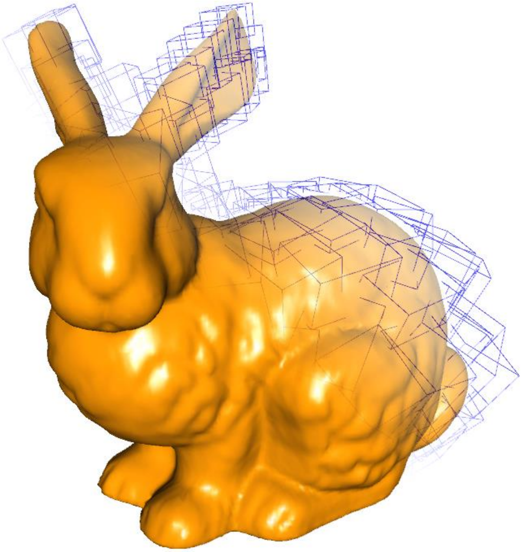
Optimizing Collision Detection based on OBB Trees Generated with a Genetic Algorithm

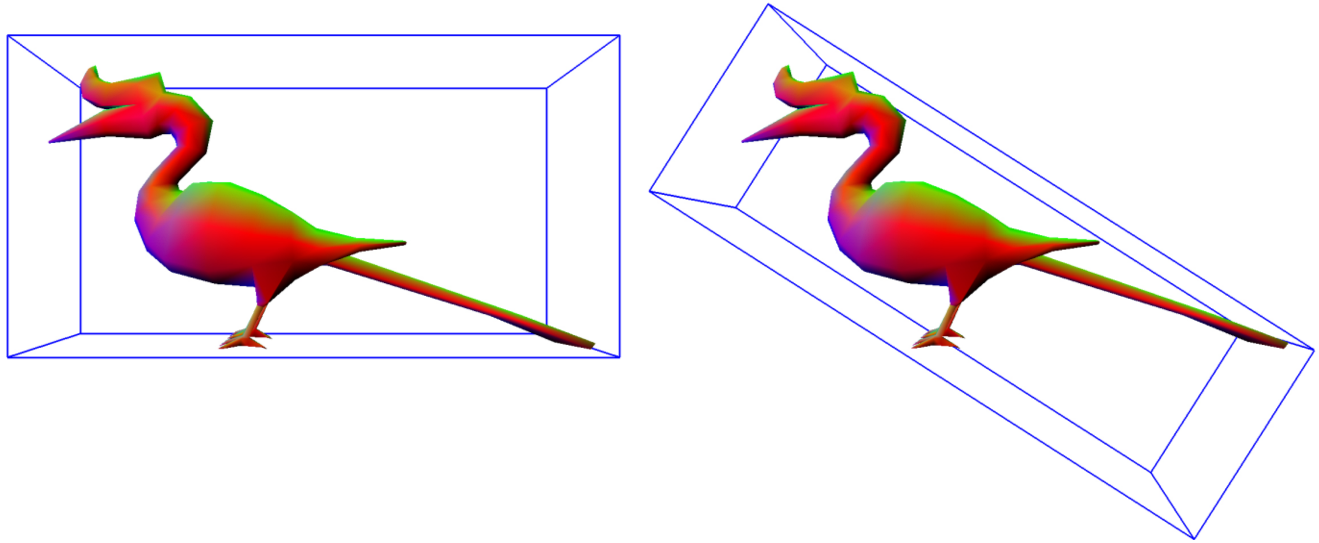
E. Ramírez, H. Navarro,
R. Carmona, J. Dos Ramos
hector.navarro@ciens.ucv.ve

