer vector, other sketches also could be produced. This case depicts the robustness and advantage of suggested method, capable of generating wide diversity of forms just by a single fundamental form. In a similar fashion, we can reach to fig. 6-b if we repeat the vertexes eight times with a 45° rotating angle. Likewise, if we perform it 16 times with a 22.5° step angle, fig. 6-c will be produced.

In general, if we symmetrically rotate fig. 4 with the degree of $2\pi/n$, then the result will be a n-petals rosette which is surrounded in a regular n-gone.

Fig. 8 shows the relevant procedure to create a rosette flower. In this procedure, *x* is the first parameter for procedure *CreateRosette* and the *side length* of tetragon

```

Procedure CreateRosette (const x , x0 , y0 , n : Integer) ;
Var
  i : integer ;
  teta : Real ;
Begin
  teta := (2 * Pi) / n;
  for i := 1 to n do
    Begin
      MoveTo (VertexA0.X , VertexA0.Y) ;
      LineTo (VertexB0.X , VertexB0.Y) ;
      LineTo (VertexC0.X , VertexC0.Y) ;
      LineTo (VertexD0.X , VertexD0.Y) ;
      LineTo (VertexE0.X , VertexE0.Y) ;

      Rotation (VertexA0 , teta , x0 , y0) ;
      Rotation (VertexB0 , teta , x0 , y0) ;
      Rotation (VertexC0 , teta , x0 , y0) ;
      Rotation (VertexD0 , teta , x0 , y0) ;
      Rotation (VertexE0 , teta , x0 , y0) ;
    End ;
  End ;

```

Fig. 8 Rosette Creation Procedure

which is introduced in fig. 2. Second and third parameters-($x0,y0$) are coordinate centre of the tetragon that was described in fig. 4. ($x0,y0$) is the centre of regular n -gon which surrounds the rosette. n the last parameter, determines the number of petals. Finally, given these four parameters a n -petals rosette would be created.

Although procedure *CreateVertex* works for n -angles with 2^i sides while $i > 1$, it doesn't work properly for a limited number of flowers used in Islamic patterns. A revised version of this procedure-*FindVertex* is developed as fig. 9 to cope with this limitation to produce n -petals flowers.

By adding parameters $b1, b2, b3, b4$ the function becomes more flexible to find vertex points of the flower and more powerful to create n -petals flowers. Fig. 10 shows some selected samples. All depicted patterns in this paper are real outputs from our developed software-GeoPat which is written in Delphi with all above procedures built in.

```
Procedure FindVertex_Ex (const x , x0 , y0 , b1 , b2 , b3 , b4 : integer);
```

```
Begin
```

```
    VertexA0.X := x0 + b1 - b2;
    VertexA0.Y := y0 + b2;
    VertexB0.X := x0 + b4;
    VertexB0.Y := y0 - b3;
    VertexC0.X := x0 + (x div 2);
    VertexC0.Y := y0 - (x div 2);
    VertexD0.X := x0 + b3;
    VertexD0.Y := y0 - b4;
    VertexE0.X := x0 - b2;
    VertexE0.Y := y0 - b1 + b2;
```

```
End;
```

Fig. 9 Extended form of Procedure FindVertex (Fig. 5.)

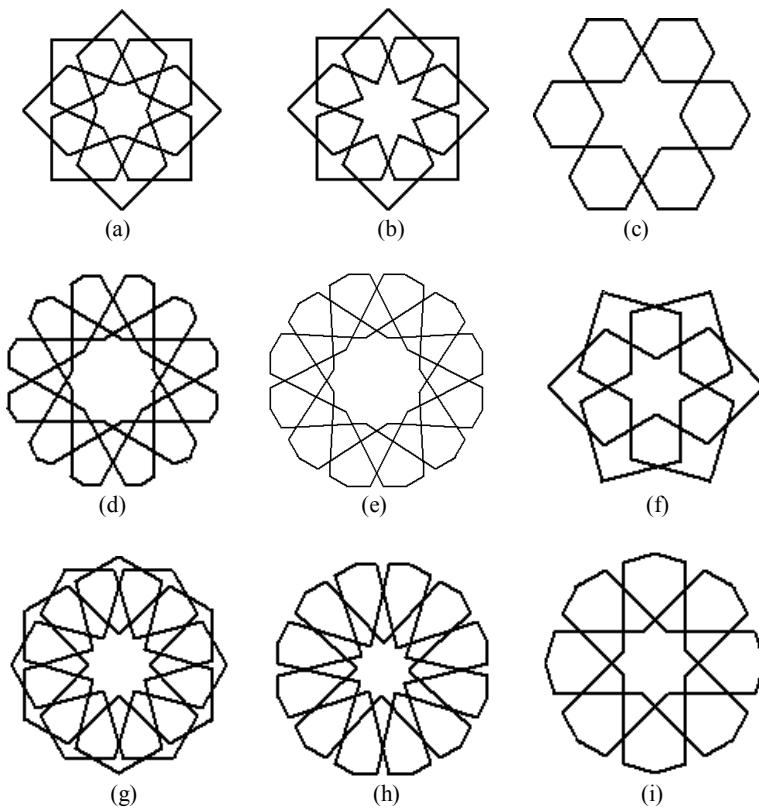


Fig. 10 Some basic fundamental patterns

If we extend the patterns of fig. 10 each of which with different transformation vector, then various textures could be created. Some sample textures are provided as fig. 11 and 12.

The textures in fig. 11 and 12 are created based on a feature from fig. 10. In addition by changing the transmission vector, some other textures can be created.

In GeoPat, The function Foodfill is considered to colorize the pattern, this option could be found in menu options.

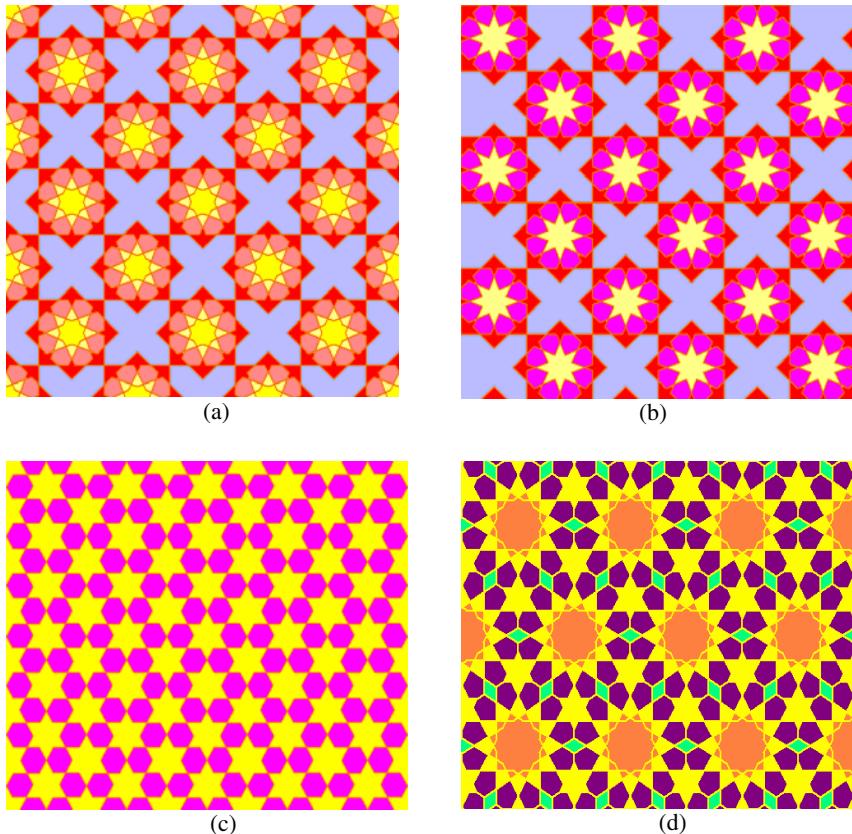


Fig. 11 Created patterns via transform of fig. 10-a, 10-b, 10-c and 10-d respectively

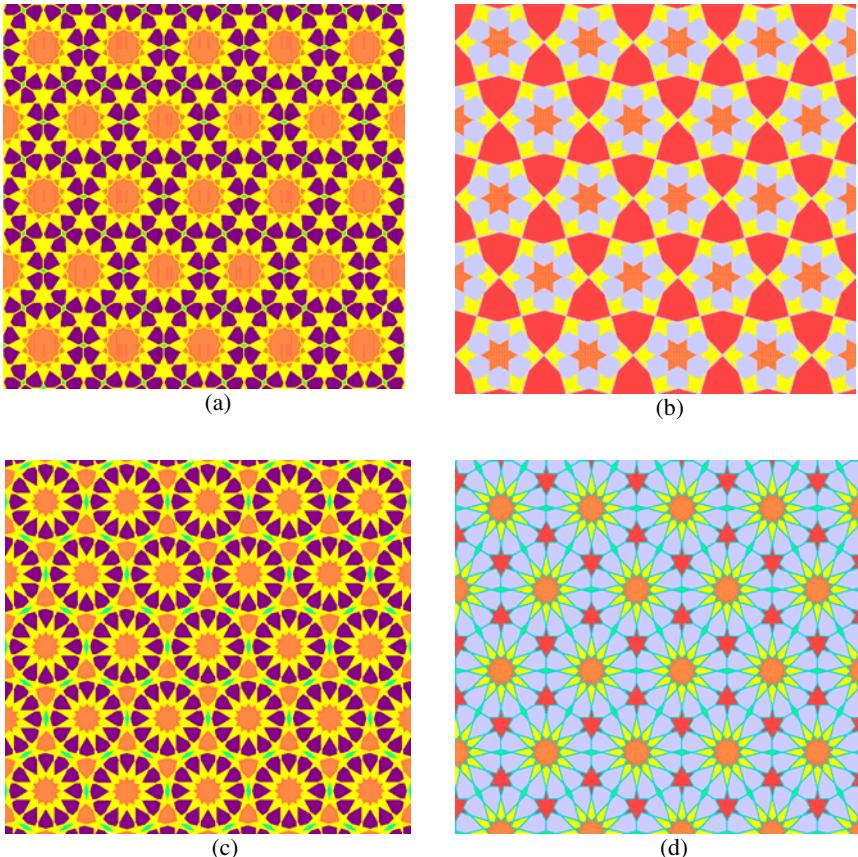


Fig. 12 (a) transformed texture via 10-e (b) Transformed texture via 10-f (c) Transformed texture via 10-g (d) transformed texture via 10-h

In order to have an initial sketch so to develop a texture we can perform tiling methods. Utilizing different tiling methods will creates different textures. Fig. 13 depicts how a tiling can create a texture. In fact, fig. 13-b is the same as fig. 12-d. actually tiling shows the transmission vectors in a clearer manner.

Fig. 13 is a result of expanding a tiling via a regular dodecagon. As can be seen there are also some regular triangles among these dodecagons. Filling the empty spaces of patterns by such triangles, we can reach more complex textures, similar to figure 14.

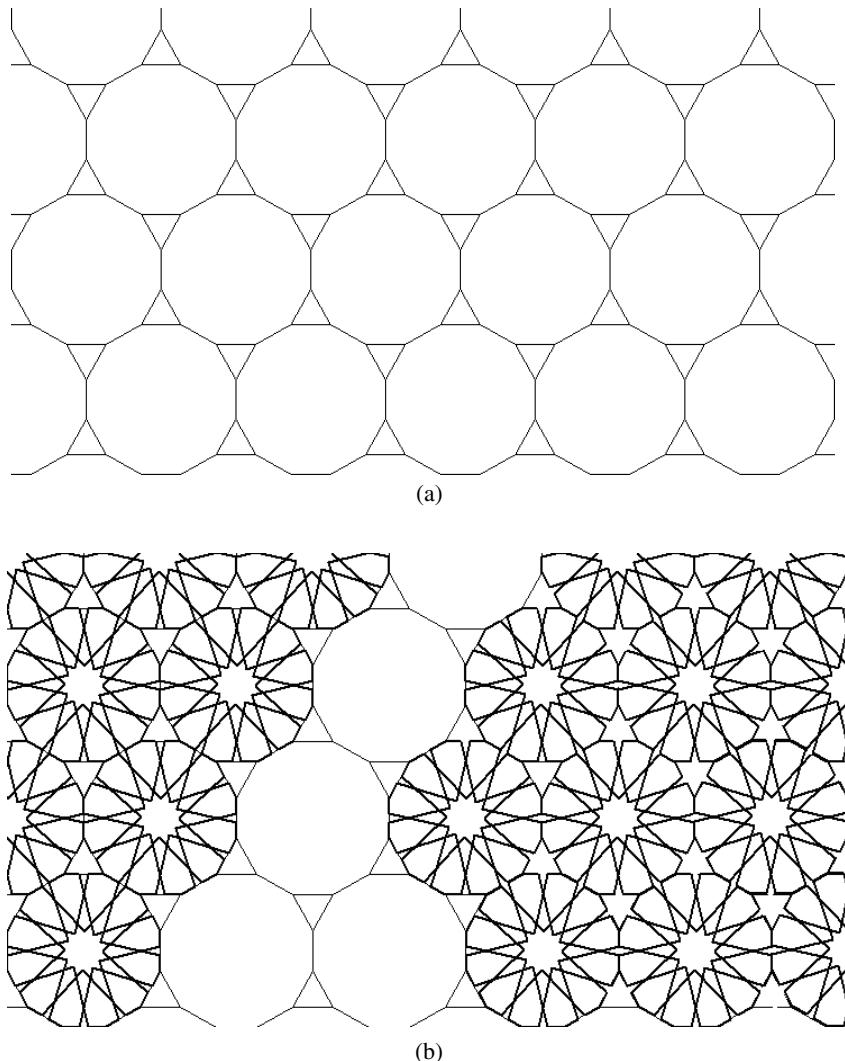


Fig. 13 (a) a regular dodecagon tiling sketch (b) a Developed texture by combining pattern 10-h and regular dodecagon tiling method.

In fig. 14, the regular dodecagon tiling is filled by pattern 10-g. Then regular triangles spaces are filled by a regular hexagon so in this way no irregular free space will remain.

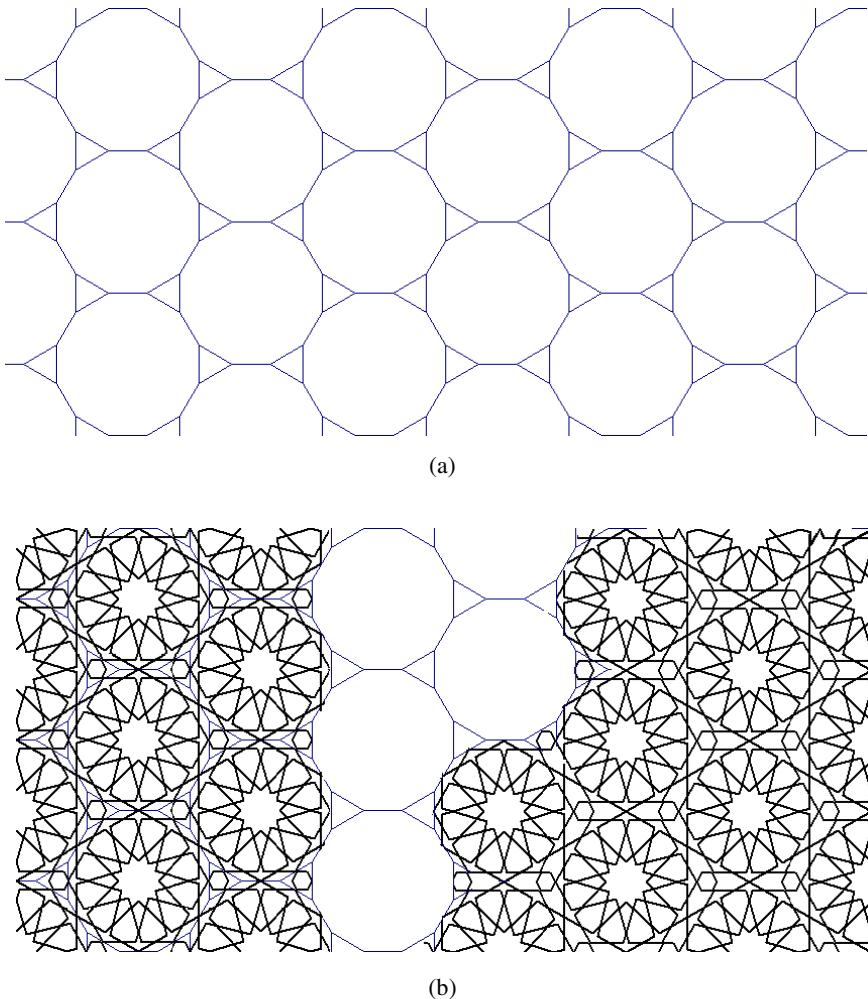


Fig. 14 (a) Regular dodecagon tiling layout (b) Final pattern combining 10-g and dodecagon tiling

Fig. 15-a shows a tiling pattern via a regular octagon. We use this pattern to produce the texture from fig. 17-d. In this layout, a regular tetragon is included within regular octagons. By expanding pattern 10-i the free space of tetragons will be filled by some regular octagons.

