Sensor Modeling for a Walking Robot Simulation

L. France[1,2], A. Girault[1], J-D. Gascuel[2], B. Espiau[1]
[1] INRIA, Grenoble, FRANCE
[2] *i*MAGIS, GRAVIR/IMAG, Grenoble, FRANCE

Abstract

This paper proposes models of short-range sensors. The target application is the simulation of the environment perception for an autonomous biped robot that evolves in an unknown surroundings. These proximity sensors can be of different types, emulating real sensors such as laser range finders, ultrasonic sensors or reflectance sensors. These sensors will be used to provide real-time local information about the environment to the robot in a simulation. Strategies to avoid and circumvent an obstacle can then be easily tested.

1 Introduction

In the context of control of mobile legged robots in unknown environments, the focus in robotics research is to make the robots react correctly to their environment. However, the development of a robot is a very long process. Before experimenting with the real robot, robotics researchers prefer to use a simulation system in order to have a first idea of high level controls that have to be developed. Therefore, there is a crucial need for the simulation of interaction between the "virtual" robot and its future environment.

In computer graphics, a solution is to provide to the animated actor, what we call robot, a direct access to the exact knowledge of each object in the environment. This approach is convenient for small and simple environments, but becomes impractical for large or complex environments. Moreover, this solution rarely fits reality, and thus is not suitable to our robotics study. We prefer an approach that deals with the local environment of the robot, to provide pertinent information for direct robot tasks, such as collision avoidance. In robotics, proximity sensors are used to perform this kind of tasks. We thus want to provide the robot with the best perception of its environment, so that it evolves safely in it, i.e., without colliding with obstacles and falling. Our aim is then to model such real short-range sensors. The sensors are affixed on particular places on the walking biped robot.

The remainder of the paper is organized as follows. In Section 2 we present an overview of related work on sensor simulation. In Section 3 we present what are the physical characteristics of proximity sensors. In Section 4 we explain computer graphics techniques used to emulate these sensors. We introduce our models of proximity sensors in Section 5 and their uses in the particular case of a biped robot in Section 6. Finally, in Section 7, we conclude with some perspectives.

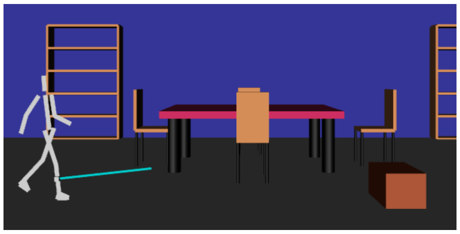

Figure 1: *The biped robot in a virtual environment, with a sensor placed on its leg.*

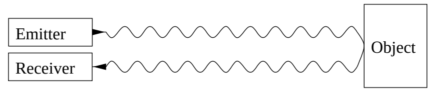

Figure 2: *Sensor devices are mainly based on wave propagation.*

2 Related Work

In computer graphics, environment sensing is rarely considered. Animators take into account the global knowledge about the scene. They use the underlying representation of the environment to make actors know where objects are in the scene. For instance, an actor knows when and where he has to walk over a step or to bend his head under a beam [1] without having to detect such an obstacle.

Concerning our aim, the closest related work are based on visual perception simulation. "Synthetic vision" is provided to virtual actors who utilize vision techniques to extract useful informations from an image rendered from their view point. The advantages of synthetic vision are the locality of information and the "weaker" dependence on the underlying representation of the environment. The actor has a virtual vision sensor which renders the scene using a graphics camera mounted at the position and orientation of the sensor, simulating a real sensor device. The resulting RGB image is extracted from the frame buffer and allows the recovering of key features of its immediate environment, which are then used to guide the movement of the actor. Blumberg [2] uses the resulting image to generate a potential field from the actor's perspective. A gradient field is calculated and used to derive a bearing away from areas of high potential. Thalmann *et al.* [3] apply synthetic vision to autonomous virtual actors. These actors only know their position and vision image, and they are able to perform tasks such as vision based navigation and collision avoidance.

However, such techniques raise the following problem: animation is hardly done in real-time. Indeed, vision is known to be a complex and time-consuming process of recognition of environment captured by camera. Thalmann *et al.* [4] treat this problem by providing extra information to actors to make them more intelligent and also faster. Actors are anew dependent from the internal representation of the virtual environment. In this work, we prefer to focus on simulating existing physical devices as it will provide a much more adequate information for simulation.