Universidad Central de Venezuela



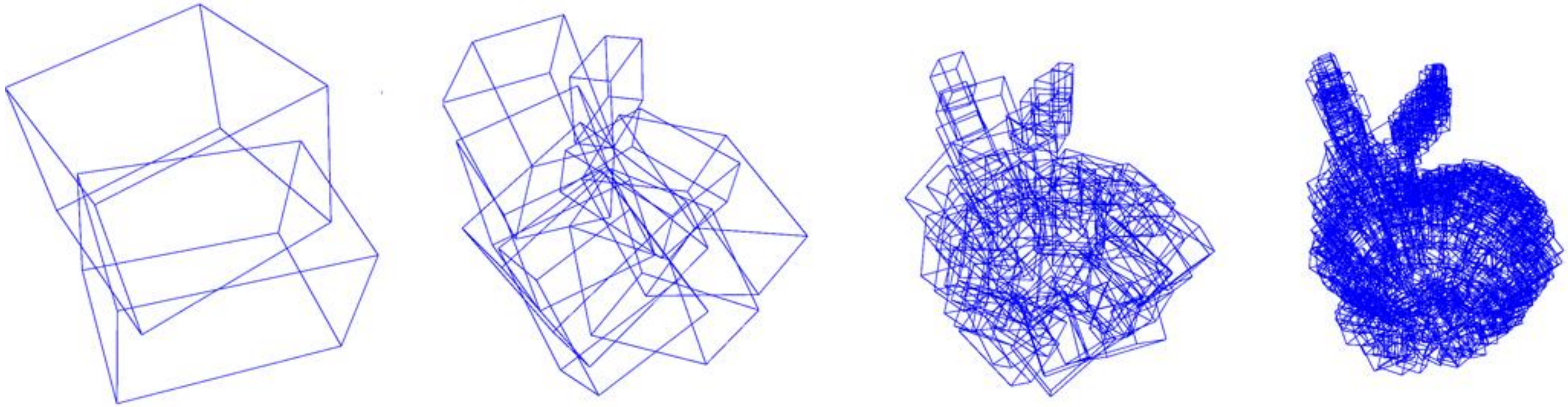
Introduction

- Collision detection is a very important field in computer graphics
- Hierarchical CD techniques are widely used
- Performance of hierarchical techniques depends heavily on accurateness of the bounding volume



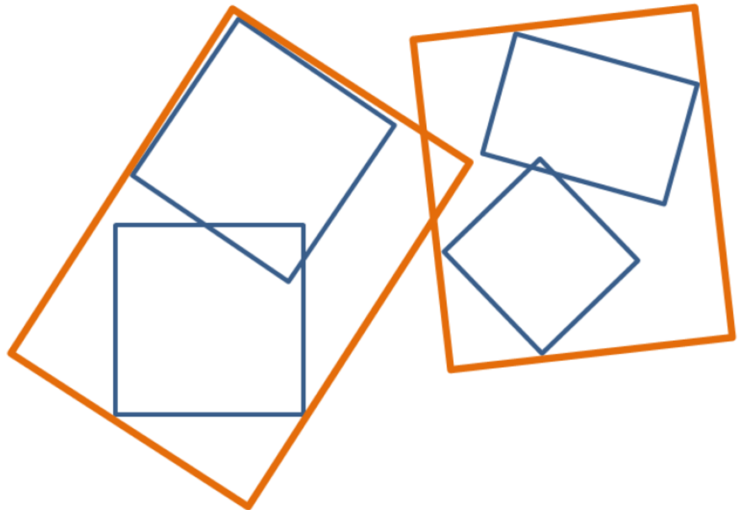
Collision Detection

- Checking, **if**, **when** and **where** two or more objects of the scene have collided
- Improves realism on the scene
- Hierarchical methods decompose objects in bounding volumes recursively



Collision Detection

Consider the following 2D example:



- In order to check for collisions, at first the highest level bounding volumes (orange) must be intersected
- Since those bounding volumes collide, it is necessary to check for collisions at the second level (blue)
- At this level a collision is discarded, because there is no intersection between bounding volumes of different objects

