A panoramic synthetic vision model for improving collision avoidance in crowd simulation

Ricardo Bustamante de Queiroz*, Teófilo B. Dutra*, Creto A. Vidal* and Joaquim B. Cavalcante-Neto*
*Departamento de Computação, Universidade Federal do Ceará, Fortaleza, CE, Brasil
Emails: {ricardobqueiroz, teofilo.dutra}@gmail.com, {joaquimb, cvidal}@lia.ufc.br

***Abstract*—Crowd Simulation is very important in many virtual reality applications, because it improves the sense of immersion of the users by making the population of agents in the environment to move as real crowds do. Recently, models for simulating crowds, in which each agent is equipped with a synthetic vision system, have shown interesting results regarding the natural manner in which the agents navigate inside the environment thanks to their visual perception. In this article, we propose an upgrade to the agent's visual system with a panoramic view in order to allow an agent to expand its vision beyond the limit of 180° imposed by the common projection provided by rendering APIs. Also, we analyze different parameters, which are used to define the field of view, to investigate the influence they have on the agent's behavior. The impacts that those changes may cause on the efficiency of the algorithms are also analysed. A visible change on the agent's behavior is achieved by using the technique, with a slight loss of performance.**

***Keywords*-crowd simulation; synthetic vision; panoramic vision; collision avoidance**

I. Introduction

An important characteristic in a virtual reality environment is the credibility it gives to the spectator or to the user that is interacting with it. The more credible the environment is, the greater the users' sense of immersion will be. Many aspects contribute to improve the immersion, from the realism of the modeled environment to the addition of autonomous virtual avatars (agents) that will interact with such environment. More specifically, researchers from the Crowd Simulation field have dedicated their efforts to simulate the behavior of a diversity of autonomous agents in virtual environments.

Several crowd simulation models have been proposed in recent years. A recent class of agent-based algorithms equipped with a synthetic vision intends to simulate crowds through the use of a visual perception system based on the human vision system. That kind of model is also referred to as velocity-based model for using dynamic information, such as the velocity, to predicting movement. Models based on synthetic vision have received some attention as much for simulating the human visual perception simulation as for presenting results that are visually pleasant.

This work has two important contributions. The first is an analysis of the influence of the field of view on vision-based crowd simulation models. This analysis is motivated by the absence of such studies in models of this type up to now. The first vision-based model for crowd simulation [1], for instance, uses arbitrary values to define the field of view of its agents. The second contribution consists of the implementation of a panoramic field of view that allows vision-based models to use a field of view of up to 360°.

The study analyzes the influence of several variables involved in the definition of the field of view, not limited only to the horizontal angle, but examining the vertical angle and the visual orientation as well. The obtained results demonstrate that the usual parameters of the field of view used in [1] and [2] seem appropriate to solve the proposed problem. Notwithstanding, with increasing angle, you gain the power of simulating new situations with non-humanoid agents, or situations in which the agents have peripheral vision.

The remainder of the article is organized into five sections. In Section II, the works related to the study are presented. In Section III, the algorithm used for the panoramic vision is presented, as well as its integration to the models proposed by Ondřej [1] and Dutra [2] is described. In Section IV, the methodology of the analysis is presented. In Section V, the results of the analysis are exposed and discussed. Finally, in Section VI, some conclusions are made and future works is discussed.

II. Related Works

The main objective of Crowd Simulation is to determine the movement of multiple agents that results from a collective behavior. This subject has received attention for many different purposes, from games and movies industry to civil engineering, and a great number of solutions can be found in the literature [3], [4], [5]. These solutions can be classified into macroscopic and microscopic approaches regarding the crowd movement. Macroscopic approaches handle the global flow of the crowd, not depending on local interactions between agents. Models based on guidance fields [6], [7], [8], [9], [10] are the most representative examples of such approach. On the other hand, microscopic approaches model local interactions among the agents, and how they affect the movement of each other. However, global patterns might emerge on local simulations as a consequence of the local interactions between the agents.