3 Proximity Sensors

In robotics, informations on perception are provided to robots from cameras and sensors. A camera is used for general tasks, like navigation, as it provides global informations. The problem with camera is the complexity of the resulted image of the world it renders. It is very difficult and time consuming to extract informations from captured images. Other sensors are then used to facilitate the camera task. These sensors deal with some details that the camera does not have to care any more. For instance, proximity sensors give simple and fast information on short-range distance to objects, useful to avoid collision with obstacles. In this paper, we propose to model such real proximity sensors for a mobile robot with computer graphics techniques. In the next section, we detail principles and characteristics of existing proximity sensors.

3.1 Sensor Physics

A proximity sensor is a device that detects objects locally, using a particular physical interaction with the environment. Several modes of interaction with the environment exist, depending on the involved physical principles. For instance, lighting or acoustical wave propagation in the case of proximetry, telemetry and local vision [5]. A sensor consists of an emitter that emits a wave and a receiver that receives the reflected wave.

3.2 Detection Principle

We are interested in proximity sensors based on the physical principle of time-of-flight detection, namely ultrasonic sensors and laser range finders, and by reflectance measure, namely infra-red leds or laser sensors. All these types of sensors are based on wave propagation: A wave is emitted and propagated in the environment. If the wave encounters an obstacle, it is reflected in a direction that depends on the position and surface properties of the obstacle, and on the direction and the properties of the wave (see 3.3). The reflected wave is then received and analyzed by the sensor (Figure 2).

The wave intensity decreases according to the traveled distance. Moreover, if the reflected wave has low intensity, the receiver is not sensitive to this wave. Therefore, it does not detect anything too far away, as the wave gets attenuated with the distance. A general approximation established in robotics states that no multiple reflections but only one reflection is considered. This approximation is justified by the fact that detection distance is short, and wave energy decreases fast with distance travelled.

3.3 Wave Reflection

Obstacles reflect the wave because their surfaces act as semi-mirrors that re-emit all or part of the received energy, according to surface properties. In general, two physical models of surfaces are used [6]:

- the *perfect diffuse surface*: this type of surface re-emits the entire incident energy in all directions, independently of the wave direction (Figure 3 (a)).

- the *perfect specular surface* (perfect mirror): this type of surface re-emits all the incident energy in the mirror direction of the wave direction, with respect to the normal of the surface (Figure 3 (b)).

These two types of surface characterize perfect models, but in reality, surfaces are a combination of both types, in various proportions, depending on the granularity of the surface (Figure 3 (c)).

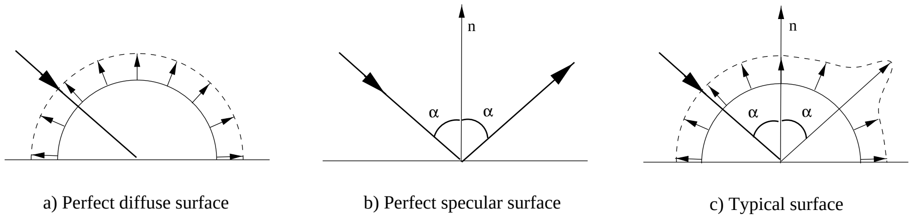

Figure 3: *Basic types of surfaces.*

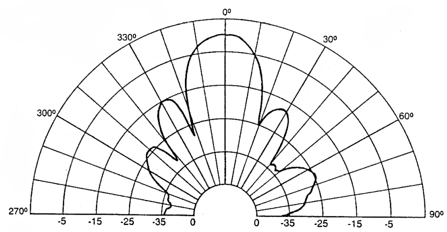

Figure 4: *Typical beam pattern for a 50 kHz ultrasonic sensor (RoboSoft™).*

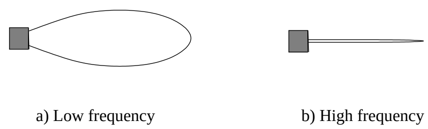

Figure 5: *Principal emission lobes of ultrasonic sensors at low frequency (a) and high frequency (b).*

3.4 Characteristics of Optical and Acoustical Sensors

In robotics, the most common proximity sensors are ultrasonic sensors, laser range finders and reflectance sensors. We want to model all these types of sensors. For this purpose, we give here characteristics associated to each type of sensor.