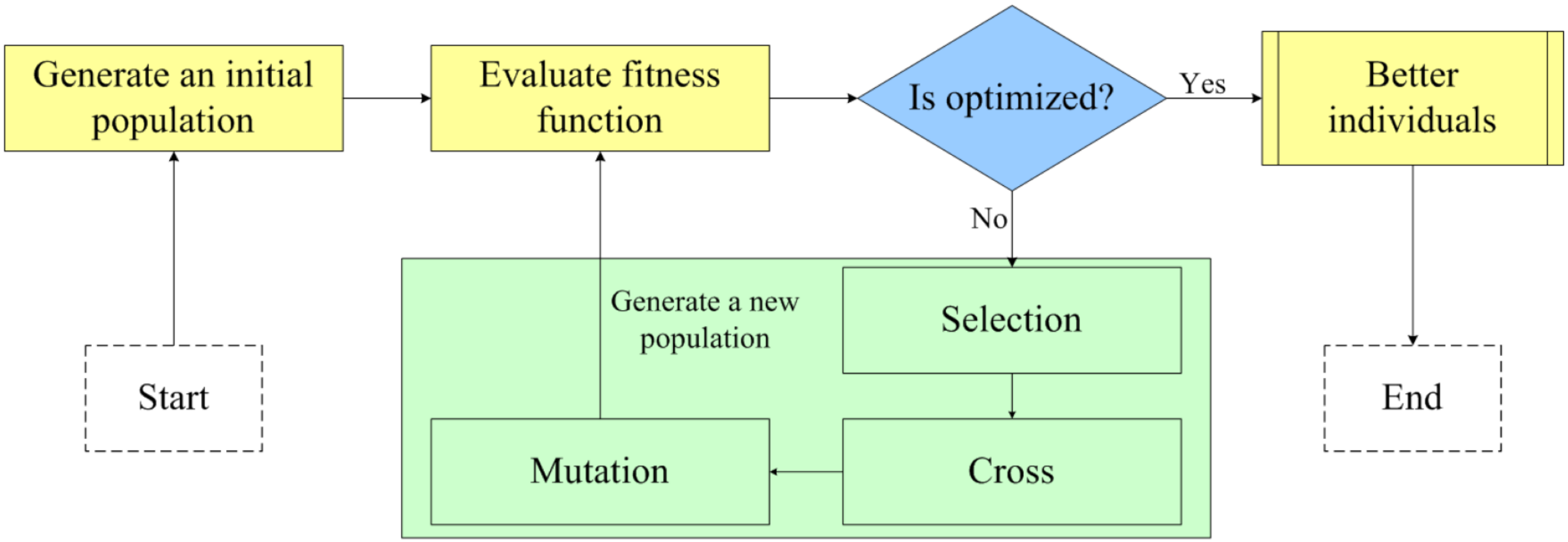
OBB Trees

Some desirable features of OBB trees are:

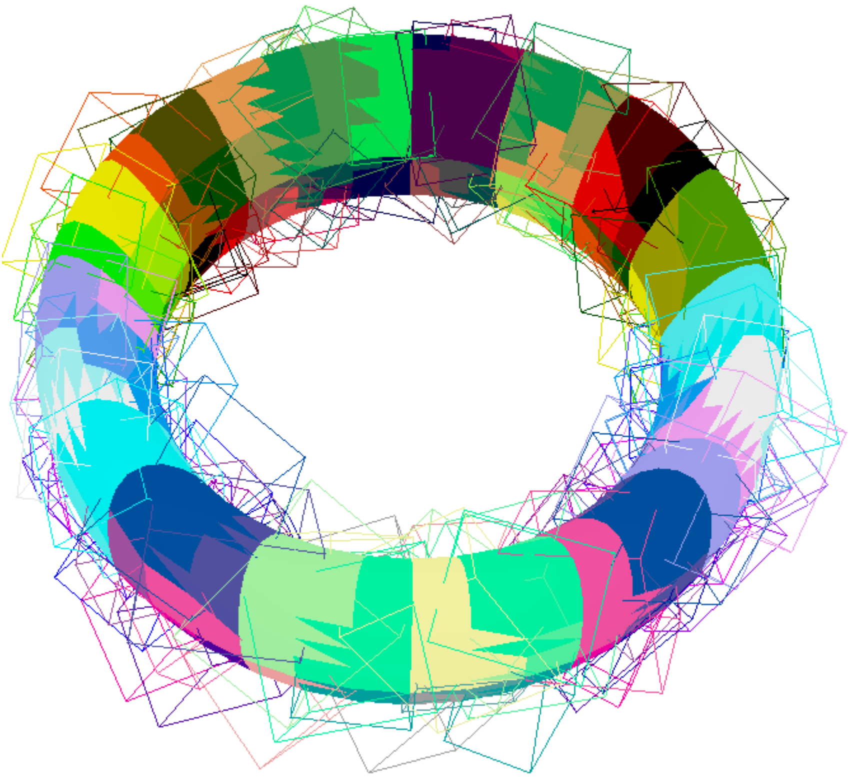
- Minimal number of triangles (*N*) on each leaf, because checking collisions between two leaves is a $O(N^2)$ algorithm
- The tree should be as balanced as possible in order to keep the collision detection of $O(N\log_2(N))$

Current methods for building OBB trees are statistical and their results tend to depend on concentrations of vertexes of the object