A huge amount of microscopic algorithms has been proposed. Local interactions can be modeled based on: examples [11], [12], rules [13], [14], [15]; or forces, as it is done in particle systems [16], [17]. Recently, velocity-based approaches have received increasing attention. Such models anticipate the danger by extrapolating the agents' trajectories in order to detect possible collisions in the near future [18], [19], [20], [21], [22], [9], [23], [24].

A specific category of approaches equips each agent with a synthetic vision (SV) system. SV models mimic the way humans foresee future risks of collision and how they react to those risks. Those algorithms are inspired by the literature that acknowledges the importance of the human vision system in the action-perception loop used for locomotion [25], [26], [27]. In pioneering works, the field of view of an agent could be modelled as a geometric area [13] or a volumetric representation of the scene [28], where the agents would interact with obstacles found within their field of view. The first explicit simulation of locomotion resorting to an agent's synthetic vision was introduced in [29]. The first approach to use synthetic vision for crowd simulation was proposed by [1]. That model transforms the visual input of each agent into images containing information that allows it to detect the risk of collision with any obstacle or agent in the scene. Agents will react to the stimulus by turning to avoid a future collision when detected with anticipation, and by reducing its speed to avoid imminent collisions. Ondřej's model is purely reactive, where the agents react only to the most imminent danger. An approach that takes into account a wide range of solutions was introduced in [2]. That work describes a gradient-based model for steering agents with synthetic vision, where a cost function, used to evaluate the agent's current situation, has to be optimized at each step in order to adapt the agent's motion, so it avoids collisions and moves toward the goal.

Models based on SV [1], [30], [2] have presented good results and represent a great advance for the crowd simulation area. Nevertheless, there are still some questions to be investigated for this kind of model. The visual system is way more complex than what is presented in those works. In fact, multiple variables can influence the visual system. Considering only the field of view, we can see that both the horizontal and vertical fields of view can be changed by movements of the eye or the head. In those models, it is possible to notice that the parameters of the field of view were defined in an arbitrary way, where the authors used the horizontal angle of $150°$ and the vertical angle of $80°$ with the vision rotated by $-40°$, whereas the literature testifies that the human field of view is approximately $180°$ and $135°$ on the horizontal and vertical directions, respectively [31], [32].

The literature acknowledges the importance of the field of view to human navigation. Experiments with constrained field of view have resulted in: locomotion difficulties, wider but slower movements with the head, difficulty to keep the posture and difficulties on the environment spatial representation [32]. The importance of the field of view has also been studied from the user's point of view while navigating in virtual environments [33], [34]. However, its influence on the locomotion of autonomous agents using models based on synthetic vision has not been discussed. For this reason, we present in this work a study of the influence of the agents' field of view on their locomotion using the models described in [1], [2]. Despite the existence of diverse variables to be analysed on the vision, this study will focus on only three of them: horizontal field of view, vertical field of view and the vertical orientation of the vision. Finally, in order to allow the use of horizontal angles equal or greater than $180°$ during the experiments, a panoramic vision system was implemented based on PanQuake by Van Oortmerssen [35]. That panoramic vision was applied to the agents used by the models selected for the experiments.

III. Panoramic view on a model based on synthetic vision

Generally, the field of view of agents in virtual environments is based on the pinhole camera model. That kind of model defines the vision as a frustum, to which planar projections are associated. It is possible to see in Figure 1 that a field defined in that way is limited to angles lower than $180°$. Moreover, distortions start to appear as the angle increases due to the planar nature of the projection.

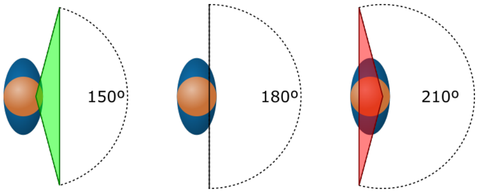

Figure 1. Example illustrating different projections of fields of view. The image on the left shows the example of an horizontal angle of $150°$. The image on the center shows an example with an angle of $180°$. In this case, the planes *near* and *far* coincide, i.e., the frustum volume is zero. The image on the right shows an example with an angle of $210°$. The problem in that case is that the *far* plane is behind the agent's head causing undesired errors of projection.