Ultrasonic sensors

The emitter sends pulse trains. The time of flight is measured between the emission and the reception of impulse trains. The same device is usually used for both the emitter and the receiver. The switching time makes a "dead zone" represented by the minimal detectable range. Ultrasonic waves can be of two types, either high frequency (100-400 kHz) or low frequency (40-100 kHz). The drawback with ultrasonic sensors is their low directivity (Figure 4). No high accuracy in position measure can be obtained with such sensors. The result is improved by the use of high frequency waves that are more directive than low frequency waves (Figure 5), even if the distance traveled by the wave is smaller. We thus consider ultrasonic sensors of high frequency. Furthermore, because of ultrasonic wave properties, all reflecting surfaces can be considered as specular surfaces. We then make the assumption that ultra-sounds are only received when the incidence angle with the obstacle is in the range $[-15^o, +15^o]$.

Laser range finders

It is impossible to accurately measure the time of flight of a laser, because this time is too small (1 nanosecond for a distance of $30cm$). The wave is modulated instead, and the sensor measures the phase shift between the emitted wave and the received wave. This phase shift is mainly function of the distance between the sensor and the obstacle, but also depends on the surface and the incidence angle. For practical reason, instead of modeling every single detail, like all objects with their respective diffusion characteristics, we assume that light is received when the incidence angle is in the range $[-45^o, +45^o]$, which is a good approximation in practice.

Reflectance sensors

Whereas previous sensors give directly the actual distance, reflectance sensors provide diffused light intensity. This intensity is function of the distance and the incidence angle. We take the following equation to simulate signal behavior:

$$s = \frac{(d_{max} - d)}{d^2} . \cos^2 \alpha . (1 + Noise) \tag{1}$$

with d_{max} the maximum distance, d the detected distance, α the incidence angle, $Noise$ the amount of noise of the sensor. This equation indicates that the signal is inversely proportional to the distance, and tends towards zero when the distance becomes the maximum distance. Beyond this limit, the distance returned by the sensor remains d_{max}, and the signal is thresholded to zero. The signal is also function of the incidence angle because the smaller the incidence angle, the higher the light intensity.

4 Modeling of the Physical Principle of Detection

In this section we present an interaction mode for sensors in the environment. Since studied sensors are based on the principle of wave propagation, we decide to model this wave by a ray and we emulate a ray casting. We compute the distance from the first intersection of the ray with objects. This approximation seems to be convenient in the diffuse wave detection case, as the reflected wave is highly attenuated by each reflection. Nevertheless, it seems to be less convenient in the specular wave detection case, but then multiple reflections would generate wrong distance, much higher than the real one, and would generally not be considered. Once the distance to the obstacle is computed, we make approximations to take into account real sensor characteristics.

4.1 Ray Implementation

We define a ray that simulates the wave. Ray characteristics are: the start position, the ray direction, the minimum threshold (d_{min}), corresponding to the minimum range of the sensor, and the maximum threshold (d_{max}), corresponding to the maximum range of the sensor. To simulate the ray, we use a well-known Computer Graphics technique called *ray tracing*. This method has existed over the past 20 years, and algorithms to compute intersections between a ray and a complex scene in an efficient way have been developped [7]. Using such optimized calculations, we easily find the intersection point between our wave and our scene. Then, according to the threshold, we can output: the distance of the first detected object point, and the incidence angle between the direction of the ray and the normal of the object surface in the detected point.

4.2 Approximation for the Incidence Angle

The signal depends on the incidence angle of the wave with the object surface. Indeed, according to the orientation of the encountered obstacle with respect to the emitted wave direction, the reflected wave will be or not be detected by the sensor. We define a range $[-\alpha^*, \alpha^*]$ for the incidence angle. If the incidence angle is in that range, the sensor will detect the reflected wave. Otherwise, the sensor will not detect anything. Note that this range depends on the kind of waves we use, ultrasonic or laser.