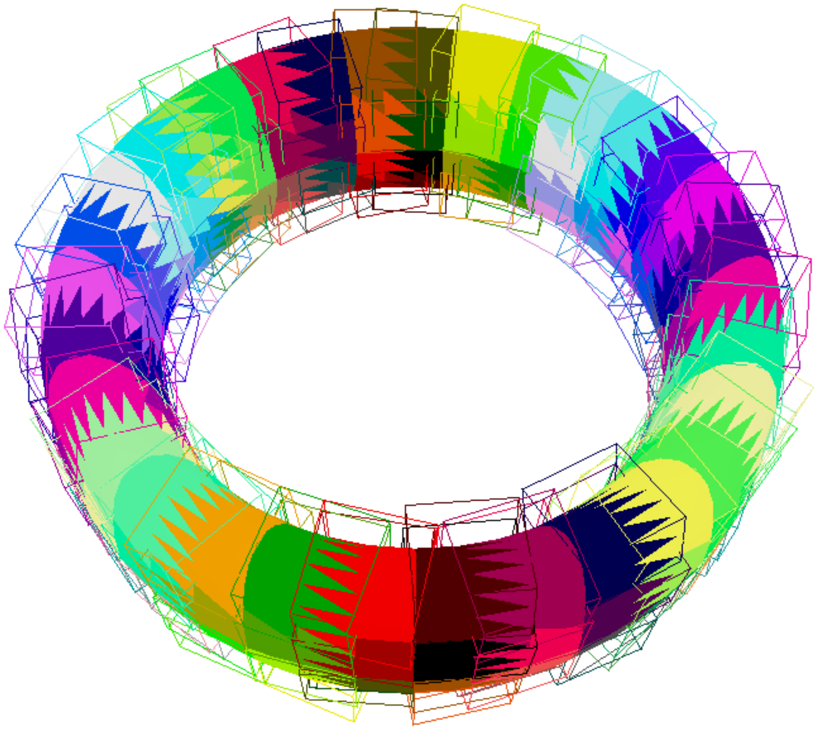
Genetic Algorithms



Approach



Covariance method



Our method

Chromosomal Representation

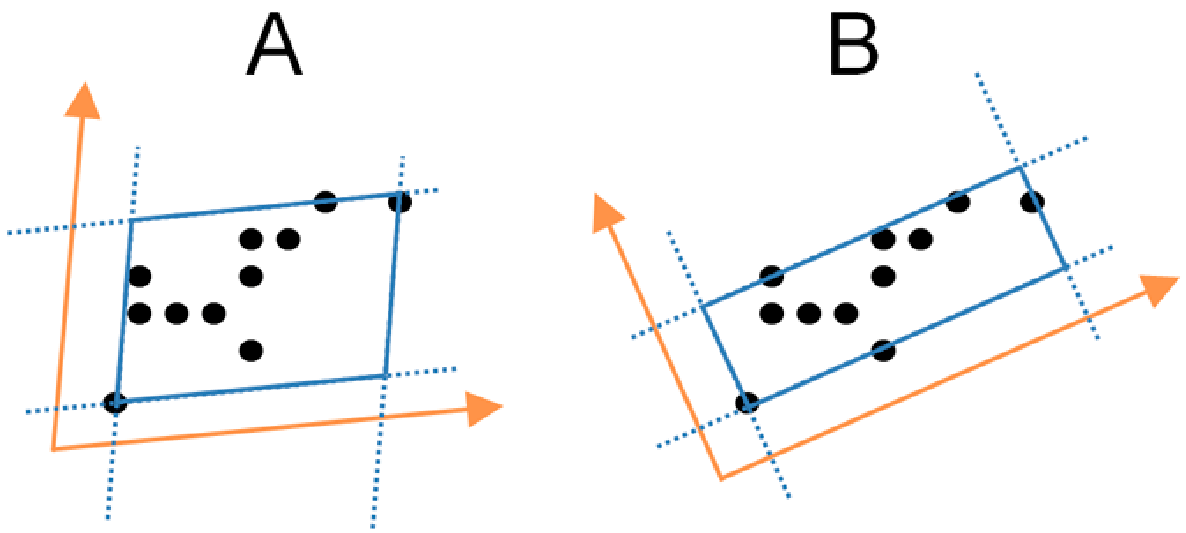
- OBBs can be represented specifying its orientation
- Given an orientation the minimal bounding box that encloses the object can be easily computed
- We use quaternions to represent orientations as they are compact and have important mathematical properties
- A quaternion is a tuple $q = (x, y, z, w)$ which represents a rotation around an arbitrary axis

Chromosomal Representation

- Each component of the quaternion needs to be represented as a floating point
- We use a 16 bit fixed representation for each component of the quaternion
- Range for *x*, *y* and *z* is $[-1, 1]$, with a precision of $2/2^{16} \approx 3.05 \times 10^{-5}$
- Range for *w* is $[0, 360]$, with a precision of $360/2^{16} \approx 5.49 \times 10^{-3}$

