

IMPROVEMENT OF CONTROL AND SUPERVISION OF WEB CONNECTED MOBILE ROBOTS USING PPU COMPUTATION

Submitted 27th June 2011; accepted 16th September 2011

Janusz Będkowski, Andrzej Mastowski

Abstract:

The paper concerns the research related to the improvement of control and supervision of web connected mobile robots using Physic Processing Unit (PPU). PPU computations taken into the consideration include rigid body dynamics, collision detection and raycasting. The result is improved by Human Machine Interface that allows performing semantic simulation during multi robot task execution. Semantic simulation engine provides tools to implement the mobile robot simulation, which is based on real data delivered by robot's observations in INDOOR environment. The supervision of real objects such as robots is performed by association with its virtual representation in the simulation, therefore events such as object intersection, robot orientation - pitch and roll are able to be monitored. The simulation can be integrated with real part of the system with an assumption of robust localization of real entities, therefore Augmented Reality capabilities are available.

Keywords: *semantic mapping, Human Machine Interface, mobile robot*

1. Introduction

The main problem undertaken in the paper is improved control and supervision of web connected mobile robots [2], for example, inspection intervention robot system, with an application of Physic Processing Unit (PPU). The main applications of multi-robot inspection intervention system are actions in a disaster area, covering all the consequences of fire, chemical hazards, and for example the effect of terrorist attack. The environment of the system forces short time of inspection, and determines basic goals for the system. This provides clearly defined working conditions, the criteria for checking correctness of control and supervision algorithms.

Many studies have shown extensive technical development in the area of mobile robotics. There have been many solutions [8] for technical issues related to unique mechatronics designs of mobile robots. Many new robots have high mobility [11] in difficult terrain. In addition, number of robots equipped with sophisticated sensors [12] increases, which enhances the effectiveness for example search and detection of victims [13], [14].

The paper concerns semantic mapping related to robot ontology [16] that represents a neutral knowledge representation capturing relevant information about robots, their capabilities and environment. It can assist in the development and testing effective technologies for sensing, mobility, navigation, planning integration and opera-

tor interaction. In general, ontologies make all pertinent knowledge about a domain [15] explicit and are represented in a computer-interpretable fashion that allows software to reason over that knowledge to infer with additional information. Applied ontology in semantic simulation engine [1] allows automatic generation of virtual environments based on robot's observations and to perform semantic simulation based on a PPU capabilities.

Physics engines dedicated for video games typically contain two core components, a collision detection/collision system, and the dynamics simulation component responsible for solving the forces affecting the simulated objects. Modern physics engines may also contain fluid simulations, animation control systems and asset integration tools. There are three major paradigms for the physical simulation of solids:

- Penalty methods, where interactions are commonly modeled as mass-spring systems. This type of engine is popular for deformable, or soft-body physics.
- Constraint based methods, where constraint equations are solved that estimate physical laws.
- Impulse based methods, where impulses are applied to object interactions.

Finally, hybrid methods are possible that combine aspects of the above paradigms.

An early academic PPU research project [3], [4] named SPARTA (Simulation of Physics on A Real-Time Architecture) was carried out at Penn State [5] and University of Georgia. This was a simple FPGA based PPU that was limited to two dimensions. First commercial available dedicated PPU PhysX from Ageia was appeared in February 2006 [6]. The unit was most effective in accelerating particle systems, with only a small performance improvement measured for rigid body physics. PhysX is a proprietary realtime physics engine middleware SDK acquired by Ageia [7]. Computer programs such as modern video games supporting hardware acceleration by PhysX can be accelerated by either a PhysX PPU or a CUDA-enabled GeForce GPU. Thus offloading physics calculations from the CPU allow it to perform other tasks instead – resulting in a smoother gaming experience and additional visual effects. An analysis published in [10] shows that most of the code used in PhysX applications is based on x87 instructions without any multi-threading optimization. This could cause significant performance drops when running PhysX code on the CPU. It was suggested that a PhysX rewritten using SSE instructions may substantially lessen the performance discrepancy between CPU PhysX and GPU PhysX.