4.3 Sensor Noise

We also take into account the sensor noise that represents the uncertainty on the measure. We approximate this error by a Gaussian law. The noise is then defined by: $Noise = \sigma.G + m$, with σ the standard deviation, m the mean of the Gaussian law, and G a Gaussian variable that can be computed by $G = \sqrt{-2.\ln U_1}.\cos(2\pi.U_2)$, where U_1 and U_2 are two variables of uniform law in $[0, 1]$.

5 Modeling of Sensors

We can simulate laser range finders, ultrasonic sensors and reflectance sensors. We developed an interface in order to add sensors on the robot model. We can visualize what sensors detect by displaying light rays. For each modeled sensor, we give its specific parameters, concerning the signal.

A laser range finder

size:	0.05 m x 0.05 m x 0.05 m
threshold min.:	d_{min} = 0 m
threshold max.:	d_{max} = 2 m
distance:	$d \in [d_{min}, d_{max}]$
incidence angle:	$\alpha \in [-45^o, +45^o]$
noise N:	$\sigma = 0.01, \quad m = 0.$

The result distance is computed by:

if $d \in [d_{min}, d_{max}]$
and $\alpha \in [-45^o, +45^o]$
then $d_r = d.(1 + N)$
else $d_r = d_{max}$

An ultrasonic sensor

size:	0.05 m x 0.05 m x 0.05 m
threshold min.:	d_{min} = 0.1 m
threshold max.:	d_{max} = 1 m
distance:	$d \in [d_{min}, d_{max}]$
incidence angle:	$\alpha \in [-15^o, +15^o]$
noise N:	$\sigma = 0.01, \quad m = 0.$

The result distance is computed by:

if $d \in [d_{min}, d_{max}]$ and
$\alpha \in [-15^o, +15^o]$
then $d_r = d.(1 + N)$
else $d_r = d_{max}$

A reflectance sensor (infra-red led or laser)

size:	0.05 m x 0.05 m x 0.05 m
threshold min.:	d_{min} = 0 m
threshold max.:	d_{max} = 1 m
distance:	$d \in [d_{min}, d_{max}]$
incidence angle:	$\alpha \in [-90^o, +90^o]$
noise N:	$\sigma = 0.01, \quad m = 0.$

The result distance si computed by (see explanation below):

if $d \in [d_{min}, d_{max}]$ and
$\alpha \in [-90^o, +90^o]$
then $s = \frac{(d_{max} - d)}{d^2}.\cos^2\alpha.(1 + N)$
$d_r = \frac{-1+\sqrt{1+4.s.d_{max}}}{2.s}$
else $d_r = d_{max}$

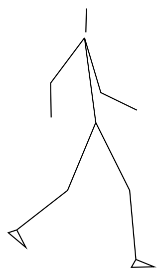

Figure 6: *Model of a biped robot.*

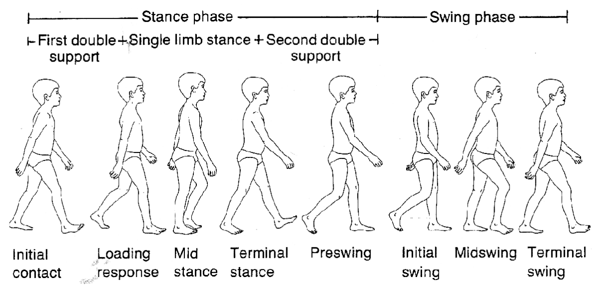

Figure 7: *Gait cycle for a child, analogous for the biped robot (from [8]).*

The measured signal depends here on the distance, the incidence angle and the noise: $s = f(d, \alpha, N)$ (cf. Equation (1)). We can not directly deduce the detected distance. We make the following approximation: $f^{-1}(s, \alpha = 0, N = 0) = \frac{-1 \pm \sqrt{1+4.s.d_{max}}}{2s}$. This equation gives both roots of the second degree polynomial $s.d^2 + d - d_{max}$, computed from Equation (1), where α and $Noise$ are null. The second root is always negative, so we use the only positive root under the given assumptions, to compute the approximated distance.

We place such sensors on the robot and we can detect walls and other vertical obstacles in front of the robot. In the next section, we explain how we can use a biped robot to scan the environment vertically.