Chromosomal Representation

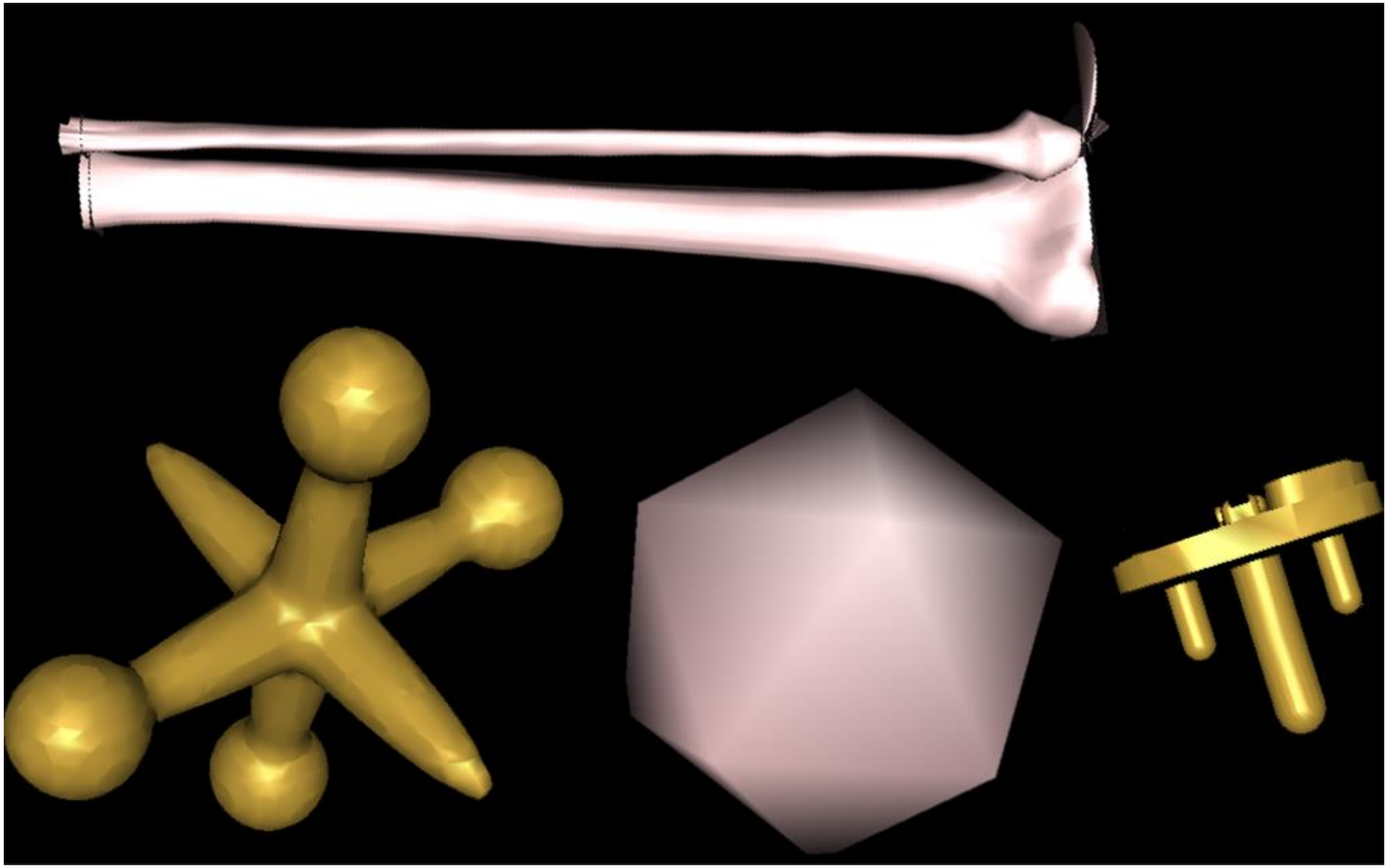
The quaternion is used to rotate the basis axes (1, 0, 0), (0, 1, 0) and (0, 0, 1), obtaining a new set of axis which will be used to build the OBB