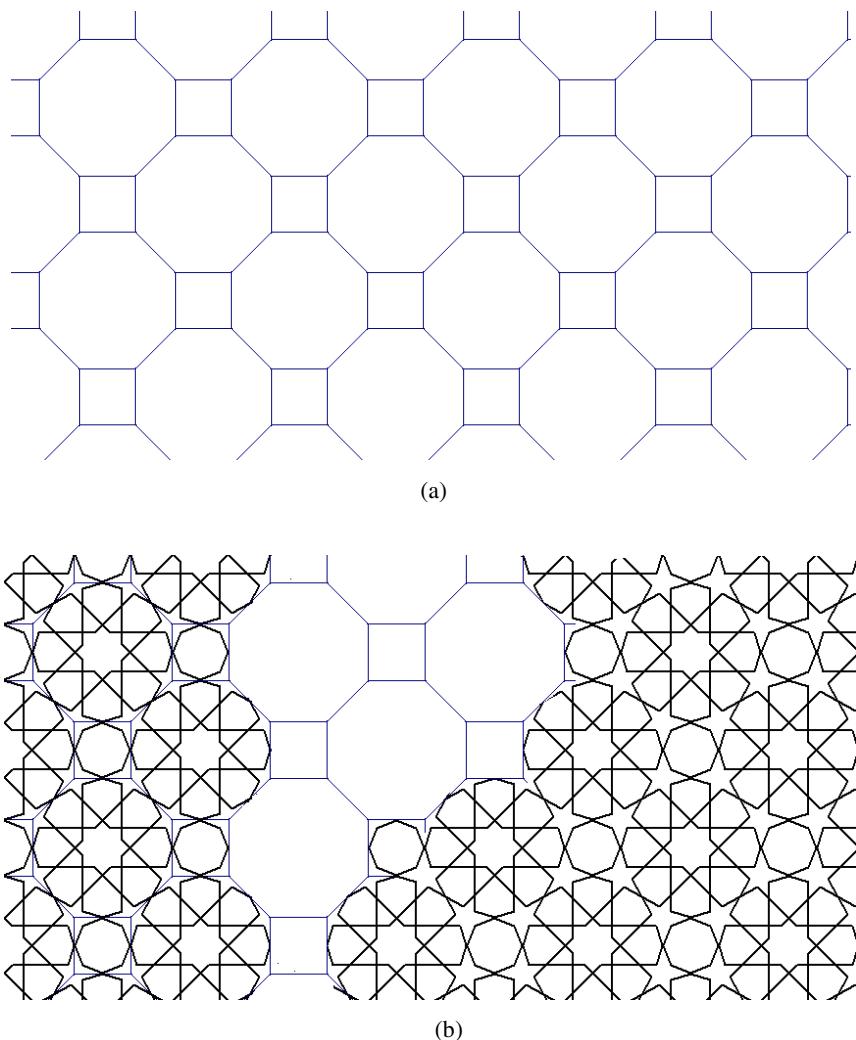


Fig. 15 (a) Regular octagon tiling pattern (b) Final texture mixing 10-i and regular octagon tiling pattern

In fig. 16-a a complex tiling that is made from regular dodecagon and tetragon can be seen. In this complex picture some triangles can be seen as well.

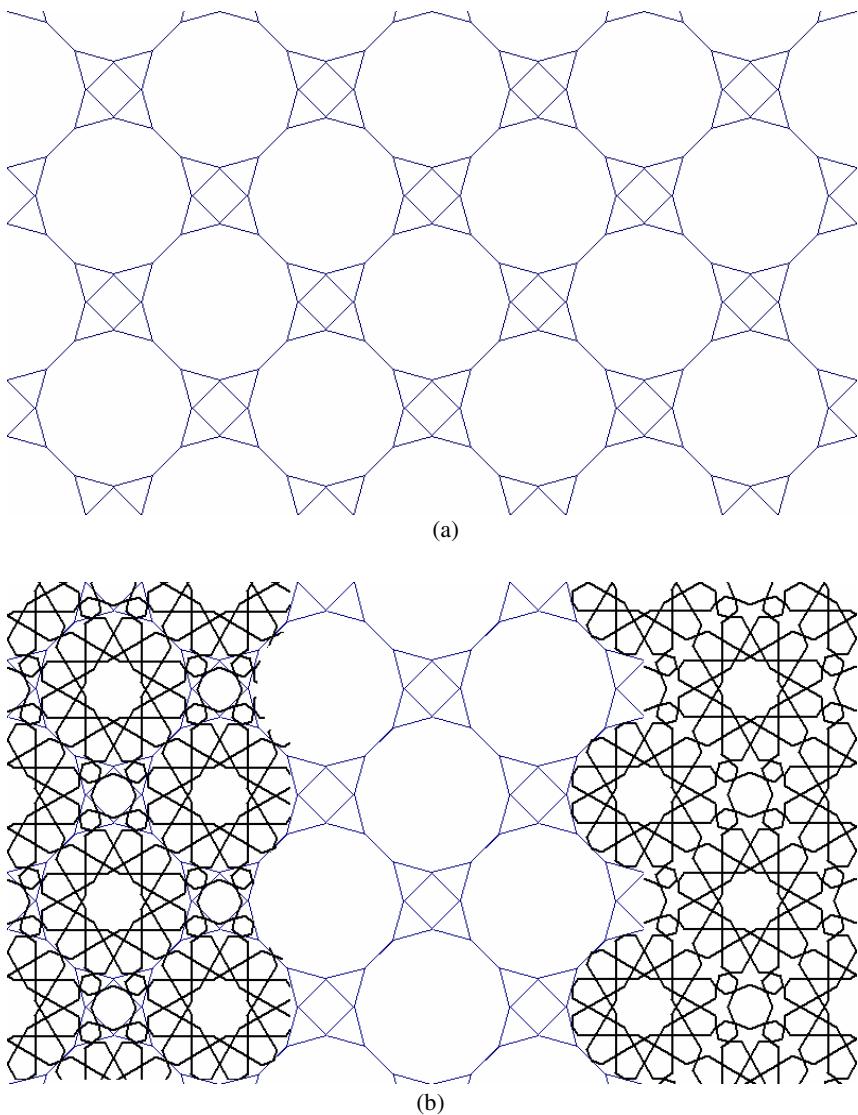


Fig. 16 (a) Hybrid tiling using regular dodecagons and tetragons (b) Final pattern via combining 10-d and regular dodecagons and tetragons

Fig. 16-a represents a hybrid tiling pattern, made of dodecagons and tetragons. Combining these two regular polygons plus some extra triangles has made a nice complex pattern.

In fig. 16-b by combining and filling “dodecagons with 10-d” and “tetragons with octagons” and then “triangle with hexagon” we can reach Fig. 17-a. These are some new textures made of with more than a pattern.

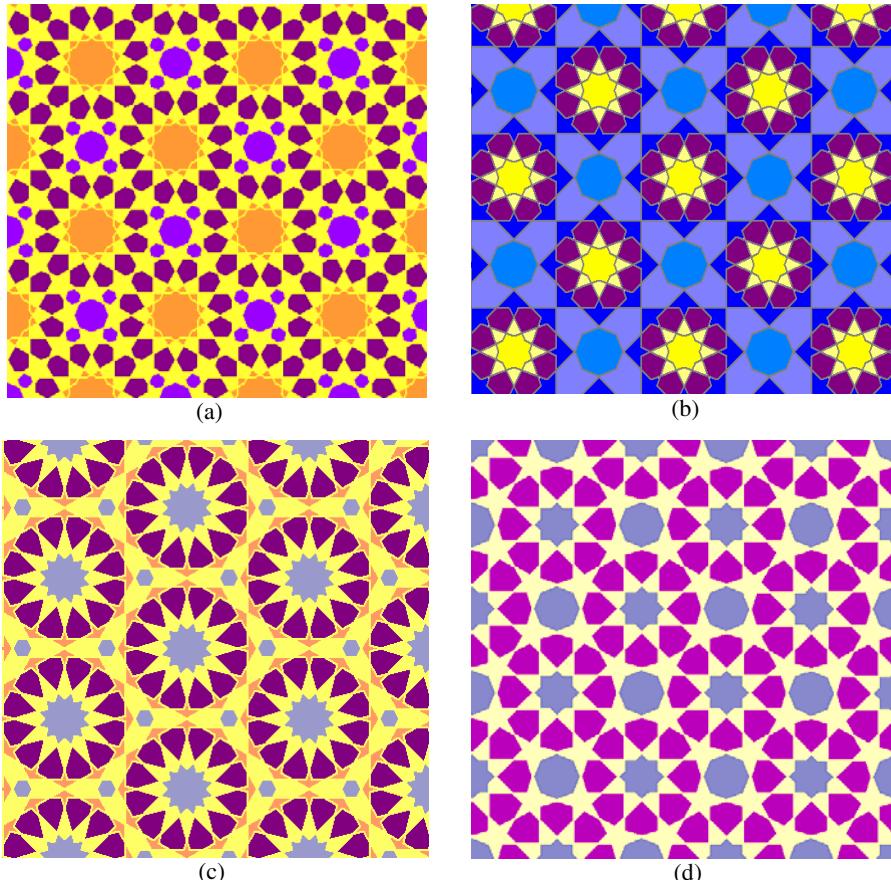


Fig. 17 (a) Hybrid texture via 10-d, regular hexagons and octagons (b) Texture made up of transformed 10-a and regular octagons (c) texture from transformed 10-g and regular hexagons (d) texture form transformed 10-I and regular octagons

3.3 Stars

Other patterns that are abundantly observed in Islamic Geometric patterns are the existing patterns in fig. 18 which are classified in the Islamic star set. The existing pattern in fig. 18-a is made of the repetition of a smaller part shown in fig. 19.

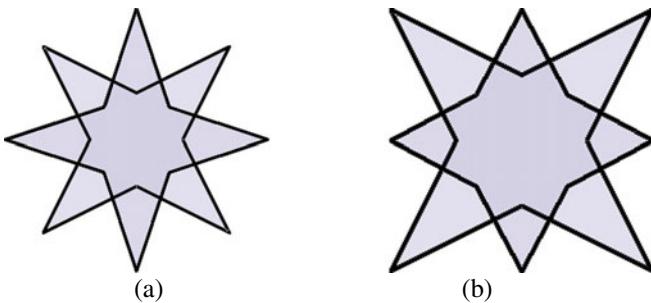


Fig. 18 Octagonal Stars

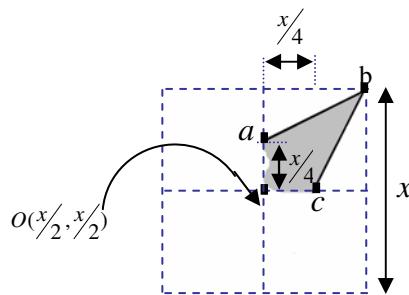


Fig. 19 Initial part of fig. 5

As we assumed (x_0, y_0) as the centre of coordinate system in fig. 3, now we produce fig. 19 based on parameter ‘ x ’. Below is the relevant coordinate axes for points a , b and c :

$$a: (x_0, (y_0 - (x/4)))$$

$$b: ((x_0 + (x/2)), (y_0 - (x/2)))$$

$$c: ((x_0 + (x/4)), y_0)$$

Joining a to b and b to c , fig. 19 will be produced. Fig. 18-a is the rotated form of fig. 19 in a 45° angle for all points of a , b and c .

In fig. 18-b points a and c have the 45° rotation angle, but point b will be moved on the tetragon sides. It means that points a and c have periodic symmetry and point b have transfer symmetry on the tetragon sides.

In fig. 19, if points a and c join to each other and rotate with a 90° angle around the tetragon center of (x_0, y_0) , then fig. 20 will be obtained. The same way is correct for fig. 4. That is, if the vertexes a and e join each other, then the provided figure will produce divergent sketches.

Fig. 21 shows a transfer vector used in different directions and length. Here, we prepared some diversity of sketches which are the extended version of fig. 18 and 20.

In fig. 21, horizontal transfer vector is $(x, 0)$ and vertical transfer vector is $(0, x)$. Considering all provided sketches so far, we just discussed about sketches which obtained as a result of symmetry and fundamental pattern repetition on the surface.

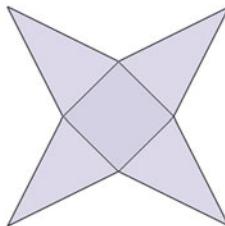
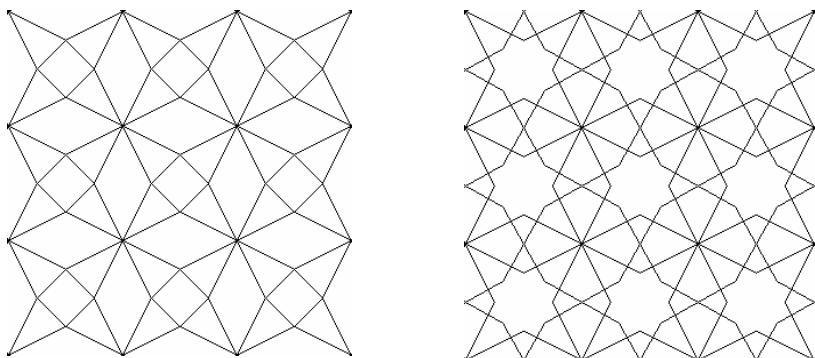


Fig. 20 Reorganized form of Fig. 19 by jointing points a and c , plus a 90° rotation.



(a) Extended version of fig. 7.
 $(x, 0)$: horizontal transferred vector
 $(0, x)$: vertical transferred vector

(b) Extended version of fig. 5.
 $(x, 0)$: horizontal transferred vector
 $(0, x)$: vertical transferred vector

Fig. 21 Transfer vector produced in different directions and length

Fig. 21 indicates some more complex sketches which are provided as the result of combination and expansion of multi-patterns on the surface. Fig. 22-a is an extended and combined version of fig. 6-a and 18-a, with the transfer vector of $(2x, 0)$ for horizontal expansion, and transfer vector of $(0, 2x)$ for vertical expansion.

Fig. 22-b is the result of combination and expansion for fig. 6-a and 20. Similarly, fig. 22-c is the outputted result of combination and expansion which is applied on fig. 6-a and 18-b.

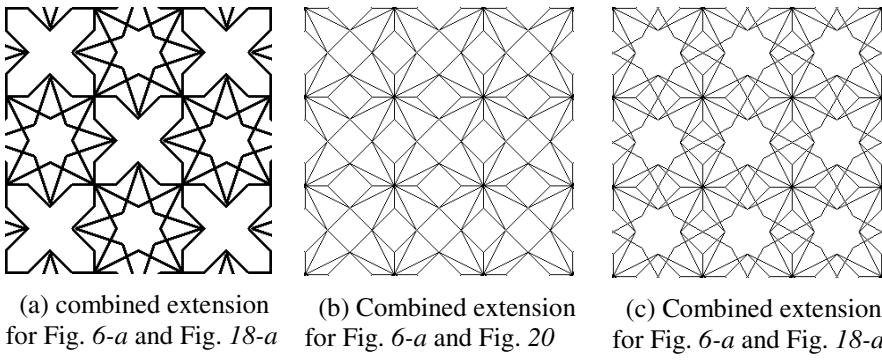


Fig. 22 Complex sketches as the result of “expansion” and “combination” applied on different patterns

3.4 Other Fundamental Pattern

Another fundamental region to create diversity patterns is shown in fig. 23. This figure is composed of an octagon and four diagonal lines. By changing the vertexes of this figure some other fundamental regions so a wide variety of new patterns could be created. Here, one of the lines is highlighted by the vertexes a, b and c . Similar to the method mentioned for fig. 4, and using parameter x , the position and coordinate of a, b and c is reachable.

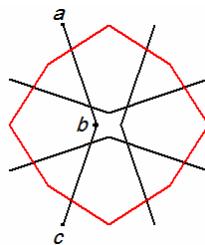


Fig. 23 A new sample fundamental region

After producing the fundamental forms, some samples of which we have mentioned, the sketches should be made with fundamental forms. As mentioned before, transfer vectors are used to produce sketches.

In fig. 24, some other patterns are given that are originated from fundamental region of fig. 23. The differences of these six patterns are rotation angle of vertexes and transform vector.

In fig. 25, another pattern is provided. This sketch is made up of the extended fundamental pattern in fig. 26. Analyzing fig. 26-a, we will find it as the forming part of fig. 26-b. In order to reach fig. 26-b, it just needed to transform the coordinate vector of points b , c , d in fig. 4 toward the given direction in fig. 27.