6 Application to a Walking Robot

We are working in a project that builds a real biped robot that has to evolve in unknown environments. We thus want to affix sensors on the robot, in order to perceive and reconstruct the scene. In that context, we simulate sensors and develop algorithms for environment reconstruction. We are particularly interested in the study of sensor positioning on a biped robot (Figure 6). Indeed, a major advantage of biped robots is that a sensor device placed on the legs will vertically scan the environment, following the ups and downs of the legs during the walk. This additional movement provides further information. For instance, when a sensor is positioned on the tibia, the sensor scans also the ground while its support leg is in the swing phase (Figure 7). In the case of mobile robots, only vertical obstacles are detected. With the biped robot, we also detect eventual holes in the ground, or how the ground is uneven, or even stairs. In the following sections, we present tools to better appreciate the information given by a sensor.

6.1 Distance Curves

Sensors move along with the robot when they are tethered to them. In consequence, signals from the sensors continuously provide information on the closest obstacles detected. We can draw a curve in time that indicates the variations of the outputs of these

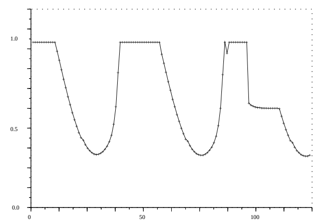

Figure 8: *Example of ideal distance curve.*

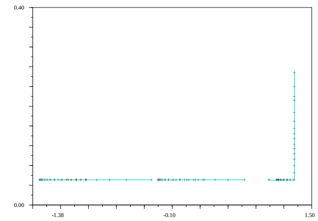

Figure 9: *Example of ideal profile computed from the distance curve of Figure 8.*

sensors, as depicted in Figure 8. The type of curves, that we will call *distance curves*, are hard to interpret directly, as sensors do not follow a straight line. We then need to extract relevant information from these raw data.

6.2 Profile Computation

With distance curves, we get information on the distance to close obstacles at any given time. Even if it is sufficient if we just want to detect potential collisions, we would like to extract from these data the profile of the environment of the robot, in order to adapt or plan the motion of the robot appropriately. We cannot hope for more information than a simple profile, as we only scan the environment vertically. To draw this profile, we start by computing the world position and orientation of the sensor from its relative position and orientation on the robot, the robot geometry, and the robot referential relative to the initial referential. And knowing the detected distance, we can compute the 3D point that has been hit. If the detected distance is the maximum threshold of the sensor, it means that nothing is detected by the sensor and the path is clear. Otherwise we add the 3D point to the profile. We then display the profile of the sensed environment as it gets constructed. Figure 9 shows a profile computed from data provided by an ideal sensor.

6.3 Tests

We made tests with sensors modeled in Section 5 that are positioned on a robot that walks in a dining room environment. The simulation provides about one frame par second on an SGI O2 R5000 workstation. We present results for each case, corresponding to the detection of the ground, and at time $t = 90s$, of a small obstacle.

- In the ideal case: Figure 8 shows the exact detected distances. The sensor detects the ground only for two periods, before detecting the obstacle on the ground. The animation stops before the robot and the obstacle collide. The corresponding profile is represented in Figure 9.
- For the laser range finder: Figure 10 shows the distance detected by the laser range finder modeled in Section 5. When the incidence angle is too large, it is out of range for the sensor. This makes the curve decrease more rapidly than in Figure 8. In Figure 11, less points are represented compared to Figure 9.
- For the ultrasonic sensor: Figure 12 shows that the ultrasonic sensor modeled in Section 5, detects only the vertical obstacle. Indeed, the incidence angle varies

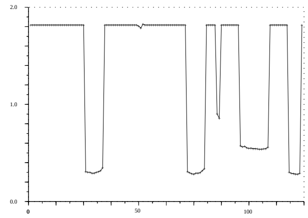

Figure 10: *Distance curve of the laser range finder.*

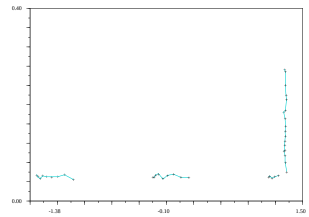

Figure 11: *Profile of the laser range finder, computed from the distance curve of Figure 10.*