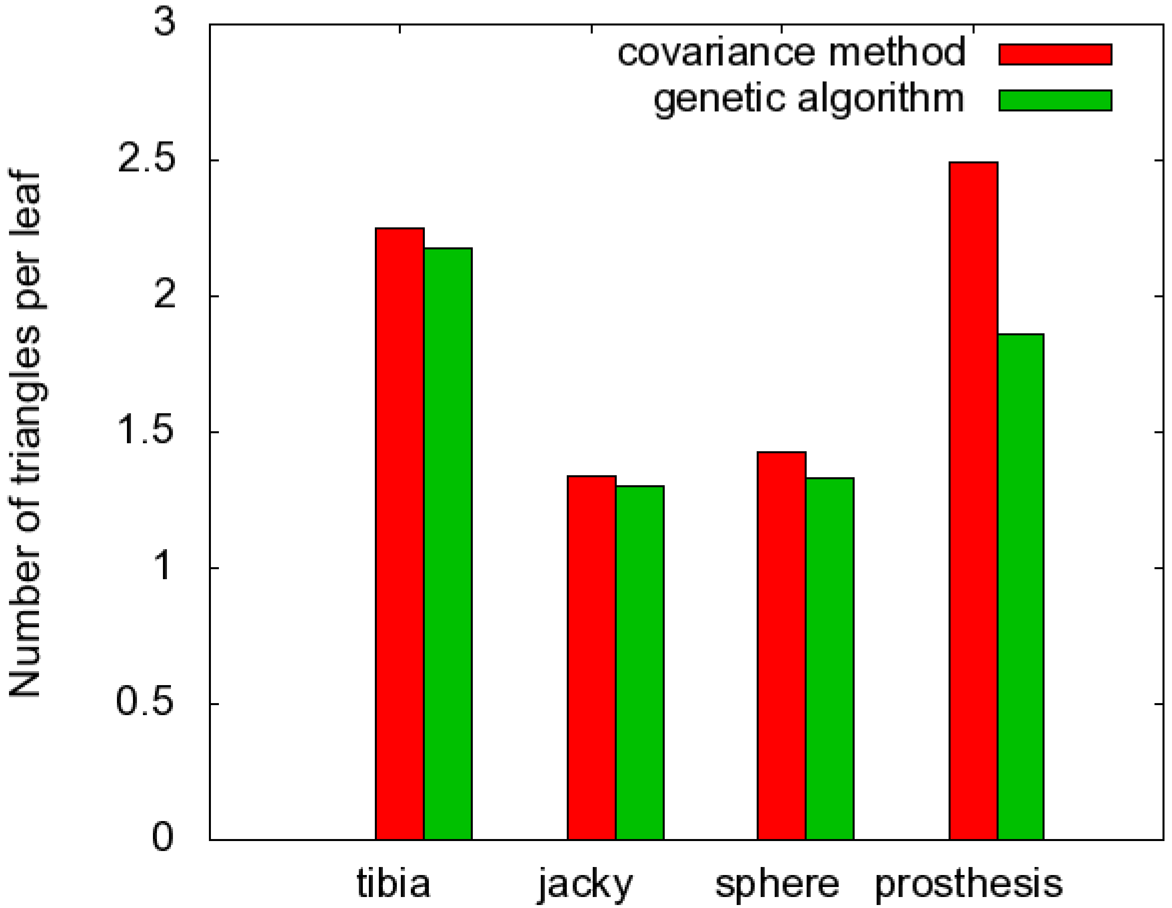
Genetic Algorithm Parameters

Generation Gap	20% of population is replaced
Fitness function	1/V, V is volume of the OBB
Selection method	Roulette wheel
Cross over operator	Single point with p=0.8
Mutation	Probability p=0.03
Generations	150