The paper is organized as follows: section 2 describes multi robotic system structure. Section 3 is related to

PPU computation applied for rigid body simulation/collision detection, raycasting and triggers. Section 4 describes proposed semantic simulation engine with an implementation of semantic Human Machine Interface (main improvement). Conclusions are noted in section 5.

2. Multi robotic system structure

The main object of research is a web connected mobile robot system, for example, the inspection – intervention system consisting of a mobile base station, a group of autonomous robots and remote-controlled robot, equipped with a manipulator with n degrees of freedom. Figure 1 shows structure of such a system.

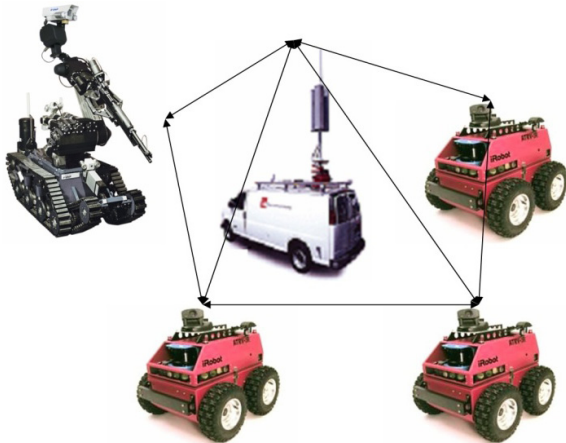


Fig. 1. Inspection – intervention system

The main tasks of the system is an inspection and intervention of hazardous environment for human activities. More details concerning base station, mobile robots and supervision and control can be found in [2].

3. PPU computation

Physic Processing Unit (PPU) is a dedicated microprocessor for physic calculations. With central processing unit (CPU) and general-purpose computing on graphics processing units (GPGPU) PPU builds modern computers capable to execute sophisticated computational tasks in parallel. Examples of calculations involving a PPU include rigid body dynamics, soft body dynamics, collision detection, fluid dynamics, hair and clothing simulation, finite element analysis, and fracturing of objects. The main goal for PPU is to decrease CPU computation. In this paper we are focused on NVIDIA Ageia PhysX chip that is a complete one designed, marketed and supported PPU available with GeForce graphic cards. PhysX is an open standard and gives an opportunity to develop advanced software for mobile robotics for example Microsoft Robotics Developer Studio (MRDS) [9].

3.1. Actors – basic elements of the simulation

An actor typically contains shapes sometimes one, sometimes more, sometimes none. The shapes represent the physical space that actor occupies in the simulation, and they are the core of collision detection. It is possible to create an actor without shapes, which will then behave as a “ghost” and collides with nothing (though it may still be jointed to other actors and affected by gravity and other applied forces or torques). In semantic simulation all semantic entities are related to actors. Geometric at-

tributes correspond to a shape of actor, for example wall will be simulated using rectangular prism. Robot is represented in the simulation by set of actors connected *via* joints. Each actor is associated with a shape – rectangular prism (box shape). Relationships between semantic entities are modeled using joints, for example solid wall orthogonal to floor corresponds to rectangular prism connected to a plane *via* fixed joint.

3.2. Continuous collision detection

It is useful for fast moving objects. With traditional collision detection, fast moving objects pass through other objects during a single time step. This effect is known as tunneling [17]. To deal with this problem, a technique known as Continuous Collision Detection (CCD) is necessary. Instead of testing for collision at discrete points, it tests an extruded volume which represents the object’s motion during the whole time step. If a collision is detected, the time of impact can be computed and the object’s motion constrained appropriately. The example is shown in Figure 2:

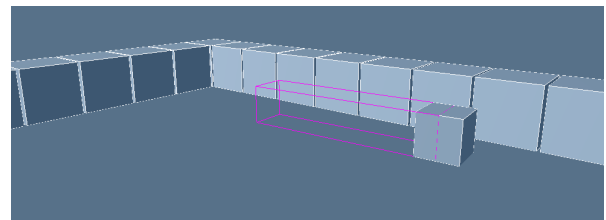


Fig. 2. An example of Continuous Collision Detection (CCD)