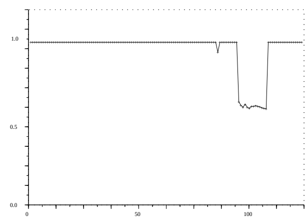

Figure 12: *Distance curve of the ultrasonic sensor.*

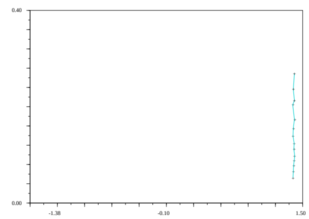

Figure 13: *Profile of the ultrasonic sensor, computed from the distance curve of Figure 12.*

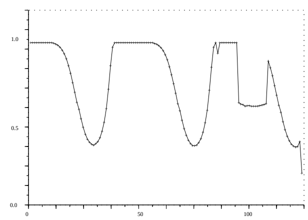

Figure 14: *Distance curve of the reflectance sensor.*

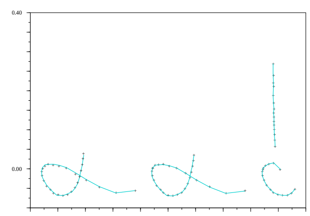

Figure 15: *Profile of the reflectance sensor, computed from the distance curve of Figure 14.*

in the range $[33^o, 78^o]$, out of the ultrasonic sensor range ($[-15^o, +15^o]$). Figure 13 gives the corresponding profile.

- For the reflectance sensor: Figure 14 shows what the reflectance sensor modeled in Section 5 detects. Figure 15 gives the corresponding profile. We can see that this kind of sensor is not perfectly adapted to our purposes.

7 Conclusion and future work

In this paper, we have proposed a model for proximity sensors, emulating laser range finders, ultrasonic sensors and reflectance sensors. We placed these virtual sensors on a biped robot, walking in an unknown environment. The simulated sensors provide information on distance to obstacles, from which we extract the environment profile for path planning. Videos showing some results presented in this paper can be found at the following http address:

http://www.inrialpes.fr/bip/people/france/cas99.html

This method remains simple and fast to establish the environment of the robot. It could help global vision in handling local characteristics, simplifying the task of global vision, that could focus on general characteristics still useful in robotics.

We have started to study the placement of sensors on the robot. Indeed, placing few sensors at some strategic positions can make the collision detection more robust. Another interesting future work, mixing computer graphics and vision, would be to reconstruct the whole environment using all the 3D points gathered, based on the work of [9, 10]. Further developments include a higher level control of the path planning according to the results of this segmentation: a complete simulation environment for biped robots would then be achieved as a test-bed for real robots.

Acknowledgments

We would like to thank Mathieu Desbrun for his precious help in reviewing this paper.

References

[1] A. Witkin and Z. Popovic. Motion Warping. *SIGGRAPH'95*, 105–108, 1995.

[2] B. Blumberg and T. Galyean. Multi-level Direction of Autonomous Creatures for Real-Time Virtual Environments. *SIGGRAPH'95*, 47–54, 1995.

[3] O. Renault, N. Magnenat-Thalmann and D. Thalmann. A Vision-based Approach to Behavioral Animation. *The Journal of Visualization and Computer Animation*, 1(1):18–21, 1991.

[4] H. Noser,). Renault, D. Thalmann and N. Magnenat-Thalmann. Navigation for Digital Actors Based on Synthetic Vision, Memory, and Learning. *Computer & Graphics*, 19(1):7–19, 1995.

[5] B. Espiau. An Overview of Local Environment Sensing in Robotics Applications. *Sensors and Sensory Systems for Advanced Robots*, Springer-Verlag, 43:125–151, 1988.

[6] F. Sillion and C. Puech. Radiosity and Global Illumination. *Morgan Kaufmann Publish.*, San Francisco, 1994.

[7] F. Cazals and G. Drettakis and C. Puech. Filtering, Clustering and Hierarchy Construction: a New Solution for Ray Tracing Very Complex Environments. *Eurographics'95*, Maastricht, the Netherlands, September 1995.

[8] C. L. Vaughan, B. L. Davis and J. C. O'Connor. Dynamics of Human Gait. *Human Kinetics Publishers*, Champaign, Illinois, 1992.

[9] Y. Roth-Tabak and R. Jain. Building an Environment Model Using Depth Information. *Computer*, 22(6):85–90, June 1989.

[10] A. Elfes. Occupancy Grid: A Stochastic Spatial Representation for Active Robot Perception. *Proceeding of the Sixth Conference on Uncertainty in AI*, July 1990.