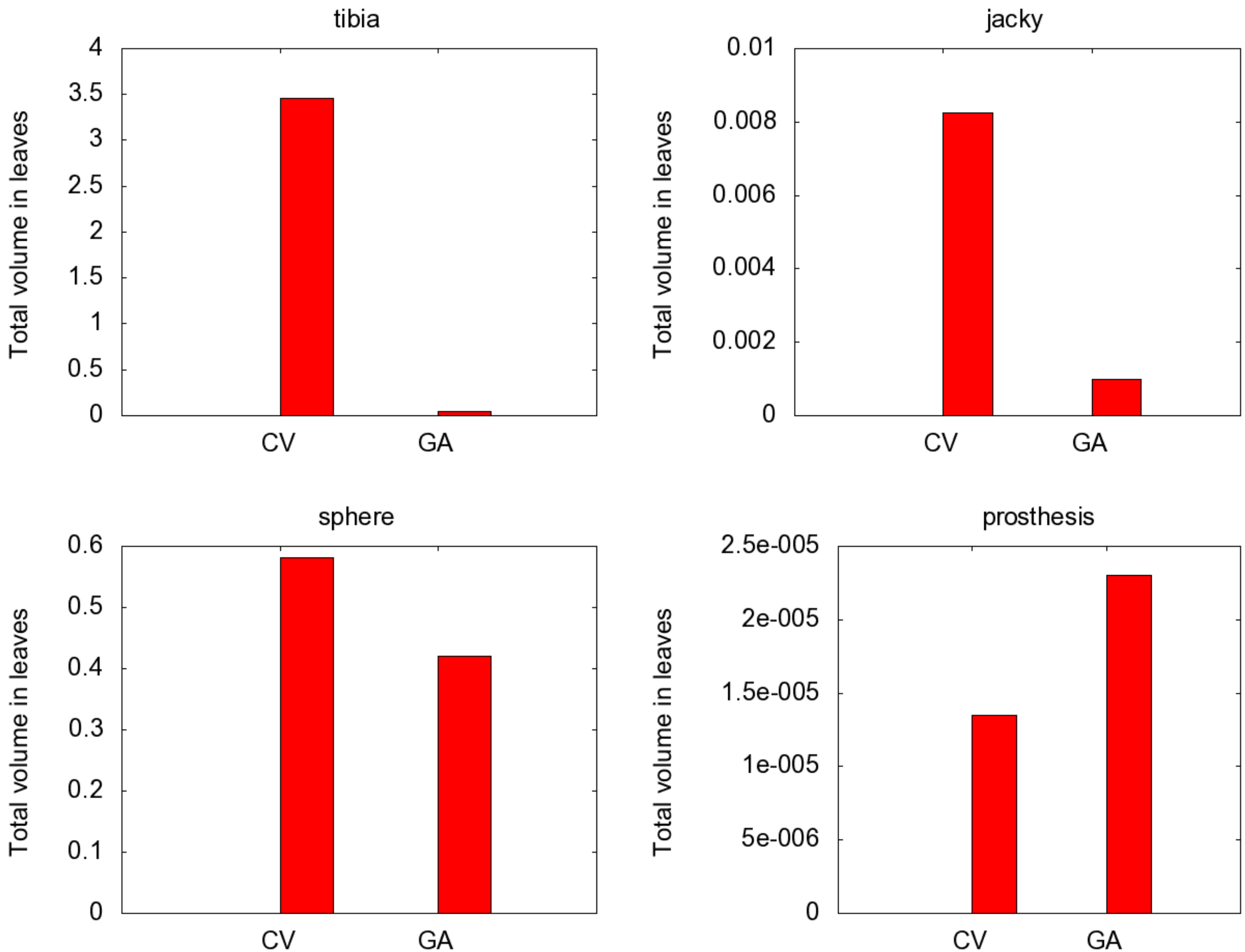
Test and Results



Test and Results



Test and Results



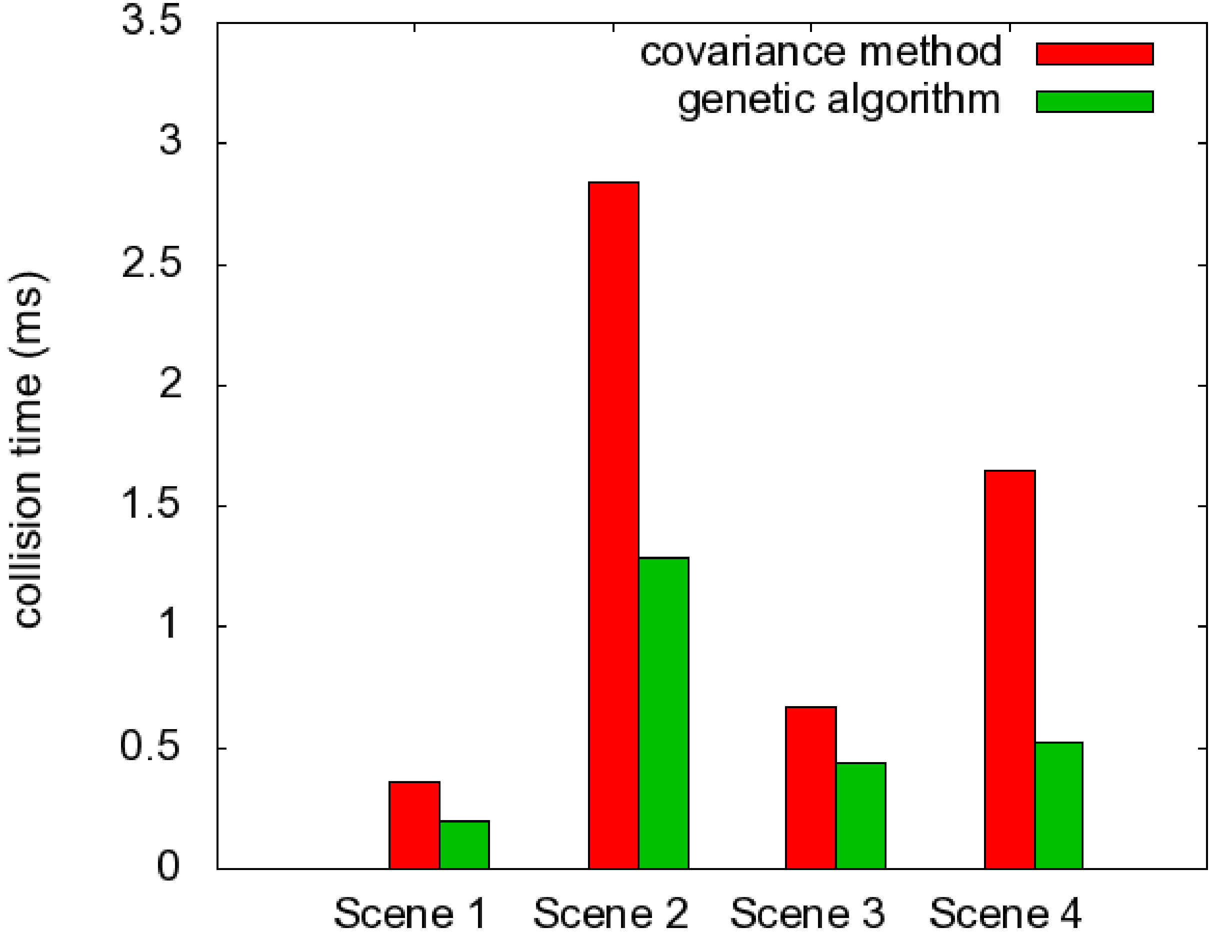
Centro de Computación Gráfica
Ciencias - U.C.V