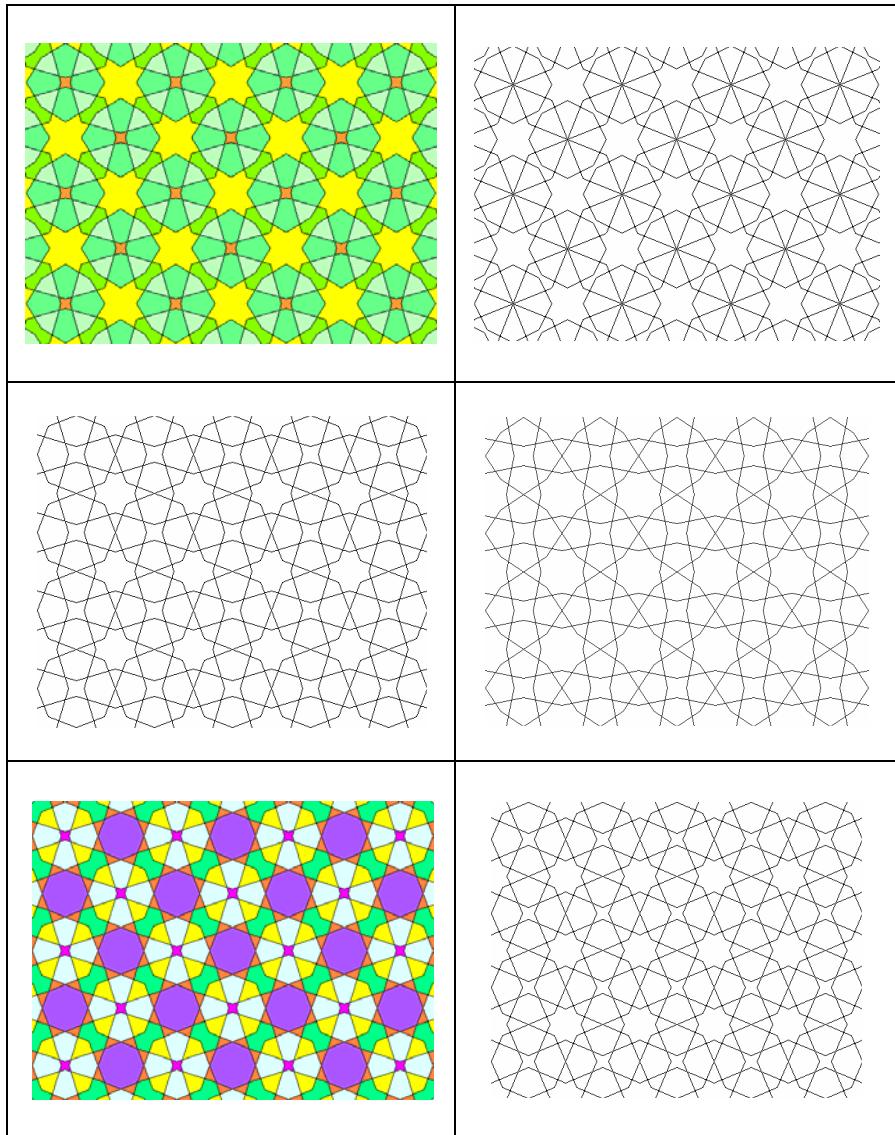


Fig. 24 Some new pattern beginning with fig. 23

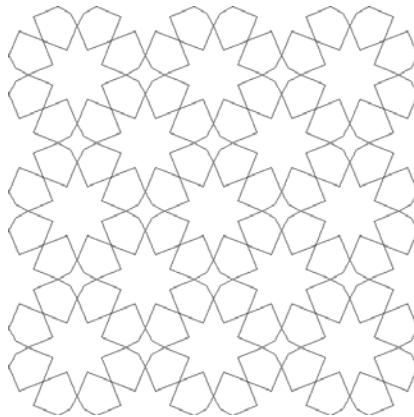
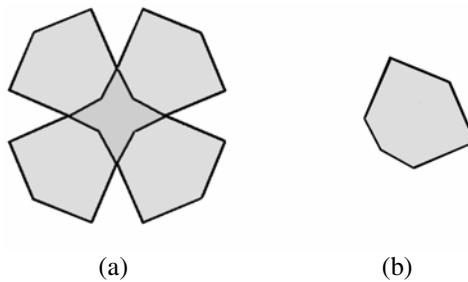


Fig. 25 Another Complex Sketch made up of fig. 26.



(a)

(b)

Fig. 26 Another extension of fig. 4

In fig. 27 it can be seen how fig. 4, 19 and 26-b can be accorded and how these forms can be transformed to each other by replacement of points *b*, *c* and *d*. So, in that way we can easily create both rosettes and stars just by one procedure.

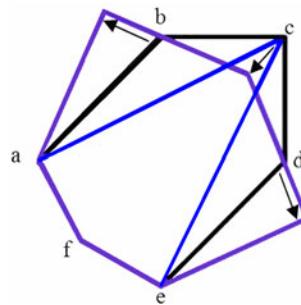


Fig. 27 Accorded form for fig. 4, 19 and 26-b.

Utilizing the given method in fig. 27, a more flexibility will be gained to create more complex patterns. As an illustration, via this method fig. 17-d can be converted to fig. 28.

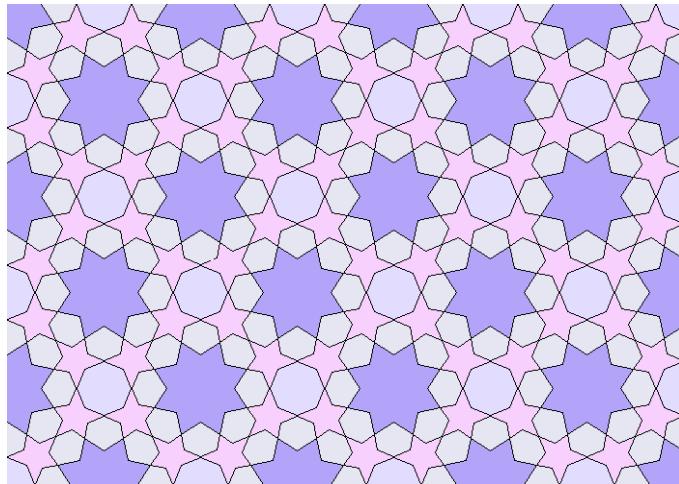


Fig. 28 Converted form of fig. 17-d

3.5 Figure Transformation

Figure transformation is one of the ornamental sketch styles created by William Huff and then generalized by Dongles Hofstadter [17]. In fact, transformation means starting from a figure, then changing the figure and finally transforming it to another figure.

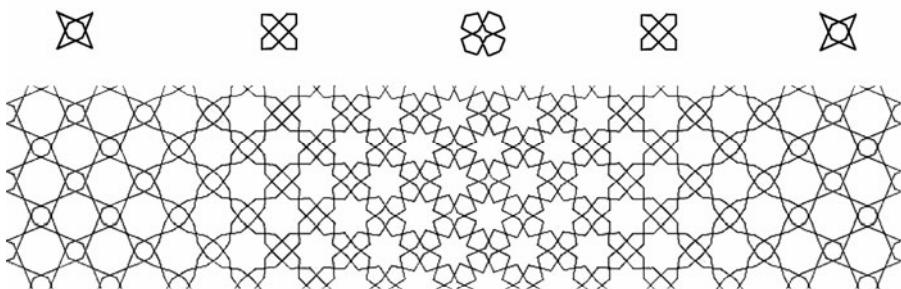


Fig. 29 A Figure Transform Sample for texture development

Fig. 29 illustrates a sample transformation. Applying the proposed method, we can easily change the vertexes of the coordinate in the fundamental figure, so we are capable of creating such a transformation and to make the fundamental area.

Fig. 30 is the GUI of the written software in Delphi.

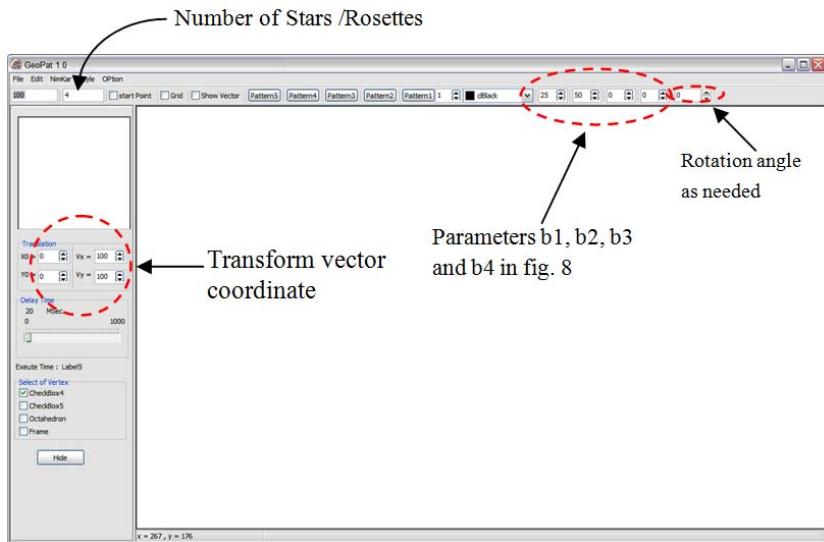


Fig. 30 Graphical User Interface to implement ornamental sketch

By selecting the desired sketch from *Style* menu and then entering the required parameters, the overall form of sketch will be created. Then by defining and entering the transformation coordinate vector, the target pattern will be created.

Fig. 31 shows a sample pattern and its relevant parameters

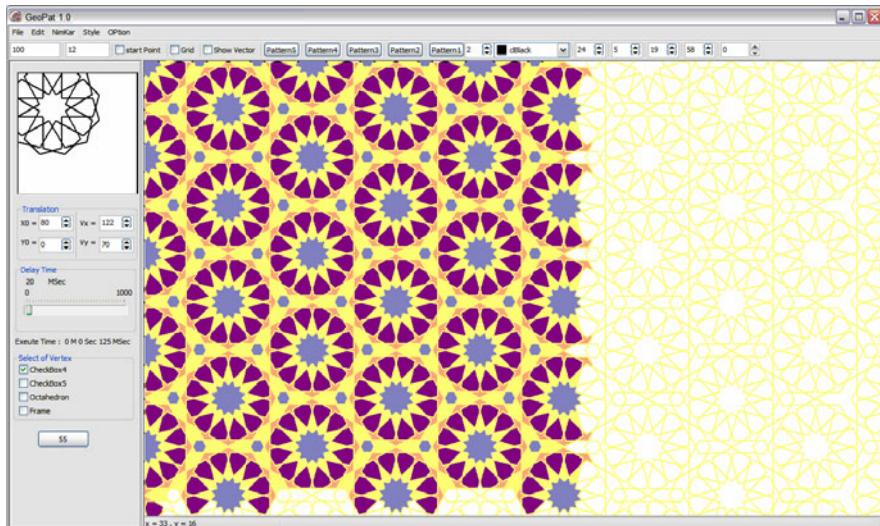


Fig. 31 A sample pattern created by GeoPat version 1.0

4 Conclusions

In this article we offered a new algorithm by which we created several patterns of Islamic Geometric patterns. Introducing a novel point of view for complicated patterns is the prominence of the proposed method. In addition, utilizing basic geometric concepts, the method pursue for individual fundamental sketches, so the computerized implementation would be systematic and easier to understand.

Most of the other's presented methods either focus on generating new patterns or try to map traditional patterns. In contrast, our method not only is able to create and develop sophisticated patterns, but also can expand both traditional and contemporary shapes and patterns.

The proposed method is capable of producing diversity geometric patterns. Having a new view to complicated patterns and sketches is the strength and the highlighted point of this method. The method uses simple individual figures and approaches to a highly sophisticated pattern through some basic geometric concepts.

Most of currently available methods are trying either to create new patterns or just to find a retrieval method to create traditional graphs; however, our method is capable to working on both, that is, possibility of creating new patterns as well as traditional patterns.

As the future work, we will work on 3D colourful Islamic Geometric patterns and try to provide a hybrid algorithm based on the top mentioned method, other researcher's endeavours and image processing techniques. Our objective is to cope with disadvantages and defect of ordinary algorithms in the field. One other thing we are going to improve is filling the existing gaps in the sketches. The authors are currently working on the software to aim above.

References

- [1] Kaplan, C.S.: Computer generated Islamic Star Patterns. In: Sarhangi, R. (ed.) Bridges 2000 Proceedings (2000)
- [2] Ahmad, M., Aljamali, Banissi, E.: Grid Method Classification of Islamic Geometric Patterns. In: Proceeding of WSCG 2003, February 3-7. UNION Agency – Science Pres, Plzen (2003)
- [3] Bonner, J.: Geodazzlers-Eighteen Ornamental Polyhedra for Easy Construction. Design Science Toys Publishing, ASIN: B0010OMLLQ (1997)
- [4] Kaplan, C.S., Salesin, D.H.: Islamic Star Patterns in Absolute Geometry. ACM Transactions on Graphics (TOG) archive 23(2), 97–119 (2004)
- [5] Lee, A.J.: Islamic Star Patterns. In: Grabar, O. (ed.) Muqarnas IV: An Annual on Islamic Art and Architecture, E.J. Brill, Leiden (1987)
- [6] Ostromoukhov, V.: Mathematical Tools for Computer-generated Ornamental Patterns. In: Hersch, R.D., André, J., Brown, H. (eds.) RIDT 1998 and EPub 1998. LNCS, vol. 1375, pp. 193–223. Springer, Heidelberg (1998)
- [7] Bonner, J.: Three traditions of self-similarity in fourteenth and fifteenth century Islamic geometric ornament. In: Proceedings ISAMA/Bridges, Granada, pp. 1–12 (2003)

- [8] Bonner, J.: Islamic Geometric Patterns: Their Historical Development and Traditional Methods of Derivation. Unpublished (2000)
- [9] Hankin, E.H.: The Drawing of Geometric Patterns in Saracenic Art. Memoirs of the Archaeological Society of India 15, LCCN: 96909213 (1925)
- [10] Salman, A.S., Abas, S.J.: Symmetries of Islamic Geometrical Patterns. World Scientific Pub. Co., Singapore (1995) ISBN: 9810217048
- [11] Dewdney, A.K.: The Tinkertoy Computer and Other Machinations, pp. 222–230. W. H. Freeman Publishing, New York (1993) ISBN: 071672491X
- [12] Abas, S.J., Salman, A.S.: Geometric and group-theoretic methods for computer graphics studies of Islamic symmetric patterns. Computer Graphics Forum 11(1), 43–53 (1992)
- [13] Grünbaum, B., Shephard, G.C.: Interlace patterns in Islamic and Moorish art. In: The visual mind: art and mathematics, pp. 147–155. MIT Press, Cambridge (1993)
- [14] Dispot, F.: Arabeske Studio, a Java Tool (2002),
<http://www.wozzeck.net/arabeske/index.html>
- [15] Valiente, J.M., Albert, F., Gomis, J.M.: A Computational Model for Pattern and Tile Designs Classification Using Plane Symmetry Groups. In: Sanfeliu, A., Cortés, M.L. (eds.) CIARP 2005. LNCS, vol. 3773, pp. 849–860. Springer, Heidelberg (2005)
- [16] Lalvani, H.: Coding and generating complex periodic patterns. Journal: The Visual computer 5(4), 180–202 (1989) ISSN: 0178-2789
- [17] Hofstadter, D.: Metamagical Themes: Questing for the Essence of Mind and Pattern. Basic Books (1986) ISBN: 0-465-04566-9
- [18] Lu, P.J., Steinhardt, P.J.: Decagonal and Quasi-Crystalline Tilings in Medieval Islamic Architecture. Journal of Science 315(5815), 1106–1110 (2007)
- [19] Rasouli, P., Bastanfarid, A., Rezvanian, A., Jalilian, O.: Fast Algorithms for Computer Generated Islamic Patterns of 8-ZOHREH and 8-SILI. In: Huang, Y.-M.R., Xu, C., Cheng, K.-S., Yang, J.-F.K., Swamy, M.N.S., Li, S., Ding, J.-W. (eds.) PCM 2008. LNCS, vol. 5353, pp. 825–829. Springer, Heidelberg (2008)
- [20] Cromwell, P.R.: The Search for Quasi-Periodicity in Islamic 5-fold Ornament. The Mathematical Intelligencer 31(1) (January 2009) ISSN: 0343-6993
- [21] Grunbaum, B., Shephard, G.C.: Tilings and Patterns. W.H. Freeman, New York (1987)

Personalized View Selection of 3D Molecular Proteins

Nikolaos Doulamis, Evangelos Chronis, George Miaoulis, and Dimitri Plemenos

Abstract. A non-linear classifier is adopted in this chapter to represent the best view for 3D molecule of a protein onto the 2D screen plane. The classifier receives as inputs visual as well as semantic features and actually model the entropy needed to display with high performance the protein. The visual descriptors have been extracted in our case using the OpenCV toolkit of the Intel Corporation, while the semantic information includes additional knowledge for the protein. Finally, an XML –based middleware is used to embed complex computer vision algorithms into the contemporary protein viewers which allow only limited transformations on the protein data structure. Experimental results on real-life protein molecules are presented to demonstrate the outperformance of the proposed algorithm.