3.3. Raycasting

Raycasting is basically a collision detection using rays. However, instead of attaching the line segment to an actor like other shapes, it gets cast into the scene at the user’s request, stabbing one or more shapes in the process. It has many uses, for example picking up objects in the Human Machine Interface to assign new goal for robot in 3D environment. In our robotic application raycasting is used also for laser system measurement simulation (Figure 3).

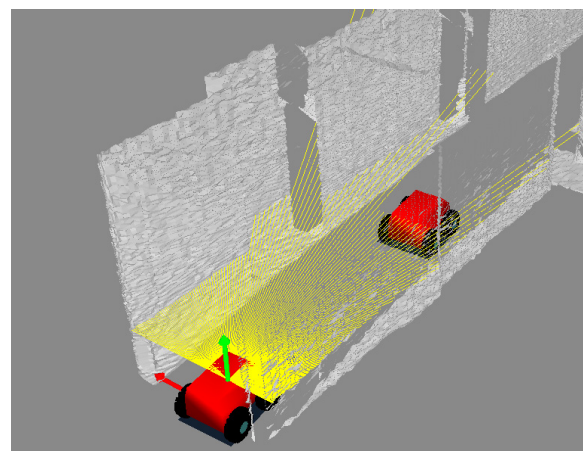


Fig. 3. Raycasting example – laser measurement system simulation

3.4. Triggers

A trigger is a shape that permits other shapes to pass through it. Each shape passing through it can create an

event for the user when it enters, leaves, or simply stays inside the shape. Triggers are used to implement the monitoring of robot's position.

4. Semantic simulation engine

The concept of semantic simulation engine applied in mobile robot system is a new idea, and its strength lies on the semantic map integration with mobile robot simulator. Semantic simulation engine is composed of data registration modules, semantic entities identification (data segmentation) modules and semantic simulation module. It provides software tools for the implementation of the mobile robot simulation based on real data delivered by robot and processed on-line using parallel computation. Semantic entities identification modules can classify door, walls, door, ceiling, stairs in INDOOR environment. Data can be delivered by robot's observation based on modern sensors such as laser measurement system 3D and RGB-D cameras. Real objects are associated with virtual entities of simulated environment.

Semantic entities identification module uses semantic net shown in Figure 4. From computation point of view image processing methods are applied for automatic semantic elements identification based on images derived from projected 3D cloud of points onto OXY plane. An example of automatic generated semantic entities such as walls, door and stairs is shown in Figure 5. More details concerning semantic prerequisites generation can be found in [1].

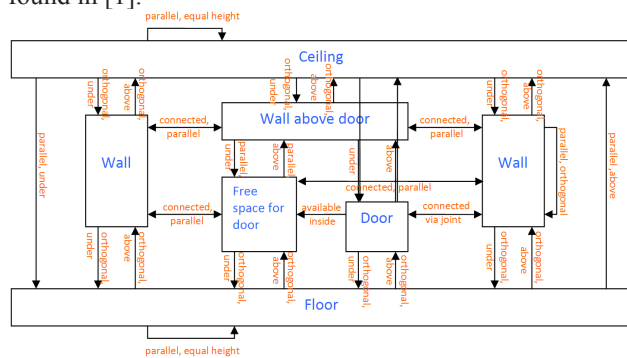


Fig. 4. Semantic net

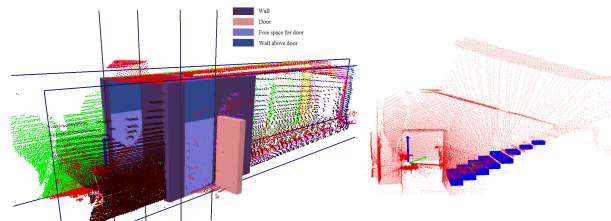


Fig. 5. Semantic entities generated from 3D cloud of points

4.1. Semantic simulation

Accurate positioning of mobile robots, accurate encoders of inspection robot arm and satisfying tracking system are needed to update virtual entities position during real robot task execution. The semantic simulation is composed of: semantic map nodes(entities) $L_{sm} = \{\text{Wall, Wall above door, Floor, Ceiling, Door, Free space for door, Stairs...}\}$, (It is important to noticed that the L_{sm} set