Test and Results

Four scenes were defined for collision detection timing

Scene	First Object	Second Object	T-T pairs
1	jacky	sphere	132,080
2	jacky	tibia	149,659,848
3	prosthesis	tibia	223,764,588
4	tibia	tibia	513,566,244

Test and Results



Conclusions

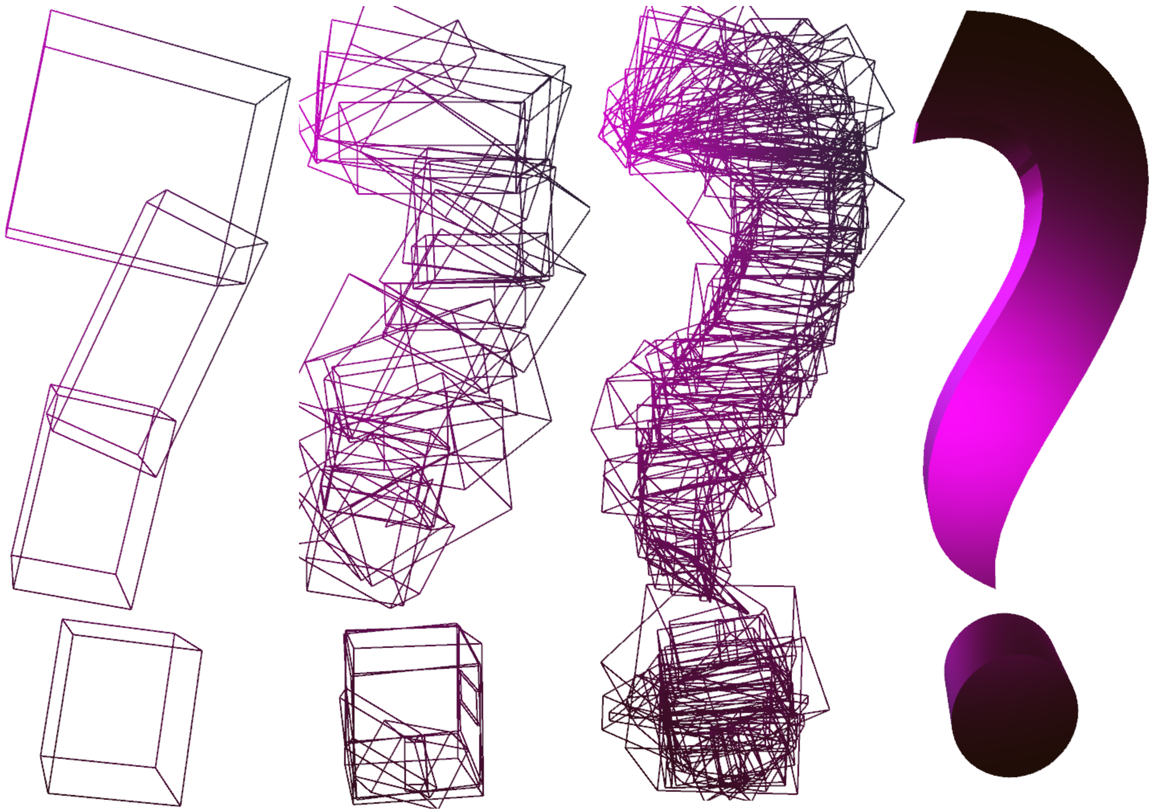
- The proposed GA for generating oriented bounding boxes achieves better results than the covariance method
- The volume of the resulting OBBs using GA is in average lower than the volume of the OBB created with the covariance method
- Time required for detecting collisions is between 40 and 80% lower
- The GA takes more time to build the OBB tree than the covariance method, but it has to be built only one time

Future works

- Recalculate the OBBs when the geometry of the model is slightly modified, so the GA method can be used on dynamic scenes
- Allow the GA to choose the axis along which each OBB will be divided
- Use multithreading in both the genetic algorithm and the collision detection algorithm in order to reduce the required time for these processes

Centro de Computación Gráfica
Ciencias - U.C.V

Questions



http://ccg.ciens.ucv.ve

{esmitt.ramirez,hector.navarro,rhadames.carmona}@ciens.ucv.ve