Keywords: Viewpoint entropy, 3D object visualization, image processing/computer vision, artificial intelligence.

1 Introduction

One of the most critical aspects in molecular science is the visualization of the protein molecules so that their complex structures in the three dimensional (3D)

Nikolaos Doulamis · Evangelos Chronis · George Miaoulis
Technological Education Institute of Athens,
Department of Informatics
Ag.Spyridonos St., 122 10 Egaleo, Greece
Tel.: (+30) 2 10 53 85 312; Fax: (+30) 2 10 59 10 975
Email: ndoulam@cs.ntua.gr, idrag@teiath.gr

Dimitri Plemenos
Laboratoire XLIM, Faculté des Sciences,
Université de Limoges
83, rue d'Isle, 87060 Limoges cedex, France
Tel.: (+33) 5 55 43 69 74; Fax: (+33) 5 55 43 69 77
Email: plemenos@unilim.fr

chemical space will be clearly understandable. Nucleic acid and protein sequences contain rich information of great interest to the molecular biologists since the genome forms the blue-print of an organism [1]. As the molecular sequence data continue to grow exponentially, due to the decoding of humans' genome, the significance in understanding the protein structures continually increases affecting drugs' design, identification on the humans' diseases and finding of better medical treatment strategies.

Proteins are knots of complicated networks which, apart from interactions with small diffusing molecules, are mediated by inter-protein surface of varying stability. One of the most important aspects in understanding the protein's structure is the selection of the "best" molecular view since this plays a vital role in understanding proteins' properties. For this reason, several software tools have been developed recently to improve visualization strategies on protein molecules so as to get a quick understanding their structures. Examples include the Swiss PDB viewer [2][3], the visual molecular dynamics [4][5] and the MDL Crime which is a plug-in used by web browsers to display the three dimensional structures of molecules [5].

Nonetheless, the main drawback of all these software tools is the fact that they support a limited programmable interface which is often inadequate for fully understanding the protein structure. For instance, only a set of limited geometric transformations can be supported on the protein molecules via a script language interface. Instead, an efficient algorithm for "best" view selection usually requires application of more complicated image processing/computer vision methodologies that cannot be provided by the proposed software interfaces.

To overcome this problem, the work of [9] proposes a *middleware framework*, able to act as a mediator among the functionalities supported by the protein viewers and the capabilities offered by a computer language interface, like C or Java. This framework offers a powerful medium for applying more complicated algorithms and transformations without, however, losing the capabilities and utilities supported by the protein viewers. In particular, the interconnection is performed using an XML simple schema. More specifically, the script programmable interface of the protein viewer exports an XML file to the middleware. The XML tags are appropriately selected to contain all the necessary inputs for the viewer. Then, the inputs are exploited by the computer programming language so that any complicated protein view selection algorithm can be applied. This constitutes the main contribution of the work of [9]; to be able to apply any generic algorithm on the protein viewer without spoiling the capabilities offered by the viewer and without altering its main capabilities. It should be mentioned that in [9] the Swiss PDB viewer [2][3] has been chosen for conducting all the experiments as in our case.

Despite, however, the method used for selecting the "best" view for a protein, this view is often a subjective issue. Different criteria should be applied for different proteins, or even different criteria should be used if arrays of proteins are displays, i.e., sequences of different proteins. Thus, it is important to introduce a dynamic classifier able to update its performance with respect to the protein structure and selects the most suitable view for the molecule based on the percentages of significant bonds display instead of other, often subjective optimization criteria.

This is the main contribution of this book chapter which is described in Sections 4 and 5. Before this, however, we will briefly give an overview of the recent proposed algorithms for protein' best view selection so that the reader can account the advantages and drawbacks for each technique.

1.1 Brief Literature Overview

When projecting a 3D protein onto the 3D image plane significant information is lost. It is also clear that different structural parts of the protein are revealed on different projections while some other parts are hidden. In addition, optical parallax phenomena may be encountered to further deteriorate the 2D display of the protein preventing the biologist from being analyzing its protein structure. Occasionally, different criteria have been proposed in the literature for selecting the best protein view. In particular, some works exploits the geometric structure of the protein. However, such approaches provide satisfactory results only for simplified objects' structures which are not the case of protein molecules [10]. Towards this direction, minimization of the number of degenerate faces of the three-dimensional object when it is projected onto the two-dimensional scene has been proposed in case of an orthogonal projection [11], [12]. Nevertheless, this algorithm fails when comparing scenes with equal numbers of degenerate faces. Modification of the work of [11] has been presented in [13] [14] to deal with a perspective projection of the object. However, as the authors of [13] [14] explicitly admit, they are unable to define a good weighting scheme for different factors, which actually constitutes the main drawback of this algorithm.

Another criterion is to use an information-theoretic approach via the entropy measure [15][16]. Like the Shannon's information theory, in which the entropy measures an uncertainty degree, the viewpoint entropy defines the "complexity" of the 3D dimensional object onto a 2D screen. Thus, the viewpoint entropy can be used as a powerful tool able to determine the structural complication of a scene when projecting onto the 2D screen plane and thus it can be indirectly used for selecting the most suitable 3D transformation which should be applied so as to get the "best" 2D view of a complicated object, like a protein. The main disadvantages of these approaches is the fact that they adopt complicated software tools, like the open Computer Vision (OpenCV) toolkits of the Intel corporation [17]. In addition, this method does not take into account different optimization criteria for each protein and it does not exploit image processing capabilities of the protein structure to improve the analysis. An improved approach has been proposed in [18][19]. These works exploit both topological and semantic information for choosing the best protein view. Thus, subjective criteria and specific structural information are incorporated to get a more semantic presentation performance for the 3D objects. In addition, in [9], an XML-based middleware platform is introduced to handle the limitation of the current protein displays and thus providing an interoperable and universal accessible framework through which any type of complicated algorithm can be supported by the current viewers though their limited programming capabilities.

1.2 *The Proposed Contribution*

In this book chapter, we describe a new method for selecting the best protein view taking into account the image characteristics of the protein as are projected onto the 2D screen plane. The method provides different results per protein structure yielding a differentiated presentation on how the protein should be displayed. More specifically, an information-theoretic approach is adopted to quantify the protein match onto the 2D screen plane based on the entropy measure. Instead, however, of just selecting as the most appropriate view the one that maximizes the entropy (usually suitable for displaying single protein structure) or the one that minimizes the entropy (usually suitable for displaying an array of proteins), in this chapter, we select the one based on a non-linear relationship of the image characteristics of the 3D projected protein.

In particular, at each different view, a set of appropriate visual features is extracted to represent protein complexity. These features are then fed to a non-linear classifier which provides as output an indicator of how suitable the protein is displayed on the screen. In this chapter, actually the classifier models the non-linear relation of the image properties on the protein structure. A learning vector quantizer is adopted in our case for the classifier structure. The features selected are global and local image processing descriptors. The first provide a generic representation of the proteins in terms of its suitability onto the screen. The second characterizes local patterns (e.g., chemical bonds) so that a more meaningful and semantic information can be extracted and the display is accomplished with respect to semantic meanings instead of simple optimization criteria. This way, we can result into different angles views for different protein structures. Instead our previous work of [9] adopts only two different optimization scenarios for the best view selection. The first refers to the maximization of the entropy measure, which, as we have stated above, is only suitable for a single protein display, while the second for a minimization suitable only suitable for an array of proteins. This means that for the first case, the view with the most complexity is depicted, while for the second case, the view with the minimal complexity so that the arrangement of proteins is better displayed.

The software middleware of [9] is also used to provide an interoperable communication among the viewer components and the programming language interface. Thus, the main contribution of this chapter is that we have built a new best view selection algorithm using a non-linear classification scheme and this algorithm has been embedded on the software middleware of [9]. The new algorithm exploits visual properties on how the protein will be finally displayed onto the 2D screen.

This chapter is organized as follows. Section 2 presents the entropy-based protein view selection algorithm. Section 3 discusses the visual and the semantic features required to be extracted so that a more reliable representation of the protein structure is accomplished. The new proposed best view selection algorithm that combines computer vision and artificial intelligence is shown in Section 4. Experimental results are depicted in Section 5. Finally Section 7 draws the conclusions for this chapter.

2 Viewpoint Entropy in Object Visualization

Since only a 2D displaying method of a 3D object is available, we need to provide appropriate algorithms able to visualize 3D structures so that their spatial relations can be as much as possible visible onto the 2D space. This poses some difficulties in recognizing or understanding the objects since the projected 2D images may present different topology to their 3D counterparts, or might not show enough information of the scene or object. On the other hand, there are some projections, for certain 3D structures, that result in the so-called regular projections of 3D objects. This simply means that the projected 2D faces tolerate the minimal distortion compared to the complete 3D structure. Depending on the application in mind, there are a number of different criteria that can be used to get a good protein view. Moreover, the goodness or quality of a view is often very difficult to be automatically compute because the parameters that determine if a camera position is in good position or not are often difficult to characterize and sometimes such characterization is subjective, that is, the same camera position may be the best one for one user and not for another one. For this reason, the non-linear classifier is adopted in this chapter to model this subjectivity in delivering a 2D protein structure.

Different proteins views serve for different purposes. For instance, a view that shows all the atoms will probably give little information on how the molecule is distributed in space. The opposite is valid when the view shows a small number of atoms [15]. Hence, among the two opposite edges, there are several other paths that we can be followed. Thus how we need to decide the most appropriate selection is generally is tough issue. Generally speaking, protein representation has the following features (a) Atoms (vertices) and bonds (edges) which have a volume, usually displayed as spheres and cylinders. (b) Information which is encoded in the colours and sizes of the atoms' spheres. (C) Orthogonal projection is adopted.

To handle the subjective representation of a molecule onto the 2D image plane, we exploit visual information in this chapter. This means that according to a non-linear model and the values of the visual descriptors extracted for a projected protein onto a 2D screen, we can estimate how suitable would be such projection in term of describing the complexity and diversity of the protein.

The entropy-based representation is used in this chapter as a suitable measure for representing how much information is embedded onto the 2D projected structured compared to the 3D one. In particular, according to Shannon's theory [20], the entropy of a discrete random variable X with values in the set $\{a_1, a_2, \dots, a_n\}$ is given by

$$H(X) = - \sum_{i=1}^n p_i \log_2 p_i \quad (1)$$

where $p_i = \Pr(X = a_i)$ (\Pr denotes the probability), $\log_2 p_i$ is the 2 base logarithm of p_i while for continuity reasons the $p_i \log_2 p_i = 0$ when $p_i = 0$.

Similarly, if we assume a set of N faces, we can use as probability distribution the relative area of the projected faces over the sphere of directions centred in the viewpoint p , that is, [21]

$$H(p) = - \sum_{i=0}^N \frac{A_i}{4\pi} \log \frac{A_i}{4\pi} \quad (2)$$

where A_i is the projected area of face i . Since 4π is the solid angle of a sphere, variable $A_i / 4\pi$ indicates the visibility of face i with respect to the point p .

The previous measure assumes that each face is a different element of information. Especially for the case of protein depiction, some fundamental differences can be considered. The first is that molecules are formed by atoms and bonds which are represented by spheres and cylinders respectively and the user needs to see only a part of the sphere or of the cylinder to identify it. This helps with the computation process as the number of different objects to measure (atoms plus bonds) drops to some hundreds or thousands [16]. Furthermore, for a molecular representation, an orthographic projection is adequate and this significantly simplifies the entropy as each pixel covers the same solid angle.

Thus, in this case the entropy of (2) can be written as

$$H(p) = - \sum_{i=0}^N \frac{N_i}{N_{Total}} \log \frac{N_i}{N_{Total}} \quad (3)$$

where N_i is the number of projected pixels for the i^{th} object and N_{Total} the total number of image pixels.

Using equation (3), we can select an algorithm for selecting the most appropriate 2D views. The same criterion has been adopted in the [9]. However, as we have mentioned above, two simple optimization criteria are taken into account in this case; the entropy maximization and the entropy minimization. These two criteria are generally accepted for a single molecular depiction or for a depiction of an array of molecules.

Single Molecular: For a single molecular, the best view is the one with the highest amount of information. This means that the “best view” shows most of the atoms and bonds. Therefore, it can be used for identifying the elements of the molecule and their relative positions. This means that for a single molecular the best view is given from the *maximum entropy value* [see equation (3)].

Arrangement of molecules: On the other hand, for an arrangement of molecules, we are interesting in directions of regular ordering. This viewpoint is estimated by the minimum value of the entropy.

Both approaches, however, do not take into account specific visual properties of the protein as it is projected onto the 2D screen. The visual characteristics are very important since on them the inspection of the protein will be accomplished. In addition, semantic information on a protein molecule should be exploited. This constitutes the main contribution of this chapter; differentiated selection of the protein view with respect to its particular structure based on visual and semantic properties

that have been extracted when the protein molecule is depicted onto the 2D screen. For this reason, in the following section, we modify the entropy measure so that it includes more complicated features so that a better protein view is accomplished.

3 Visual Descriptors for Estimating Viewpoint Entropy

Two types of descriptors are used in this chapter to describe the 2D projected protein structure. The first are visual features which depicts the visual complexity of the projected molecule. The second are semantic information which is included in the molecule so as to describe its type, chemical origin and structure. Both features are quantified and are fed in a classifier to improve the molecule view and make the selection to be subject to the proteins' types.

3.1 Visual Descriptors in Object Visualization

Let us denote as I the 2D protein image after its projection onto the view screen. Let us also assume that the image plane consists of $N_1 \times N_2$ pixels and let us assume that the image I is partitioned into K mutually exclusive regions R_i , $i=1,2,\dots,K$. For each region R_i , several attributes are extracted to represent the image visual content. Let \mathbf{d}_i be a *descriptor vector*, which contains all the extracted attributes associated with region R_i . Then, vector \mathbf{d}_i can be written as

$$\mathbf{d}_i = [D_1(R_i) \ D_2(R_i) \cdots D_P(R_i)]^T \quad (4)$$

where $D_j(R_i)$, $j=1,2,\dots,P$ are operators, which return the j th attribute of R_i . The $D_j(R_i)$ can be either a scalar value, (e.g., the operator returns the area of region R_i), or a vector value, (e.g., the operator returns the three color components of R_i).

A single image pixel p_i , $i=1,2,\dots,N_1 \cdot N_2$ (Case 1) or an image object (segment) O_i , $i=1,2,\dots,K^o$ (Case 2), obtained by applying, for example, a segmentation algorithm, can be considered as image region R_i . This means that in,

$$\text{Case 1} \quad R_i \equiv p_i, i=1,2,\dots,K=N_1 \cdot N_2 \quad (5a)$$

and in

$$\text{Case 2} \quad R_i \equiv O_i, i=1,2,\dots,K=K^o \quad (5b)$$

In the first case, the extracted attributes are referred as *global-based descriptors*, since no local processing of the image is accomplished. Instead, in the second case, the descriptors are called *object-based descriptors*.

For global-based descriptors, the spatial distribution of color within an image is used similarly to the Moving Picture Expert Group (MPEG)-7 color layout descriptors. The descriptor uses an array of representative colors for the image, expressed in the YCbCr color space, as the starting point for the descriptor definition. The size of the array is fixed to 8x8 elements to ensure scale invariance of the descriptor [22].

In Case 2, the image is first partitioned into several segments (objects). In this chapter, only color segments are considered. A multiresolution implementation of the Recursive Shortest Spanning Tree (RSST) algorithm, called M-RSST [23] has been adopted to perform the segmentation task. More specifically, the adopted scheme recursively applies the RSST [24] to images of increasing resolution. The advantage of the M-RSST compared to the RSST algorithm is that it yields much faster execution time, especially for images of large size, while keeping almost the same performance.

For the second case, the dominant color can be also used. This color descriptor aims to describe global as well as spatial color distribution in images. Colors in a given area are clustered into a small number of representative colors. The descriptor consists of the representative colors, their presences in a region, spatial coherency of the color (optional), and the color variance (optional). Colors are clustered into a small region of representative colors. The number of dominant colors can vary from image to image and a maximum of eight colors can be used to represent the dominant colors.

Both descriptors are fed fused together to create a complete feature vector, say, \mathbf{d}_i . Other descriptors' types can be adopted by the proposed system to improve the visual description of a protein.

3.2 Semantic Description

Apart from the visual descriptors, in this chapter we also use semantic information to improve the analysis. This is very important since different types of proteins or even arrangements of proteins should be differently handled and other views should be considered as the most suitable.

The semantic information is filled in by the expert users (e.g., biologists). It can include the name of the protein, its code, some biological characteristics, arrays of proteins and how they are connected together as well as any other biologically important information that can yield to a differentiated analysis of the results as far as the best 2D projected view is concerned.