Both [1] and [2] models for controlling autonomous agents based on vision use a planar projection with field of view of $150° \times 80°$ (Figure 1 (left)). Moreover, that kind of projection is limited to angles lower than $180°$. This restricted field of view limits the model to simulating only entities with narrow fields of view, where several different entities with wide horizontal fields of view (greater than $180°$) found in nature, such as birds [36], are neglected.

That limitation motivated the addition of a panoramic vision to this kind of model. Inspired by PanQuake [35], the panoramic vision was adapted to render wider horizontal field of view angles on the agent's vision. The approach is based on images and consists of rendering the environment from multiple points of view (by rotating the camera on the Y axis) and joining these images in a panoramic view of the scene (Figura 2).

In order to describe the algorithm, two variables must be defined. The first one refers to the desired projection angle, and the second one to the number of slices (separate rendering) used to define the panorama. The changes are applied to the models at the moment the scene is rendered at the point of view of the agent. Previously, the camera was positioned at the agent's "head", and then set up. Now, as we can see in Algorithm 1, after positioning, it is

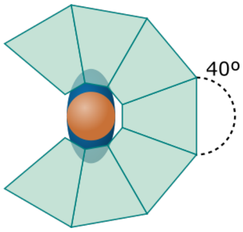

Figure 2. Panoramic view example. In this example, a panoramic field of view of 280° is composed by 7 different projections of 40°.

necessary to perform s different renderings (one for each slice) (Lines 2 to 11) adjusting the viewport (Line 4) and the frustum (Lines 6 to 9) for each one of them. All the slices are rendered within different viewports in the same image (Line 10). The function called at the Line 10 works the same way as in the original models, but here it is called once for each slice. Finally, the functions in lines 12 and 14 are executed the same way as in the original models.

Algorithm 1 Panoramic vision implementation.

```
1:  for all agent $a$ do
2:    for all slice $s$ do
3:      $sliceWidth \leftarrow \frac{textureWidth}{numberOfSlices}$
4:      $setViewport(s \times sliceWidth, 0,$
                $sliceWidth, textureHeight)$
5:      $sliceFoV \leftarrow \frac{fovX}{numberOfSlices}$
6:      $projection \leftarrow createFrustum(nearValue, farValue,$
                $sliceFoV, fovY)$
7:      $rotY \leftarrow \frac{1}{2} \times (-fovX + (2 \times s + 1) \times sliceFoV)$
8:      $projection \leftarrow rotateY(projection, rotY)$
9:      $setProjectionMatrix(projection)$
10:     $texture(s) \leftarrow renderVisionToTexture()$
11:   end for
12:   $processVision(texture)$
13: end for
14: $doModelSecondLoop()$
```

The same algorithm works for both SV models since the agent's camera set up is the same for them as well as the organization of the rest of the algorithms. Their differences lie in the shaders used for processing the motion variables and in the processing step after rendering the agent's vision. That said, the input for the model's algorithm that will determine the agent's behavior will be the same even after we incorporate the panoramic vision.

In Figure 3, some implementation results are shown, where the agent view is illustrated for different fields of view, at the initial step of the circular scenario (Figure 3 (left)).

IV. Methodology of the Analysis

The study performed in this work analyses the influence of some of the fundamental characteristics to defining the field of view of an agent equipped with a synthetic vision. In addition to the usual characteristics such as: horizontal angle, vertical angle and vision orientation; it is also analysed a new characteristic acquired by the adoption of the peripheral vision: the number of slices that composes the vision.

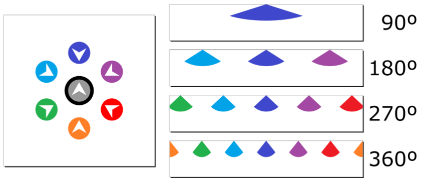

Figure 3. The initial step for a circular scenario is shown on the left. On the right, different fields of view are shown for the agent at the center on the left image. Its horizontal field of view varies from 90° to 360°, and the vertical angle fixed to 80°.