can be extended by another objects, what is dependent on robust and accurate 3D scene analysis), robot simulator nodes (entities) $L_{rs} = \{\text{robot, rigid body object, soft body object...}\}$, semantic map relationships between the entities $R_{sm} = \{\text{parallel, orthogonal, above, under, equal height, available inside, connected via joint...}\}$, robot simulator relationships between the entities $R_{rs} = \{\text{connected via joint, position...}\}$, semantic map events E_{sm} , that are the same as a robot simulator events:

$E_{sm} = E_{rs} = \{\text{movement, collision between two entities started, collision between two entities stopped, collision between two entities continued, broken joint...}\}$. The entities from semantic map correspond to actors in PhysX. L_{sm} is transformed into L_{rs} based on spatial model generated based on registered 3D scans i.e. walls, doors and stairs correspond to actors with BOX shapes. R_{sm} are transformed into R_{rs} with remark that doors are connected to walls via revolute joints. All entities/relations R_{sm} has the same initial location in R_{rs} , the location of each actor/entity may change during simulation. The transformation from E_{sm} to E_{rs} effects that events related to entities from semantic map correspond to the events related to actors representing proper entities. Following events can be noticed during simulation: robot can touch each entity, open/close the door, climb the stairs, enter empty space of the door, damage itself (broken joint between actors in robot arm), brake the joint that connects door to the wall.

It is noteworthy to mention that all robot simulator's semantic events are useful for semantic HMI, where computer can monitor simulation events. From robot control point of view semantic simulation is used also for robot path planning using "empty spaces" (see Figure 6). "Empty spaces" compose graph, therefore it is possible to use classic path planning methods.

4.2. Semantic HMI

To demonstrate proposed approach Figure 6 shows basic robot task – going forward via corridor.

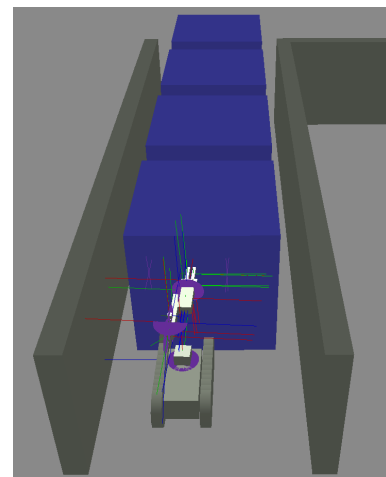


Fig. 6. Semantic HMI example. Robot, walls (green rectangular prisms), empty spaces (blue cubes)

Semantic HMI visualizes robot position represented by rectangular shapes, environment objects – in this case walls and empty spaces. Semantic HMI supervises intersection of robot shapes with environment shapes to avoid potential collisions. HMI monitors robot's position using triggers – empty spaces. Defined goal from operator console's point of view is shown in Figure 7 (where rayca-