The semantic information is quantified in our case so as to be easily handled by the non-linear classifier. This means that we map the protein types, names, and grouping information into certain numerical values and based on these values we form the feature vector.

All the visual (vectors \mathbf{d}_i) and semantic data are fused (concatenate) together so that they construct a full descriptor vector that will represent the specific protein. We denote as \mathbf{g} the fused descriptor that contains all the visual and the semantic information.

4. Personalized Selection of Object Views

A non-linear classifier is adopted in this chapter for indicating the most suitable view for a projected protein onto the screen. The classifier takes as inputs the image features as described in the previous section and the entropy measure which indicates a quality metric for the projected protein. To train the non-linear classifier, two phases are adopted. The first refers to a supervised learning in which a set of proper projected samples have been constructed and the appropriateness for selecting the most suitable view is assigned by expert users (biologists). This case, we relate the samples (visual features and the entropy measure) with a precise view of the protein.

However, such a static training is not adequate since feedback from the user cannot be taken into account. This means that incorrect matching among the descriptors / entropy and goodness of the view cannot be traced in near future as new samples arrive and new data are coming in the system. For this reason, we enrich the supervised learning process with unsupervised tools yielding a semi-supervised learning activity. In the following subsection, we describe the first initial phase used to train the non-linear classifier, while in section 4.2 we discuss how the supervised learning can be enriched with unsupervised capabilities.

4.1 Modelling of View Preferences

Let us assume that an expert user has submitted to the system several characteristics samples of proteins, say M , per each sample, some 3D transformations are applied and the 2D projected protein image is created. This represents a random view onto the 2D screen. Supposing that each protein is undergone L different 3D transformations we construct an $M*L$ training set that contains characteristics samples of L different views for M proteins along with an assessment of their performance on the 3D space. In particular, let us recall that $\mathbf{g}_{i,j}$ is the total descriptor for the i th protein at the j th 3D transformation from its original form when it is orthographically projected onto the 3D feature space. Then, if we evaluate the fitness of the 2D projected protein as $d_{i,j}$ then we can construct a training set that contains the samples

$$S = \{(\mathbf{g}_{1,1}, d_{1,1}), (\mathbf{g}_{1,2}, d_{1,2}), \dots, (\mathbf{g}_{1,L}, d_{1,L}), (\mathbf{g}_{2,1}, d_{2,1}), \dots, (\mathbf{g}_{M,L}, d_{M,L})\} \quad (6)$$

with appropriate samples and the corresponding evaluated outputs.

Let us also denote as $o_{\mathbf{w}}$ the output of a non-linear classifier when \mathbf{w} weights are used as its parameters. Then, the goal of any training algorithm is the minimize the mean squared error among all samples $M*L$ of the training set so that the classifier output to be as much as possible closer to the desired output $d_{i,j}$. That is,

$$\hat{\mathbf{w}} : \arg \min_{\mathbf{w}} J(\mathbf{w}) = \arg \min_{\mathbf{w}} \sum_{i=1}^M \sum_{j=1}^L (d_{i,j} - o_{\mathbf{w}}(\mathbf{g}_{i,j}))^2 \quad (7)$$

In equation (7), $\hat{\mathbf{w}}$ is the optimal weights of the classifier while $o_{\mathbf{w}}(\mathbf{g}_{i,j})$ denotes the classifier output when features of the i th protein on the j th representation are fed as inputs.

Based on the classifier selected, different learning strategies can be adopted to estimate weights that can approximate the optimal ones $\hat{\mathbf{w}}$. Despite, however, the classifier adopted and the method used to train the algorithm, the aforementioned training strategy follows supervised learning, in the sense that the samples should represent ideally the complete complexity of the protein space. In such complicated applications, nevertheless, selection of representative samples is actually a very arduous task, meaning that the estimate weights of the classifier cannot refer to the whole complexity of the protein space. This, in the sequel, may dramatically increase the so-called generalization effect. That is, although the error performance of the classifier on data inside the training set can be minimal, its performance on data outside the training set can be high.

In supervised learning, several heuristic techniques have been proposed to decrease generalization error and make the classifier performance robust to data outside the training set. One solution towards this direction is the proper selection of a relatively large training set. Such approach, however, despite the practical difficulties in constructing the final set, present the additional drawback of overfitting since we generally force the classifier to learn well the training data. Another approach is the division of the training set into a small validation set which can be used of assessment set of the classifier performance to data outside the training set. Again, the complexity of the protein space is so huge that a relatively small validation set can not guarantee proper performance to newly arrived protein data. For this reason, we adopt in this chapter, an adaptation process, through the use of an on-line learning strategy.

4.2 Best Protein View Selection

In this chapter, a non-linear classifier is adopted to map the visual and semantic features to the “best” protein view. In other words, the classifier learns a non-linear relation that models the protein entropy so that the best view is achieved. The reason that we adopt a non-linear relationship is due to the fact that the visual and semantic information of the data cannot represent in an efficient way the complexity of the scene in terms of molecular structure.

Several non-linear classifiers can be used for this case. We select the Learning Vector Quantization (LVQ) classifier due to its highly non-linear structure and the simultaneous simplicity in understanding the problem complexity [25].

LVQ is one example of competitive learning. The goal here is to have the network “discover” structure in the data by finding how the data is clustered. In vector quantization, we assume codebooks defined as the representatives of Q prototype vectors. We assign an input vector to one of the Q available prototypes, say the i th, if this the index is the closest to the respective prototype (closest in the sense of a metric distance or a similarity measure). This has the effect of dividing up the input space into a Voronoi tessellation. In particular, in LVQ approach, we randomly

pick one sample out of the available ones. Then, we find the “winning” node for this sample, i.e., the cluster whose distance from this sample is smaller than any other distance with other clusters, meaning that the selected sample and the respective node winner node if the optimal choice we can made at this stage of the algorithm. Then, we update the weights for the winning cluster to take into account this new information.

In other words, if we denote as \mathbf{w}_i the weight vector for the i th prototype and we recall that \mathbf{g} is an input vector for a randomly selected training sample, then we can assign \mathbf{g} to the prototype whose the distance is closest compared to all the other prototypes. That is,

$$d(\mathbf{w}_i, \mathbf{g}) < d(\mathbf{w}_k, \mathbf{g}) \quad \text{for } k \notin i \quad (8)$$

In particular, the algorithm starts at the 0th iteration in which it assumes that Q representative codebooks have been randomly derived, denoting as $\mathbf{\mu}_i(t)$ the representative at the t -th iteration. Then, picking up a random sample \mathbf{g} and denoting as $\mathbf{\mu}_k(n)$ the best (closest) representative for this sample at the n th iteration of the algorithm, then, this representative is updated so that

$$\mathbf{\mu}_k(n+1) = \mathbf{\mu}_k(n) + \mu(\mathbf{x} - \mathbf{\mu}_k(n)) \quad (9)$$

while the other representatives remain intact. Parameter η defines the convergence rate for achieving optimality.

4.3 Automatic Estimate of the Number of Representatives

In this section, we show how the number Q of the most representative prototypes is selected. In particular, if the distance of a picked sample is far away from all Q representatives, a new representative is created since this means that the sample cannot be described by any of the existing codebooks. In this case, we set the new representative equals the new sample and we follow what we have described above.

Similarly, if the values of two representatives are too close we remove one of the two since they refer to almost similar content. Between the two, we remove the representative that less describes the variations and information of all samples assigned to it. This way, we automatically update the number of representative per background area.

4.4 On-Line Learning Strategy

We propose a new algorithm for dynamic learning the classifier weights so that new samples are trusted as much as possible with a simultaneous minimal degradation of the already obtained classifier knowledge. This can be achieved through a dynamic relevance feedback strategy. In particular, upon a best view selection for a protein, the expert user (e.g., biologist) can choose some samples of interest

and evaluate the system performance on them. In other words, the expert user can assess if the view proposed by the system for the particular protein type is in accordance with what the system suggests or in an opposite direction. This information can be exploited by the system to improve its performance on data outside the training set. In particular, the assessed samples can be considered as the new data and can be included in the training set so as to re-calculate the classifier weights. In order to avoid a continuous increase of the data size, we impose in this chapter a forgetting factor for “old” samples so that the current knowledge is trusted more than the previous one. This way, we adapt the performance of the classifier to the new data sets increasing the robustness of the system.

For each newly arrived and selected sample, the visual and semantic features described in section 3 are extracted and then included in the system. Then, we find previous data samples of the same type (i.e., protein type and name) and we assign a forgetting factor so as to decrease the effect of previous samples on the learning process.

5 Experimental Results

In this section, we describe the experimental results derived by the use of the proposed architecture. Initially, we describe the protein viewer used along with the middleware framework as in [9] which can enrich the viewer with more complicated capabilities. Then, we discuss the performance of the proposed classifier-based protein view selection which takes into account visual and semantic information.

5.1 Swiss Protein Data Bank Viewer

The Swiss Protein Data Bank Viewer is a software tool used to visualize proteins’ structures. The tool supports a simple programming interface through the usage of a script language. The tool also supports the *Swiss-Model*, which is a fully automated protein structure homology-modeling server. The purpose of this server is to make Protein Modelling accessible to all biochemists and molecular biologists WorldWid [26]. Swiss PDB viewer also provides tools for protein structural alignment.

Using the Swiss PdB Viewer, the proteins can be superimposed in order to deduce their structural topology, which is a critical aspect for the protein functionality. In addition, the Swiss PDB viewer provides to the biologists and chemical engineers information about the active sites of a protein, or any other relevant part such as Amino acid mutations, H-bonds, angles and distances between atoms. This is accomplished through an intuitive graphic interface, which allows for end users to analyze several proteins at the same time (see Fig. 1).

The Swiss PDB viewer is incorporated with advanced image rendering techniques, which allow for efficient protein visualization. In particular, the viewer is incorporated with the POV (Persistence of Vision Raytracer) ray tracing program

in order to generate ray traced high quality images. **Error! Reference source not found.** presents an example of the ray traced images produced in the framework of the Swiss PDB viewer.

The Swiss PDB viewer provides a script language which allows for users to develop their own applications. The scripting language supports variables, conditional branching, loops, arrays and file access. The way of programming the Swiss PDB viewer is similar to the way used in C and Perl languages.

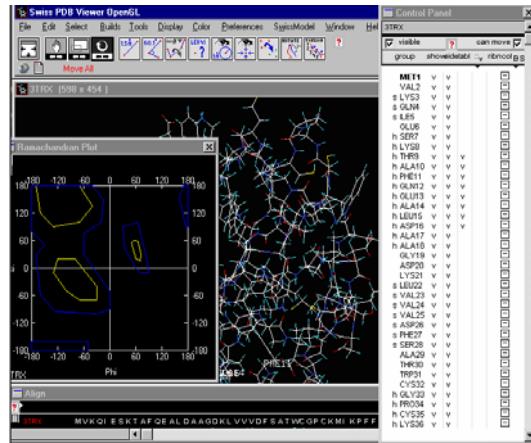


Fig. 1 The main interface of the Swiss PDB viewer

5.2 Classifier-Based Entropy Implementation

We have used the OpenCV toolkit of Intel to develop the classifier-based results [17]. OpenCV allows for a real time implementation of complicated computer vision tools since it exploits hardware properties of the Intel CPUs. In addition, the Document Object Model (DOM) is adopted to parse the XML file according to the W3C recommendations.

Once the XML file is parsing the next step is to activate the algorithm for estimate the entropy of the current viewpoint. The learning vector quantizer is used in this chapter to select the “best” protein view. In particular, visual and semantics features are extracted by the developed software per each projected protein as exported by the viewer. The developed algorithm reads a bmp image file since this type of format is exported by the PDB viewer. The features are fed as inputs to the LVQ classifier which provides as output an indication of the appropriateness of the view of the protein as far as its structure is concerned. Then, the most suitable protein view is imported to the viewer and is displayed. As we have stated

previously, all the communication among the viewer and the programming interface is accomplished using an XML schema to give interoperability to the proposed architecture.

5.3 Simulation Results

Fig. 2 depicts different ways that a protein can be depicted through the PDB viewer. All the results in Fig. 2 illustrate a protein with th code 1 REI. In particular, Fig. 2(a) shows a 2D display of the protein, Fig. 2(b) a 3D protein structure representation while Fig. 2(c) the energetic content of the respective protein. In all cases, different angle views have been selected.

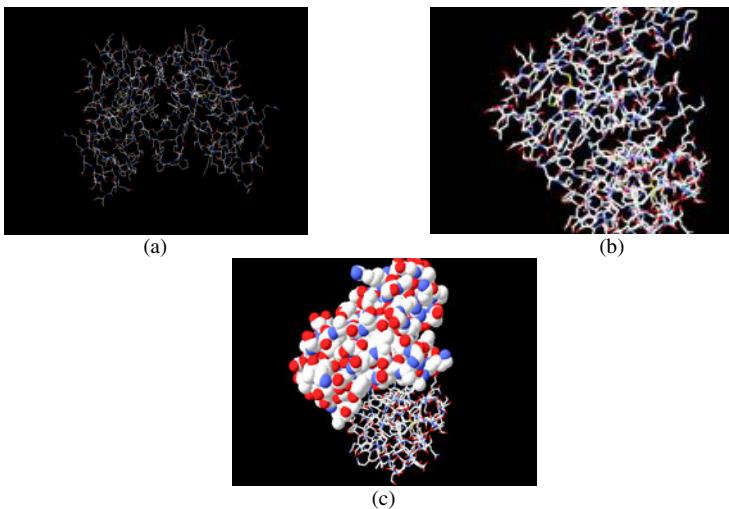


Fig. 2 Different representations for a protein by the PDB viewer

Fig. 3 shows a very simple structure for a protein. It is clear that in such simple structure the best view of the protein would be the one of the maximum entropy as depicted in Fig. 3(c). That is, in the case the simple maximization or minimization of [9] can provide satisfactory results. In case, however, of another more complicated protein, the simple optimization algorithm of [9] does not work. This is, for example, the case of the protein wit code 2KW6 which is displayed in Fig. 4. As we can observe, this is a very complicated protein with many bonds and atoms and the use of simple optimization criteria cannot yield proper results.

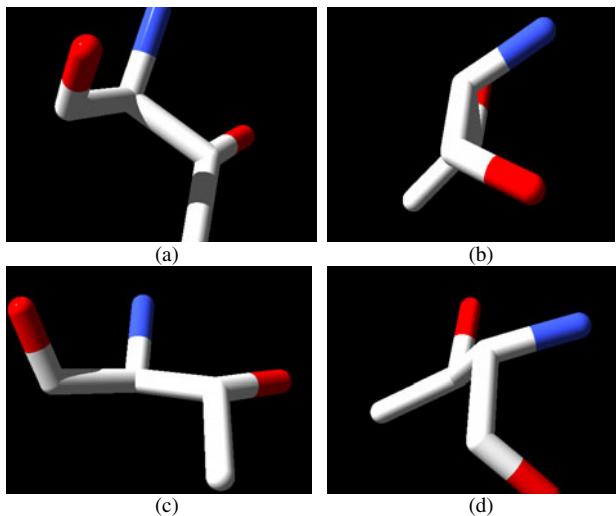


Fig. 3 The respective four different ray traced images

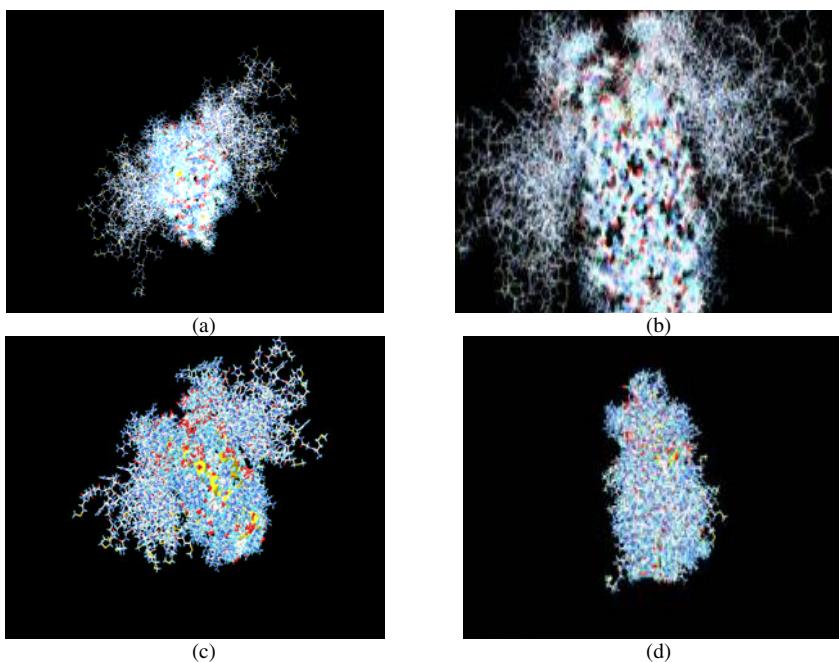


Fig. 4 Different views for a more complicated protein, with code 2KW6

The classifier-based algorithm described above has been adopted in this chapter to select the best view for this protein. Fig. 4 shows the results for the 2KW6 protein. As we can see, the presented structure is much more complex than the one shown in Fig. 3. Fig. 4(a) shows a random view for that particular protein, while Fig. 4(d) the worst angle view. Finally, Fig. 4(b) shows the best view as derived by the application of the maximum entropy (algorithm of [9]) while the proposed best view is depicted in Fig. 4(c). We can see that the developed method better represents the complexity and the structure of the derived protein.

6 Conclusions

Proteins are knots of a complicated network of interactions which, apart from the interaction with small diffusing molecules, are mediated by inter-protein surface contacts of varying stability. Understanding the role of the topology in protein is of key importance for drug design. Therefore, efficient tools for visualizing a protein structure are significant in the area of medicine and molecular science. Selection of the “*best*” molecular view is useful for understanding the chemical data structure and the respective properties.

In this chapter, we propose a new approach for selecting the best display view for a protein structure. The method exploits visual features derived through the application of computer vision algorithms onto the 2D projected protein features and semantic information which is combined with non-linear classifiers. In this chapter, a learning vector quantizer is adopted in our case for the classifier structure. This way, we can result into different angles views for different protein structures. Thus, this chapter extends one of our earliest works as the ones presented in [9], in which only simple optimization criteria for the entropy are adopted. On the other side, it keeps the interoperable middleware architecture of [9] for communicating contemporary protein viewers with usually limited capabilities with complex programming language interfaces, like the C one.

Experimental results on real-life protein molecules showed the outperformance of the proposed method compared to other classical techniques like the one proposed in [9].

References

- [1] Wu, C.H.: Artificial neural networks for molecular sequence analysis. *Computers & Chemistry* 21(4), 237–256 (1997)
- [2] Guex, N., Diemand, A., Peitsch, M.C., Schwede, T.: Swiss-Swiss-PdbViewer, DeepView 4.0, <http://spdbv.vital-it.ch/>
- [3] Guex, N., Peitsch, M.C.: Swiss-PdbViewer: A Fast and Easy-to-use PDB Viewer for Macintosh and PC. *Protein Data Bank Quaterly Newsletter* 77, 7 (1996)
- [4] Visual Molecular Dynamics, VMD,
<http://www.ks.uiuc.edu/Research/vmd/>
- [5] Dalke, A., Schulten, K.: VMD - Visual Molecular Dynamics. *William Humphrey Journal of Molecular Graphics* 14, 33–38 (1996)

- [6] Symyx Chime Pro, <http://www.symyx.com>
- [7] Incorporating Viewpoint Entropy into the S-PDB viewer
- [8] for Selecting Optimal 3D object Views
- [9] Doulamis, N., Chronis, E., Doulamis, A., Miaoulis, G., Plemenos, D.: Incorporating Viewpoint Entropy into the S-PDB viewer for Selecting Optimal 3D object Views. In: International Conference on Computer Graphics and Artificial Intelligence (3ia), Athens, Greece (May 2010)
- [10] Toussaint, G.T.: The complexity of computing nice viewpoints of objects in space. In: Vision Geometry IX, Proc. SPIE International Symposium on Optical Science and Technology, San Diego, California (2000)
- [11] Kamada, T., Kawai, S.: A simple method for computing general position in displaying three-dimensional objects. Computer Vision, Graphics, and Image Processing 41(1), 43–56 (1988)
- [12] Roberts, D.R., Marshall, A.D.: Viewpoint selection for complete surface coverage of three dimensional objects. In: Proc. of the British Machine Vision Conference (1998)
- [13] Barral, P., Dorme, G., Plemenos, D.: Scene understanding techniques using a virtual camera. In: de Sousa, A., Torres, J.C. (eds.) Proc. Eurographics 2000 (2000)
- [14] Dorme, G.: Study and Implementation of 3D Scenes Comprehension Techniques, PhD thesis, Universite de Limoges (2001) (in French)
- [15] Vazquez, P.-P., Feixas, M., Sbert, M., Heidrich, W.: Automatic view selection using viewpoint entropy and its application to image-based modeling. Computer Graphics Forum 22(4), 689–700 (2003)
- [16] Vazquez, P.-P., Feixas, M., Sbert, M., Llobet, A.: Realtime automatic selection of good molecular views. Computers & Graphics 30, 98–110 (2006)
- [17] Intel® Integrated Performance Primitives, Reference Manual, Image and Video Processing, vol. 2,
<http://software.intel.com/sites/products/documentation/hpc/ipp/pdf/ippiman.pdf>
- [18] Sokolov, D., Plemenos, D.: Virtual world exploration by using topological and semantic knowledge. The Visual Computer 24(3), 173–185 (2008)
- [19] Sokolov, D., Plemenos, D.: Automatic exploration of virtual worlds. In: International Conference WSCG 2008, Plzen, Czech Republic, February 4-7 (2008)
- [20] Shannon, E.C.: A mathematical theory of communication. The Bell System Technical Journal 27, 379–423, 623–656 (1948)
- [21] Vazquez, P.P., Feixas, M., Sbert, M., Heidrich, W.: Viewpoint selection using viewpoint entropy. In: Ertl, T., Girod, B., Greiner, G., Niemann, H., Seidel, H.-P. (eds.) Proceedings of the vision modeling and visualization conference (VMV 2001), Stuttgart, pp. 273–280. IOS Press, Amsterdam (2001)
- [22] Bober, M.: MPEG-7 visual shape descriptors. IEEE Trans. on Circuits and Systems for Video Technology 11(6), 716–719 (2001)
- [23] Avrithis, Y.S., Doulamis, N.D., Doulamis, A.D., Kollias, S.D.: Efficient Content Representation in MPEG Video Databases. In: Proc. of IEEE Inter. Workshop on Content-based Access of Image Archives (CBIR), Santa Barbara, USA (June 1998)
- [24] Morris, O.J., Lee, M.J., Constantinides, A.G.: Graph Theory for Image Analysis: an Approach based on the Shortest Spanning Tree. IEE Proceedings 133, 146–152 (1986)
- [25] Haykin, S.: Neural Networks: A Comprehensive Foundation, 2nd edn. Prentice Hall Press, Englewood Cliffs (1998)
- [26] Swiss Model, <http://swissmodel.expasy.org/>

Interactive Volume-Based Indirect Illumination of Dynamic Scenes

Athanasis Gaitatzes, Pavlos Mavridis, and Georgios Papaioannou

Abstract. In this paper we present a novel real-time algorithm to compute the global illumination of dynamic scenes with arbitrarily complex dynamic illumination. We use a virtual point light (VPL) illumination model on the volume representation of the scene. Unlike other dynamic VPL-based real-time approaches, our method handles occlusion (shadowing and masking) caused by the interference of geometry and is able to estimate diffuse inter-reflections from multiple light bounces.

1 Introduction

In order to synthesize photo-realistic images we need to capture the complex interactions of light with the environment. Light follows many different paths distributing energy among the object surfaces. This interplay between light and object surfaces can be classified as local illumination and global illumination. Local illumination algorithms take into account only the light which arrives at an object directly from a light source. Global Illumination algorithms, on the other hand, take into account the entire scene, where the light rays can bounce off the different objects in the environment or be obstructed and absorbed. Reflection, refraction and diffuse inter-reflection are examples of complex light interactions with a high computational cost that is usually not available for real-time applications.

Athanasis Gaitatzes
University of Cyprus, 75 Kallipoleos St.,
P.O.Box.20537, Nicosia CY-1678, Cyprus
e-mail: gaitat@yahoo.com

Pavlos Mavridis · Georgios Papaioannou
Athens University of Economics and Business,
Dept. of Informatics, 76 Patission St.,
Athens 10434, Greece
e-mail: pmaavridis@gmail.com, gepap@aeub.gr

In this paper we propose a method that produces photo-realistic images of diffuse, dynamic environments in real time, by estimating the illumination at discrete locations in the environment and applying the results on the scene geometry. This way, we can capture shadowing effects as well as diffuse inter-reflections from multiple secondary light bounces. The method we propose uses a uniform discretization of the scene, incorporating geometry information in the discretization structure. Instead of using the shadow map data as virtual point lights (VPLs) [3] [4] [9], our method performs a complete scene voxelization and is thus able to include occlusion information along with any direct, indirect and self-emitted illumination. Furthermore, it is capable of calculating global illumination from multiple light bounces and include energy from emissive materials in the process.

2 Previous Work

The use of a regular or hierarchical space discretization in global illumination is not new. Several non-real-time algorithms in the past have utilized volume-based acceleration methods and volume data caching to increase their performance. In the past two years, both the porting of global illumination algorithms to the GPU and the inception of new, real-time methods for approximating indirect lighting have gained significant interest from the research community and the game industry.

Radiosity based methods in voxel space have addressed the illumination problem, like Greger et al. [6] and Chatelier et al. [1] but their results were not computed in real-time and had large storage requirements. Modern advances of the same approach, Kaplanyan [9], yielded much faster results than before, but ignored indirect occlusion and secondary light bounces.

The Irradiance Volume, which was first introduced by Greger et al. [6], regards a set of single irradiance samples, parameterized by direction, storing incoming light for a particular point in space (i.e. the light that flowed through that point). The method had large storage requirements as neither an environment map nor spherical harmonics were used for the irradiance storage. With a set of samples they approximated the irradiance of a volume, which was generally time-consuming to compute but trivial to access afterwards. With an irradiance volume they efficiently estimated the global illumination of a scene.

Instant radiosity methods, introduced by Keller [10], approximate the indirect illumination of a scene using a set of VPLs. A number of photons are traced into the scene and VPLs are created at surface hit points, then the scene is rendered, as lit by each VPL. The major cost of this method is the calculation of shadows from a potentially large number of point lights but since it does not require any complex data structures it is a very good candidate for a GPU implementation. Lightcuts [18] reduce the number of the required shadow queries by clustering the VPLs in groups and using one shadow query per cluster, but the performance is still far from real time.

Reflective shadow maps [3] consider the pixels of a shadow map as VPLs, but the contribution of these lights is gathered without taking scene occlusion into

account. To achieve interactive frame rates, screen space interpolation is required and the method is limited to the first bounce of indirect illumination. An extension of this method by the same authors [4] reorganizes the computation of the indirect light to achieve better performance, but it still ignores occlusion for the indirect lighting. Imperfect shadow maps [14] use a point based representation of the scene to efficiently render extremely rough approximations of the shadow maps for all the VPLs in one pass. They achieve interactive frame rates but indirect shadows are smoothed out considerably by the imperfections and the low resolution of the shadow maps.

Jensen [7] introduced the concept of photon mapping, where in a first pass photons are traced from the light sources into the scene and stored in a k-d tree and in a second pass the indirect illumination of visible surface points is approximated by gathering the k nearest photons. McGuire [12] computes the first bounce of the photons using rasterization on the GPU, continues the photon tracing on the CPU for the rest of the bounces and finally scatters the illumination from the photons using the GPU. Since part of the photon tracing still runs on the CPU, a large number of parallel cores are required to achieve interactive frame-rates.

Ritchell [15] extends previous methods for screen space ambient occlusion calculation [16] and introduces a method to approximate the first indirect diffuse bounce of the light by only using information in the 2D frame buffer. This method has a very low computational cost but the resulting illumination is hardly accurate since it depends on the projection of the (visible only) objects on the screen.

The concept of interpolating indirect illumination from a cache was introduced by [20]. Accurate irradiance estimates are computed using ray tracing on a few surface points (irradiance sample points) and for the remaining surface points fast interpolation is used. Wang [19] presents a method to calculate the irradiance sample points in advance and implements the algorithm on the GPU. The method is accurate but it achieves interactive frame rates only in very simple scenes.

Nijasure [13] uses spherical harmonics to store the incoming radiance of the scene in a uniform grid structure. The surfaces are rendered by interpolating the radiance from the closest grid points. This method supports multiple bounces and indirect occlusion but it's very expensive because it requires the complete scene to be rendered in a cube map for the radiance estimation on each grid point.

Kaplanyan [9] also uses a regular grid to store the scene radiance, but for its calculation he uses a propagation scheme to calculate the radiance distribution on the scene. Radiance is initially injected in VPL positions and then it is iteratively propagated through empty space. The method achieves very high performance but completely ignores occlusion for the indirect light and secondary bounces.

3 Mathematical Background

3.1 Review of Spherical Harmonics

The spherical harmonics (SH) are a set of orthonormal basis functions defined over a sphere, in the same manner that the Fourier series is defined over an N -dimensional

periodical signal. The *Spherical Harmonic* (SH) functions in general are defined on imaginary numbers, but since in graphics we are interested in approximating real functions over the sphere (i.e. light intensity fields), we use the real spherical harmonics basis. Given the standard parameterization of points on the surface of the unit sphere into spherical coordinates

$$(\sin \theta \cos \phi, \sin \theta \sin \phi, \cos \theta) \rightarrow (x, y, z)$$

the the real SH basis functions of order l is defined as:

$$Y_l^m(\theta, \phi) = \begin{cases} \sqrt{2} K_l^m \cos(m\phi) P_l^m(\cos \theta) & m > 0 \\ \sqrt{2} K_l^m \sin(-m\phi) P_l^{-m}(\cos \theta) & m < 0 \\ K_l^0 P_l^0(\cos \theta) & m = 0 \end{cases} \quad (1)$$

where $l \in \mathbb{R}^+$, $-l \leq m \leq l$, P_l^m is the *associated Legendre polynomial*, K_l^m is the scaling factor to normalize the functions and is defined as:

$$K_l^m = \sqrt{\frac{(2l+1)}{4\pi} \frac{(l-|m|)!}{(l+|m|)!}} \quad (2)$$

The spherical harmonics possess several important properties, such as rotation invariance. This means that the SH approximation f of a spherical function f rotated by some rotation operator R is the same regardless of the order of application of the rotation: rotating the SH projection of f gives the same approximation as rotating f and then projecting it. This prevents aliasing artifacts from occurring while rotating functions and means that we can simply rotate the projection of a function instead of re-projecting a rotated function.

Similar to the Fourier series expansion, a function on the sphere $f(\theta, \phi)$ can be represented in terms of *spherical harmonics coefficients* $f_{l,m}$ as:

$$f(\theta, \phi) = \sum_{l=0}^{\infty} \sum_{m=-l}^l f_{l,m} Y_l^m(\theta, \phi) \quad (3)$$

A signal over a sphere is approximately reconstructed using a a truncated SH series, by projecting the initial function f onto a finite set of SH coefficients up to order $l = n$, $l \in \mathbb{N}$. Typically, in computer graphics a maximum order of 6 is used.

Additionally, because of orthnormality, the integral of two reconstructed spherical functions that have been projected in the SH basis is reduced to the inner product of the vectors of their SH coefficients. For band-limited SH functions of order n , the integral becomes:

$$\int \tilde{f}(\theta, \phi) \tilde{g}(\theta, \phi) = \sum_{l=0}^n \sum_{m=-l}^l f_{l,m} g_{l,m} \quad (4)$$

3.2 Radiance Transfer

In order to accurately model light in an environment, The complete energy transfer has to be evaluated on each surface location. The *Rendering Equation*, proposed by Kajiya [8], associates the outgoing radiance $L_o(\mathbf{x}, \omega_o)$ from a surface point \mathbf{x} along a particular viewing direction ω_o , with the intrinsic light emission $L_e(\mathbf{x}, \omega_o)$ at \mathbf{x} and the incident radiance from every direction ω_i in the hemisphere Ω above \mathbf{x} , using a BRDF that depends only on the material properties and the wavelength of the incident light. The hemisphere-integral form of the rendering equation can be written as:

$$L_o(\mathbf{x}, \omega_o) = L_e(\mathbf{x}, \omega_o) + \int_{\Omega} L_i(\mathbf{x}, \omega_i) f_r(\mathbf{x}, \omega_i \rightarrow \omega_o) \cos(\theta) d\omega_i \quad (5)$$

where $f_r(\mathbf{x}, \omega_i \rightarrow \omega_o)$ is the bidirectional reflectivity distribution function of the surface at point \mathbf{x} , expressing how much of the incoming light arriving at \mathbf{x} along direction ω_i is reflected along the outgoing direction ω_o . $L_i(\mathbf{x}, \omega_i)$ is the light arriving along direction ω_i .

If we group the cosine term and f_r into a single *transfer function* T , (Sloan et al. [17]), which expresses how point \mathbf{x} responds to incoming illumination, then Equation 5 becomes:

$$L_o(\mathbf{x}, \omega_o) = L_e(\mathbf{x}, \omega_o) + L_r(\mathbf{x}, \omega_o) = L_e(\mathbf{x}, \omega_o) + \int_{\Omega} L_i(\mathbf{x}, \omega_i) T(\mathbf{x}, \omega_i \rightarrow \omega_o) d\omega_i \quad (6)$$

The above generic energy transfer equation can be used in fact to model any variation of the rendering equation. In our case, where a point in space (voxel center) is illuminated, it is more convenient to consider an integral over the entire sphere surrounding the voxel center. Furthermore, the voxel center can behave as a spherical particle, receiving energy with maximum flow from every direction. Therefore, the cosine term is dropped as the projected solid angle towards the emitting location along ω_i always equals $d\omega_i$.