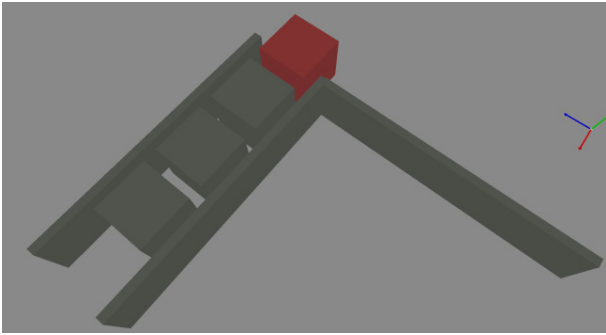


Fig. 7. Defining goal (red cube) in operator console

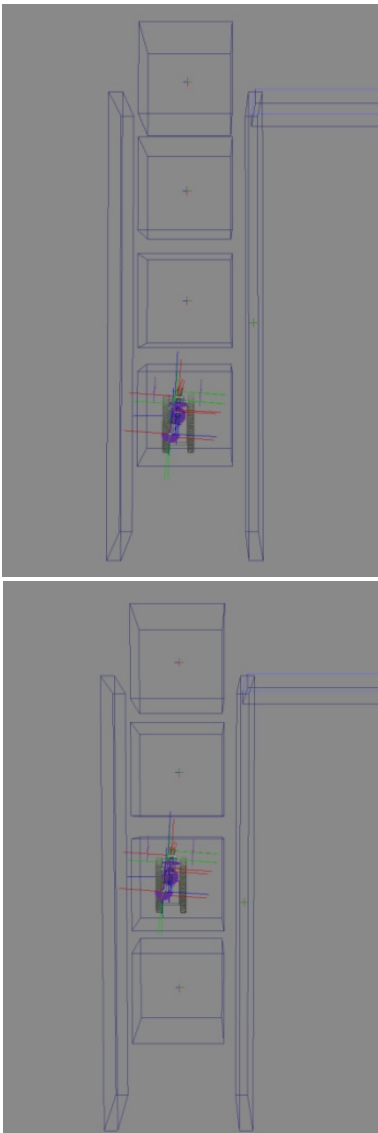


Fig. 8. Monitoring robot position

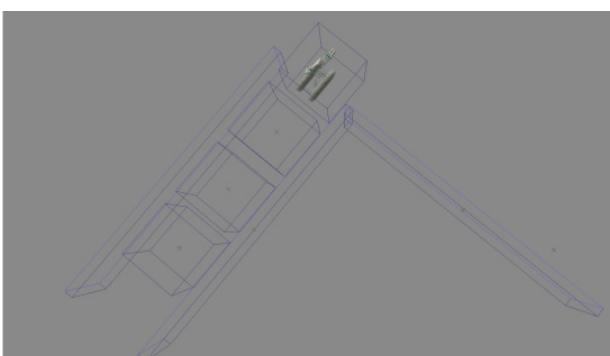


Fig. 9. Achieving goal by a robot

sting method was used to pick up cubic shape). Figures 8 shows monitoring of the robot's position, where robot is visiting new empty spaces and generate semantic events *via* triggers. Achieved goal by robot is shown in Figure 9, it is also related with a semantic event generated automatically by a trigger. The main difference comparing to a classic HMI proposed semantic HMI generates linguistic description of robot status, for example:

Robot collides with wall.

Robot is inside empty space.

Robot achieved goal.

5. Conclusion

In this paper an improvement of supervision and control of web connected mobile robots using Physic Processing Unit (PPU) computation is shown. The approach is related with semantic simulation engine that allows building robot simulation based on robot's observations (laser measurement system 3D, RGB-D cameras). Proposed ontological approach was used for creating semantic Human Machine Interface that is more ergonomic that potentially will improve human – robot interactions. The main achievement is the implementation of a semantic HMI that can be integrated with modern mobile robotic systems. HMI can integrate classic methods for path planning with sophisticated semantic map building, where decisions are made in high conceptual level. This research is showing an idea of replacing the geometrical description of the robot environment by ontology – based semantic representation, where objects are described by its attributes and relations, therefore the implementation is closer to a human natural understanding of the world. We believe that future systems will be more ergonomic by usage presented software techniques.

ACKNOWLEDGEMENTS

This work is performed during postdoctoral scholarship (CAS/19/POKL 15.05.2011-15.11.2011) in Royal Military Academy, Unmanned Ground Vehicle Center, Brussels, Belgium, funded by Center for Advanced Studies, Warsaw University of Technology (project: "Priority IV of Human Capital Operational Programme from European Social Fund").