Using the orthonormality property of the SH function basis (Eq. 4) and considering for the moment only the reflected radiance, the integral in Eq. 6 can be approximated with a finite set of terms as:

$$L_r(\mathbf{x}, \omega_o) \approx \sum_{l=0}^n \sum_{m=-l}^l L_{l,m} T_{l,m} \quad (7)$$

4 Method Overview

We have extended the work of Kaplanyan [9], in order to take into account occlusion in the light transfer process and secondary light bounces. Our method is based on the full voxelization of the geometry instead of the injection of only the reflection shadow map points (VPLs) in a volume grid. This way, the presence of

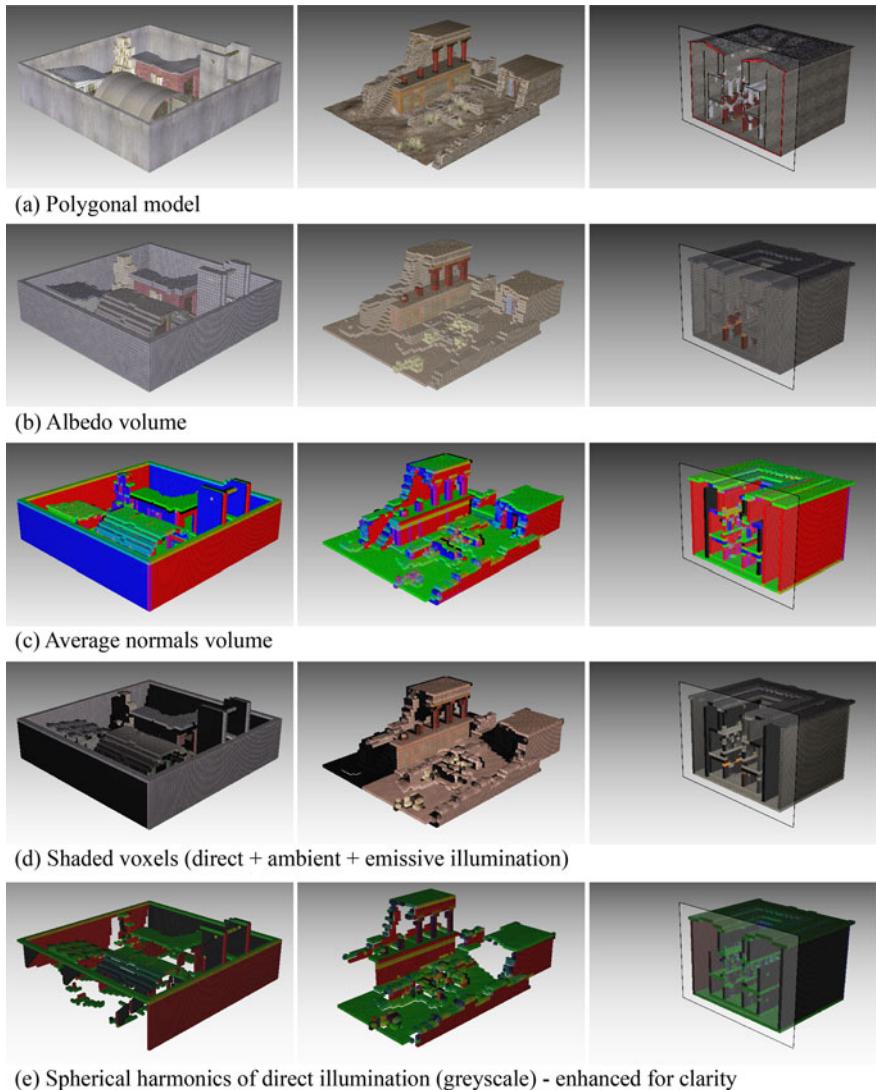


Fig. 1 Several environments voxelized into a 64^3 grid. Column 1: a model of 10,220 triangles (arena). Column 2: a model of 109,170 triangles (Knossos). Column 3: the sponza II atrium of 135,320 triangles (cross section depicted). All buffers are of floating point precision.

geometry that is unlit by the direct illumination is also known and light interception and reflection is possible. The voxelization records — among others — direct illumination and scalar occupancy data, thus enabling the indirect illumination from emissive materials and the transmission through transparent elements.

Our method consists of three stages. First the scene (or a user-centered part of it) is discretized to a voxel representation. Next, the radiance of each voxel is iteratively propagated in the volume and finally, during image rendering, the irradiance of each surface point is calculated by sampling the radiance from the nearest voxels.

To this effect, we use several 3D volume buffers. An *accumulation volume buffer* is used for the storage of radiance samples when light bounces off occupied voxels. This buffer is sampled during the final rendering pass to reconstruct the indirect illumination. For each color band it stores a spherical harmonic representation of the radiance of the corresponding scene location (4 coefficients encoded as RGBA float values). It is initialized with zero radiance. For the iterative radiance distribution, a *propagation volume buffer* is used (see Section 4.2). The propagation buffer stores a spherical harmonic representation of the radiance to be propagated in the next propagation iteration. It is initialized with the radiance from the first bounce VPLs (direct illumination). Both the accumulation and the propagation buffers are read and write buffers. Our algorithm also reads information from one more read-only volume buffer that contains information about the scene normals and surface albedo (see Fig. II(b),(c)). Average normals and space occupation (scalar voxelization value, also accounting for transparency) are compacted into a single voxel value.

To discretize our scene in real time, we create a uniform spatial partitioning structure (voxel grid - see Section 4.1) on the GPU, where we store the geometry and radiance samples. We inject VPLs in our voxel space, which are essentially hemispherical lights with a cosine falloff. The VPLs are then approximated by a compact spherical harmonic coefficients representation. See Fig. II(d) and (e) respectively. Similar to Kaplanyan [9], we use an iterative diffusion approach on the GPU to propagate energy within space. In contrast to [9] though, since we obtain the space occupation information from the voxelization (not-just VPLs), energy is propagated only in void space, from one voxel boundary to the next. The propagated radiance is reflected on occupied (voxelized) volume grid points and accumulated at these locations in the accumulation buffer (see Section 4.2). The new propagation direction is determined by the stored average voxel normal of the occupied voxel.

During the rendering pass (see Section 4.3), for each fragment, the volume is queried as a texture and the closest texels (accumulated irradiance) are used to estimate the global illumination at that point in the scene.

4.1 Real-Time Voxelization

Instead of applying one of the fast binary GPU voxelization methods, such as Eismann's et al. [5], we use a variant of Chen's et al. algorithm [2] because we need to store multi-channel scalar data in each voxel. More specifically, we use the same steps as Chen's algorithm, for the slicing of the volume. The main difference is that we do not use the originally proposed XOR operation because in practice, very few models are watertight and many volume attributes cannot be defined for interior voxels. Therefore, our method produces only volume shells.

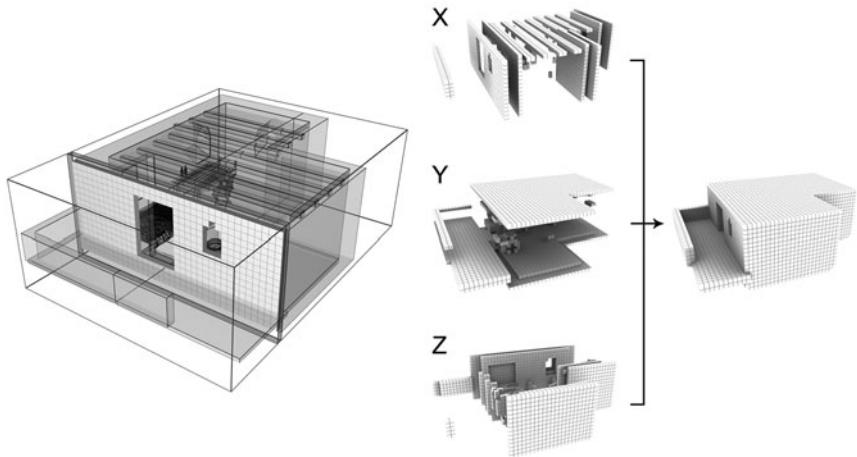


Fig. 2 Slice-based voxelization (left) and composition of the three sub-volume passes into one voxelized volume (right)

In brief, for every volume slice (see Algorithm 1), a conventional scan-conversion of the scene geometry takes place and the generated fragments correspond to the voxels of this slice (Fig. 2). Rasterization is incremental and requires that the slope of the dependent variable on the increment is less than 1. As far as the scan conversion of a polygon onto the (slice-oriented) view plane is concerned, there are no holes generated but when the fragments are stored as voxels, discontinuities along the Z-axis occur (slicing direction). The XOR operation indirectly solved this problem (by filling in the missing fragments) but in our case this is not an option. The problem is solved by repeating the scan-conversion process 3 times, once for each primary axis. This way, we ensure that the depth-discontinuity in one orientation of the view plane will be remedied in one of the two others (see Fig. 2(right)).

Algorithm 1. Scene Voxelization

```

generate a bounding box of the scene;
for  $i \leftarrow 1$  to  $N$  volume slices do
    define a voxel-deep, thin orthogonal viewing frustum along X-axis;
    execute 2D scan conversion for all object faces;
    store result in slice  $i$  of volume buffer-X;
end
repeat above loop for Y and Z axes;
combine the three temporary volumes, buffer[XYZ], into one final volume keeping the
MAX value for each corresponding cell in all three volumes;

```

During the above 3-way voxelization, the radiosity of each grid cell is computed using direct illumination (complete with shadows and emissive illumination). Three buffers on the GPU are needed to store the temporary volume results of the three slicing procedures (one for each different orientation of the object). Finally, those three volumes are combined into one buffer, using the MAX frame buffer blending operation. The maximum radiosity of each cell is stored as a spherical harmonic representation. These values will be used as the initial radiance distribution in the propagation buffer for the iterative radiance distribution.

4.2 Iterative Radiance Distribution

Once we have injected the VPLs into the initial 3D volume, we need to propagate their initial radiance to their neighboring voxels. The propagation stage consists of several sequential iterations performed entirely on the GPU. Each iteration represents one discrete step of radiance propagation in the (empty) 3D volume. We effectively perform radiance shooting at each volume location by gathering radiance instead from each one of the voxel's neighbors (see Fig. 3) and interpolating the weighted sum of the corresponding directional contributions on the GPU.

Similar to [9], we split the integral of radiance gathering (Eq. 6 for spherical integration domain) into six sub-domains corresponding to the six sides of the receiving voxel. Instead of considering only unobstructed propagation, though, the transfer function $T_{j \rightarrow i}$ between the neighboring voxel j and the current voxel i is split into a geometric (transfer) term $T_{cone(j)}$ and a reflective term $T_{r(j)}$. The four $T_{cone(j)}$ coefficients are pre-computed from the six rotated spherical harmonic functions of a 90-degree cone. $T_r(j)$ is used for the deflection of the incident radiance.

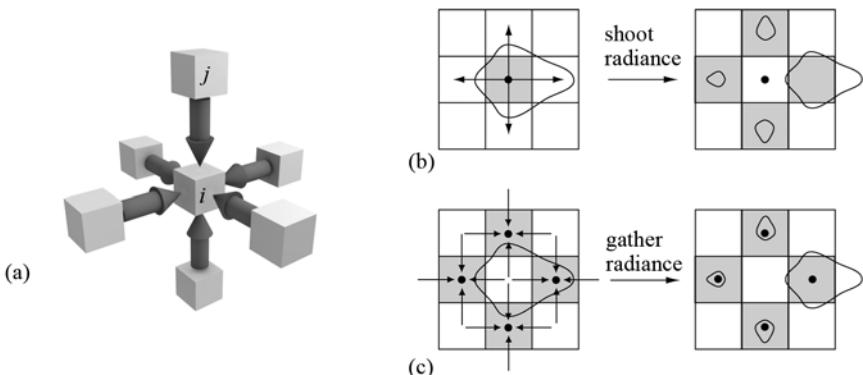


Fig. 3 Radiance Gathering Illustration (a). The radiance for the center voxel is gathered from the values stored at the voxels of the surrounding cells. Radiance shooting (b) in the radiance propagation procedure is equivalent to radiance gathering (c).

When voxel i corresponds to void space, radiance is propagated in the direction from voxel j to i . This means that when no obstacle is encountered, $T_r(j)$ coefficients are equal to 1. On the other hand, when voxel i is occupied, the spherical harmonic function of the incident radiance from voxel j should be mirrored with respect to the plane that is perpendicular to the plane of reflection and parallel to the average normal direction stored in the volume buffer. This requires two SH rotations and a mirroring operation. See [11] for details on the rotation of real spherical harmonics. For speed and simplicity though, this operation is replaced by a mirror reflection on the voxel boundary, i.e. along one of the three primary axes. Therefore, the four coefficients of $T_{r(j)}$ are 1 except the one corresponding to the mirror direction. Taking into account the above factors, the gathering operation becomes:

$$\int_S L_i(\mathbf{x}, \omega_i) T(\mathbf{x}, \omega_i \rightarrow \omega_o) d\omega_i = \sum_{j=1}^6 \sum_{l=0}^1 \sum_{m=-l}^l L_{(j),l,m} T_{cone(j),l,m} T_{r(j),l,m} \quad (8)$$

Figure 4 demonstrates the propagation and radiance accumulation process.

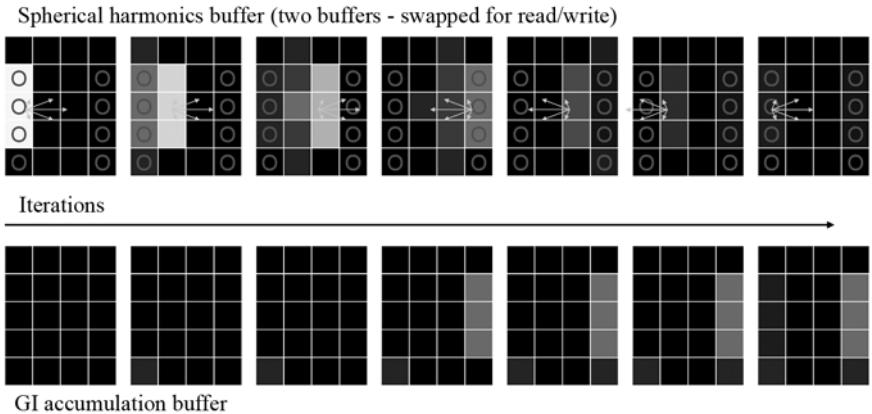


Fig. 4 Simplified example of the propagation and light reflection process

4.3 Final Illumination Reconstruction

During the final rendering pass, the irradiance at each surface point is computed from the incident radiance L_i that is stored in our uniform grid structure (accumulation buffer). The irradiance E at point \mathbf{x} can be derived by integrating the definition of incident radiance over the hemisphere above \mathbf{x} :

$$E(\mathbf{x}) = \int_{\Omega} L_i(\mathbf{x}, \omega_i) \cos \theta d\omega_i \quad (9)$$

where θ is the angle between the surface normal and the incident radiance direction ω_i . The integration domain is the hemisphere Ω defined by the surface normal \mathbf{n}_x at point \mathbf{x} .

In order to include the color filtering at the final gathering stage as well as the material emission, we can estimate the radiosity $B(\mathbf{x})$. For diffuse surfaces, $B(\mathbf{x})$ is given by the following hemisphere-integral equation, after multiplying Eq. 5 with π (and hence L_i):

$$B(\mathbf{x}) = B_e(\mathbf{x}) + \frac{\rho(\mathbf{x})}{\pi} \int_{\Omega} B_i(\mathbf{x}, \omega_i) \cos \theta d\omega_i \quad (10)$$

where $\rho(\mathbf{x})$ is the albedo of the surface. If we change the integration domain to the full sphere Ω' , the previous equation can be rewritten as follows:

$$B(\mathbf{x}) = B_e(\mathbf{x}) + \frac{\rho(\mathbf{x})}{\pi} \int_S B_i(\mathbf{x}, \omega_i) T(\mathbf{n}_x, \omega_i) \omega_i \quad (11)$$

where the function T is defined as follows:

$$T(\mathbf{n}_x, \omega_i) = \begin{cases} \cos \theta, & \theta < \pi/2 \\ 0, & \theta > \pi/2 \end{cases} \quad (12)$$

This change of the integration domain is necessary because we are going to use spherical harmonics, which are defined over the sphere and not the hemisphere.