AUTHORS

Janusz Będkowski* – Industrial Research Institute for Automation and Measurements PIAP, Al. Jerozolimskie 202, 02-486 Warsaw, Poland; Institute of Automation and Robotics, Warsaw University of Technology, św. A. Boboli 8, 02-525 Warsaw, Poland, januszbedkowski@gmail.com

Andrzej Masłowski – Industrial Research Institute for Automation and Measurements Al. Jerozolimskie 202, 02-486 Warsaw, Poland; Institute of Automation and Robotics, Warsaw University of Technology, św. A. Boboli 8, 02-525 Warsaw, Poland, amaslawski@piap.pl a.maslawski@mchtr.pw.edu.pl

*Corresponding author

References:

- [1] J. Będkowski, A. Masłowski, “Semantic simulation engine for mobile robotic applications”, *Pomiary, Automatyka, Robotyka*, 2/2011, Automation 2011, 6th-8th April, Warsaw, pp. 333-343,
- [2] J. Bedkowski, A. Masłowski, “Methodology of control and supervision of web connected mobile robots with CUDA technology application”, *Journal of Automation, Mobile Robotics and Intelligent Systems*, vol. 5, no. 2, 2011, pp. 3-11.
- [3] S. Yardi, B. Bishop, T. Kelliher, “HELLAS: A Specialized Architecture for Interactive Deformable Object Modeling”. In: *ACM Southeast Conference*, Melbourne, FL, 10th-12th March, 2006, pp. 56–61.
- [4] B. Bishop, T. Kelliher, “Specialized Hardware for Deformable Object Modeling”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 11, Nov. 2003, pp. 1074-1079.
- [5] “SPARTA Homepage”. Cse.psu.edu. <http://www.cse.psu.edu/~mdl/sparta/>. Retrieved 2010-08-16.
- [6] “Exclusive: ASUS Debuts AGEIA PhysX Hardware”. AnandTech. <http://www.anandtech.com/show/2001/4>. Retrieved 2010-08-16.
- [7] NVIDIA Corporation (2008-02-13). *NVIDIA completes Acquisition of AGEIA Technologies*. Press release. http://www.nvidia.com/object/io_1202895129984.html. Retrieved 2009-03-18.
- [8] J. Casper, R. R. Murphy, “Human-robot interactions during the robot-assisted urban search and rescue response at the World Trade Center”. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, vol. 33, 2003, pp. 367-385.
- [9] <http://msdn.microsoft.com/pl-pl/robotics/>
- [10] Real World Technologies – PhysX87: Software Deficiency
- [11] B. Shah, H. Choset, *Survey on urban search and rescue robotics*. Technical report, Carnegie Mellon University, Pittsburg, Pennsylvania, 2003.
- [12] M. Strand, R. Dillmann, “3D-Environment Modeling using an autonomous Robot”. In: *IARP-Workshop on Environmental Maintenance & Protection*, CD-ROM, 2008.
- [13] G. De Cubber, G. Marton, “Human Victim Detection”. In: *International Workshop on Robotics for risky interventions and Environmental Surveillance, RISE'2009*, Brussels, 12th-14th January 2009, CD-ROM.
- [14] H. Sugiyama, T. Tsujioka, M. Murata, “Victim detection system for urban search and rescue based on active network operation”. In: *idem, Design and application of hybrid intelligent systems*, IOS Press, Amsterdam, 2003, pp. 1104-1113.
- [15] F. Harmelen, D. McGuinness, “OWL Web Ontology Language Overview,” W3C web site: <http://www.w3.org/TR/2004/REC-owl-features-20040210/>, 2004.
- [16] C. Schlenoff, E. Messina, “A Robot Ontology for Urban Search and Rescue”. In: *Proceedings of the 2005 CIKM Conference: Workshop on Research in Knowledge Representation for Autonomous Systems*, 31st October – 5th November, 2005, Bremen, Germany.
- [17] NVIDIA PhysX SDK 2.8 – Introduction.