Equation 11 is directly evaluated per pixel to give the final indirect lighting. In our algorithm the radiance L is tri-linearly interpolated from the stored values in the uniform grid structure. From the eight closest grid points only the ones corresponding to occupied voxels are considered for interpolation. L_i is already stored and interpolated in spherical harmonic representation. We also build a spherical harmonic representation for the function T , as described in [9] and the integral is calculated per pixel as a simple dot product, as shown in Eq. 4.

5 Results

We have integrated the above algorithm in a real time deferred renderer using OpenGL and GLSL. Our proof of concept implementation uses a 2nd order spherical harmonic representation, since the four SH coefficients, map very well to the four component buffers supported by the graphics hardware. All results are rendered on an nVIDIA GeForce GTX285 at 512x512 pixels with a 32³ grid size. It should be noted here that, excluding the final interpolation stage, the performance of the indirect lighting computation in our method does not depend on the final screen resolution, but only on the voxel grid size and the number of propagation steps. This is a big advantage over instant radiosity methods, like imperfect shadow maps.

Table 1 shows comprehensive time measurements for all the scenes detailed below. All scenes are considered to have fully dynamic geometry and lighting

Table 1 Time measurements of our test scenes in milliseconds. Only the voxelization and propagation times are relevant to our work. The total rendering time includes the direct lighting computation and other effects and is given as a reference.

	triangles	grid size	iterations	voxelization (ms)	propagation (ms)	total time (ms)
test scene	48	32^3	11	10	12	22
room	704	32^3	64	3	61	69
arena	10219	32^3	12	3	13	21
sponza	66454	32^3	11	10	11	28

conditions. In all cases our algorithm achieves real time frame rates and sufficient accuracy in the reproduction of the indirect diffuse illumination, even though our implementation is not optimized in any way.

We have found that the propagation stage of our method is limited by the available memory bandwidth and not the computational speed. This is to be expected, since the propagation kernel requires 52 floating point reads and 8 floating point writes per color band. To save memory bandwidth we do not store the diffuse color of the surfaces in the voxel structure, but after the first light bounce we consider it constant and equal to 0.5 for each color band.

Figure 5 shows a direct comparison of our algorithm with a reference solution on a simple test scene. We can see that our method reproduces the shape and the properties of the indirect illumination in the reference solution.

Figure 6 shows a room lit through a set of open windows. This is a challenging scene for global illumination algorithms, because only a small region on the left wall is directly lit by the sun and the majority of the lighting in the room is indirect. We can see that the simple light propagation method completely fails to reproduce the indirect lighting in the room, since it is not taking into account secondary light bounces and occlusion. At least two bounces of indirect lighting are required to get

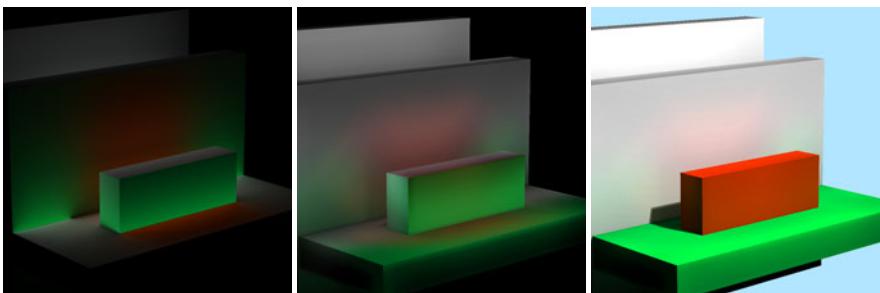


Fig. 5 From left to right: reference solution computed with ray tracing (indirect illumination only), our solution (indirect illumination only) and final image with direct and indirect lighting

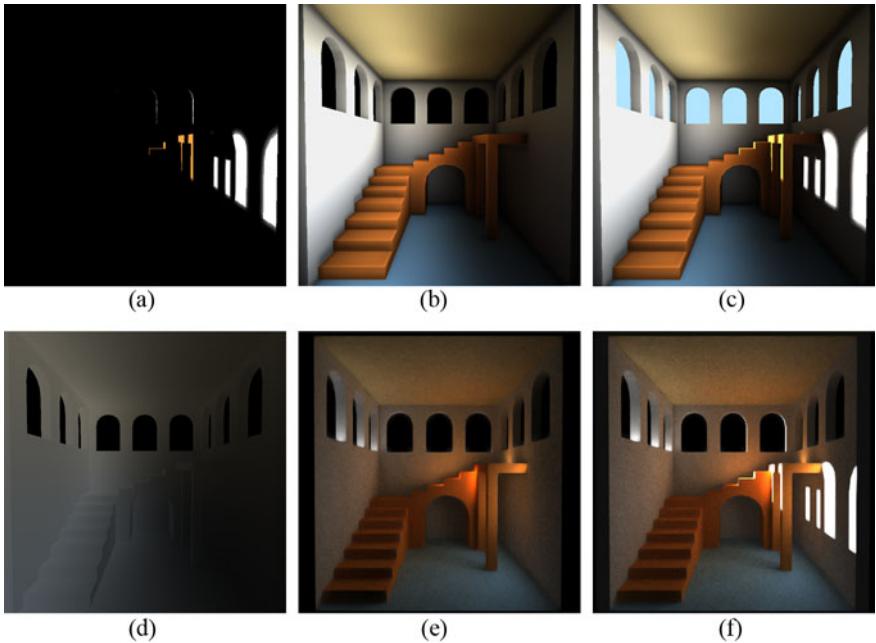


Fig. 6 (a) The room scene lit with direct lighting only. (b) Radiosity with 64 iterations. (c) Direct and indirect illumination using our method. (d) The indirect illumination using light propagation volumes [2]. (e) Reference radiosity using 2-bounce path tracing. (f) Reference final image using path tracing.

meaningful results in this case. In our method, when a grid of size N is used, the distance between the walls of the room is also N , so kN propagation steps are required to compute k bounces of indirect illumination. This is a worst case scenario, as in typical scenes light interaction from one end of the scene to the other is not required. In this particular case we have used 64 propagation steps to simulate two bounces of light on a 32^3 grid. The resulting illumination is visually pleasing, giving high contrast on the edge of the walls and the staircase. Since our algorithm takes indirect occlusion in consideration, the area below the staircase is correctly shadowed. We observe some artifacts below the windows, due to the imprecision of the spherical harmonics and the fact that the grid cell on this area covers both the edge of the wall and the empty space inside the window. Even with a number of propagation steps this high, our method maintains easily an interactive frame-rate since the propagation stage takes only 61 ms to complete.

A nice characteristic of our method is the predictable performance of the propagation stage. We can easily calculate the time for the propagation step for each individual voxel. This time is constant and independent from the scene complexity. It should be noted of course that the performance may be constant and predictable, but the same is not true for the accuracy and the quality of the resulting illumination.

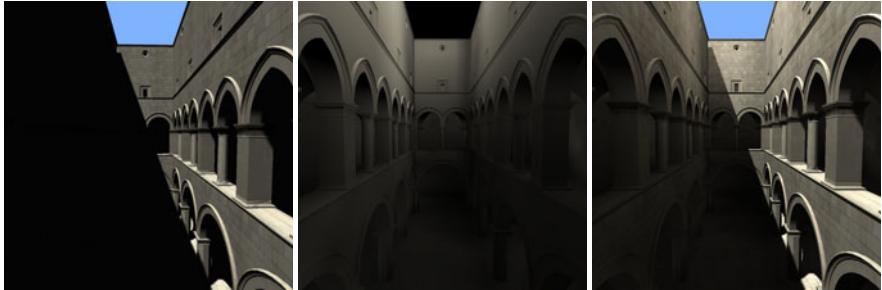


Fig. 7 From left to right: direct lighting, indirect illumination only and final image with direct and indirect lighting



Fig. 8 From left to right: direct only lighting, indirect illumination using our method and final image with direct and indirect lighting

Figure 7 shows the Sponza Atrium II, a typical scene in the global illumination literature. The right part of the scene is directly lit by the sun, the left one is lit only indirectly. As we can see, using only eleven propagation steps our method successfully reproduces the low-frequency indirect illumination which is dominant on the left part of the scene with very few visible artifacts.

Figure 8 shows an enclosed arena scene, a typical outdoor scene in video games. Twelve propagation steps are used in this case and we can see that the resulting indirect illumination greatly improves the visual quality of the final image.

5.1 Discussion

A nice feature of our method is that for scenes with static or smoothly changing geometry and lighting conditions, the cost of the indirect illumination can be amortized among many frames without introducing any visible artifacts. In other words, the rate of indirect lighting updates can be reduced to meet the final performance goals. For scenes with good temporal coherence — hence, with slow illumination changes — it is possible to perform the 3-way voxelization in an interleaved manner

(one direction per frame). In this case the volume is completely updated after three frames but the voxelization cost is reduced by a factor of three.

Since voxelization is a rough discretization of the scene geometry, secondary shadows from small scale geometric details cannot be reproduced accurately by our method. Higher voxel resolutions can always be used, but with a performance hit. Also, due to graphics hardware limitations, we only used second order spherical harmonics, which they do not have sufficient accuracy to represent high frequency indirect light. This is not crucial if the direct illumination covers large parts of a scene yielding only very low-frequency indirect shadows in the first place. Interestingly, imperfect shadow maps have exactly the same issue (but for different reasons) but we think that our method is preferable since it does not require the maintenance of a secondary point based scene representation and the performance is mostly independent from final image resolution.

The performance and quality of our method depends on two parameters: the volume resolution and the number of iterations. Empirically, we have found that a grid size of 32 is sufficient in most cases. For outdoor scenes we have found that a low iteration count (around 12) is sufficient but for indoor ones a much higher iteration count is required (around 64) to accurately simulate the bouncing of the light inside the building.



Fig. 9 The Knossos model (<http://graphics.cs.aueb.gr/downloads/knossos.rar>). From left to right: direct lighting, radiosities using our method and final image with direct and indirect lighting.

6 Conclusion and Future Work

We have presented a new method for the computation of indirect diffuse light transport in large and fully dynamic scenes in real-time. Unlike previous work, our method takes into account indirect occlusion and secondary light bounces. We have demonstrated that our method gives good results in a variety of test cases and always maintains a high frame rate. Since the test results showed that the voxelization step is relatively costly, in the future we intent to introduce a much faster voxelization scheme. Furthermore, the possibility of a more accurate but still manageable radiance deflection mechanism will be investigated. Finally, another

interesting direction of research is to extend this method to take specular light transport in to account.

References

1. Chatelier, P.Y., Malgouyres, R.: A low-complexity discrete radiosity method. *Computers and Graphics* 30(1), 37–45 (2006), <http://dx.doi.org/10.1016/j.cag.2005.10.008>
2. Chen, H., Fang, S.: Fast voxelization of three-dimensional synthetic objects. *Journal Graphics Tools* 3(4), 33–45 (1998)
3. Dachsbacher, C., Stamminger, M.: Reflective shadow maps. In: I3D 2005: Proceedings of the 2005 Symposium on Interactive 3D Graphics and Games, pp. 203–231. ACM, New York (2005), <http://doi.acm.org/10.1145/1053427.1053460>
4. Dachsbacher, C., Stamminger, M.: Splatting indirect illumination. In: I3D 2006: Proceedings of the 2006 Symposium on Interactive 3D Graphics and Games, pp. 93–100. ACM, New York (2006), <http://doi.acm.org/10.1145/1111411.1111428>
5. Eisemann, E., Décoret, X.: Single-pass gpu solid voxelization for real-time applications. In: GI 2008: Proceedings of Graphics Interface 2008, pp. 73–80. Canadian Information Processing Society, Toronto (2008)
6. Greger, G., Peter Shirley, Y., Hubbard, Z.P.M., Greenberg, D.P.: The irradiance volume. *IEEE Computer Graphics and Applications* 18, 32–43 (1998)
7. Jensen, H.W.: Global illumination using photon maps. In: Proceedings of the Eurographics Workshop on Rendering Techniques 1996, pp. 21–30. Springer, London (1996)
8. Kajiya, J.T.: The rendering equation. *SIGGRAPH Comput. Graph.* 20(4), 143–150 (1986), <http://doi.acm.org/10.1145/15886.15902>
9. Kaplanyan, A.: Light propagation volumes in cryengine 3. In: SIGGRAPH 2009: ACM SIGGRAPH 2009 courses, ACM, New York (2009)
10. Keller, A.: Instant radiosity. In: SIGGRAPH 1997: Proceedings of the 24th annual conference on Computer graphics and interactive techniques, pp. 49–56. ACM Press/Addison-Wesley Publishing Co, New York (1997), <http://doi.acm.org/10.1145/258734.258769>
11. Krivánek, J., Kontinen, J., Pattanaik, S., Bouatouch, K., Žára, J.: Fast approximation to spherical harmonics rotation. In: SIGGRAPH 2006: ACM SIGGRAPH 2006 Sketches, p. 154. ACM, New York (2006), <http://doi.acm.org/10.1145/1179849.1180042>
12. McGuire, M., Luebke, D.: Hardware-accelerated global illumination by image space photon mapping. In: HPG 2009: Proceedings of the Conference on High Performance Graphics 2009, pp. 77–89. ACM, New York (2009), <http://doi.acm.org/10.1145/1572769.1572783>
13. Nijasure, M., Pattanaik, S., Goel, V.: Real-time global illumination on the GPU. *Journal of Graphics Tools* 10(2), 55–71 (2005)
14. Ritschel, T., Grosch, T., Kim, M.H., Seidel, H.P., Dachsbacher, C., Kautz, J.: Imperfect shadow maps for efficient computation of indirect illumination. *ACM Trans. Graph* 27(5), 1–8 (2008), <http://doi.acm.org/10.1145/1409060.1409082>
15. Ritschel, T., Grosch, T., Seidel, H.P.: Approximating dynamic global illumination in image space. In: I3D 2009: Proceedings of the 2009 Symposium on Interactive 3D Graphics and Games, pp. 75–82. ACM, New York (2009), <http://doi.acm.org/10.1145/1507149.1507161>

16. Shanmugam, P., Arikan, O.: Hardware accelerated ambient occlusion techniques on gpus. In: I3D 2007: Proceedings of the 2007 Symposium on Interactive 3D Graphics and Games, pp. 73–80. ACM, New York (2007),
<http://doi.acm.org/10.1145/1230100.1230113>
17. Sloan, P.P., Kautz, J., Snyder, J.: Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In: SIGGRAPH 2002: Proceedings of the 29th annual conference on Computer graphics and interactive techniques, pp. 527–536. ACM, New York (2002),
<http://doi.acm.org/10.1145/566570.566612>
18. Walter, B., Fernandez, S., Arbree, A., Bala, K., Donikian, M., Greenberg, D.P.: Lightcuts: a scalable approach to illumination. ACM Trans. Graph. 24(3), 1098–1107 (2005),
<http://doi.acm.org/10.1145/1073204.1073318>
19. Wang, R., Wang, R., Zhou, K., Pan, M., Bao, H.: An efficient gpu-based approach for interactive global illumination. In: SIGGRAPH 2009: ACM SIGGRAPH 2009 papers, pp. 1–8. ACM, New York (2009),
<http://doi.acm.org/10.1145/1576246.1531397>
20. Ward, G.J., Rubinstein, F.M., Clear, R.D.: A ray tracing solution for diffuse interreflection. In: SIGGRAPH 1988: Proceedings of the 15th annual conference on Computer graphics and interactive techniques, pp. 85–92. ACM, New York (1988),
<http://doi.acm.org/10.1145/54852.378490>

Author Index

- Annuth, Hendrik 47
- Bardis, Georgios 105, 165
- Bastanfarid, Azam 185
- Bohn, Christian-A. 47
- Breton, T. 1
- Chendeb, Safwan 25
- Chronis, Evangelos 211
- Cussat-Blanc, S. 1
- Dandachy, Nancy 25
- de Freitas, Sara 123
- Delepouille, Samuel 145
- Doulamis, Anastasios 67
- Doulamis, Nikolaos 211
- Dragonas, John 67
- Dunwell, Ian 123
- Duthen, Y. 1
- El Hassan, Bachar 25
- Gaitatzes, Athanasios 229
- Ghazanfarpour, Djamchid 105, 165
- Izadi, Ali 185
- Lassabe, N. 1
- Leon, Alejandro 85
- Luga, H. 1
- Mavridis, Pavlos 229
- Miaoulis, George 67, 211
- Miaoulis, Georgios 105, 165
- Panzoli, David 123
- Papaioannou, Georgios 229
- Pascalie, J. 1
- Peters, Christopher 123
- Petridis, Panagiotis 123
- Plemenos, Dimitri 25, 67, 105, 165, 211
- Preux, Philippe 145
- Protopsaltis, Aristidis 123
- Renaud, Christophe 145
- Rezaei (Ghahroudi), Mahdi 185
- Rousselle, Fran ois 145
- Sanchez, St phane 123
- Scesa, Vincent 123
- Torres, Juan C. 85
- Yiakoumetsis, Christos P. 105, 165
- Zaldivar, Angel A. 85