Figure 4 shows three types of trajectories regarding two agents: diverging trajectories (on the left), parallel trajectories (on the center) and converging trajectories (on the right). In the first two trajectories, there is no risk of collision, whereas in, the third one, the collision will occur if both agents keep their current trajectories. The case on the center image represents the turning point where the agent might collide if it does any change toward the other agent. The closer the agents are, the lower will be the angle necessary for them to see each other (Figure 5).

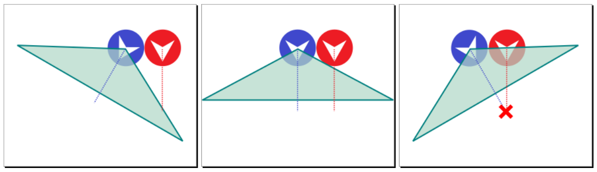

Figure 4. (left) The angle between the trajectories of the agents is greater than zero. They are getting away from each other. (center) The angle between the trajectories of the agents is zero. (right) The angle between the trajectories of the agents is lower than zero. They are in a collision route.

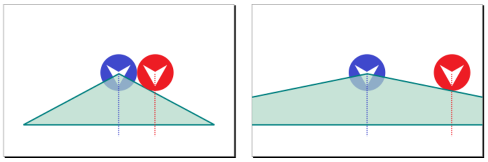

Figure 5. The closer the agents are, the lower will be the angle necessary for them to see each other.

Assuming that the agents have the same radius R, the lowest angle necessary so that the two agents can see each other is calculated from the configuration illustrated in Figure 6. In that case, the agents are side by side and the distance between their centers is equal to $2R$. Given that, the vision angle should be 2α and the value of $alpha$ is $\arccos\left(\frac{R}{2R}\right) = \arccos\left(\frac{1}{2}\right) = \frac{\Pi}{3} = 60°$. For this reason, the vision angle should be at least 120° to ensuring that at least one of the agents can see the other when there is a chance of collision.

The angle of 150° used for the horizontal angle in [1], [2] guarantees to the agent enough visibility so that they can avoid each other. The remaining parameters: vertical angle and vertical orientation, have the values 80° and −40°, respectively. Those values make the top part of the frustum to be aligned with the height of the agents,

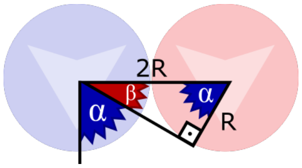

Figure 6. In an extreme situation, where agents are side by side and their radius are equals to R, 2α should be the lowest horizontal field of view angle so that an agent can see the other by its side.

and the bottom part to have a coverage able to perceive anything moving close to the observer. The tests performed consisted of changing the initial values proposed by the SV models, and studying the changes caused by it. For the horizontal angle, the values of 50°, 150° and 250° were studied. The angle values were chosen this way in order to evaluate three cases when it is: lower then the minimum angle; between the minimum and 180°; and, greater than 180°. Each case was tested for 1, 4 and 16 slices.

V. Results

In this section, the results obtained with the two models are analyzed. Throughout the section, Ondřej's model will be referred as OSV and Dutra's gradient-based model [1]) as GBM. The first results are shown in Figure 7 with a scenario where two agents walk side by side, whereas a third agent moves toward them, since they are on the way to its goal. This scenario was chosen for requiring an immediate perception from the agent's surroundings and for having obstacles that might cause collision in case of any wrong movement. In that figure, it is possible to notice that the number of slices has little influence on the overall behavior of the agent. However, since the generated images are actually projected from different angles, there might be some cases where the trajectory of the agent is significantly changed by the number of slices, specially when the symmetry is very high and any noise can cause the agent to take completely different actions. In those special cases, even taking a different path, the agent will continue respecting the interactions with the world and behaving as expected with the model.

We analyzed also the influence of the horizontal field of view while changing the number of slices. In this scenario, there was no noticeable difference between the results with different number of slices. However, the variation of the horizontal view angle greatly changed the results, creating different behaviors on the agents, as shown in Figure 8, where multiple horizontal field of view angles were tested with the SV models. When using the angle of 30° we notice that the GBM with the default parameters have way more anticipation capabilities than OSV. This caused the agent to avoid the other two agents even with a narrow field of view. In OSV, a collision occurred because the agents were too close when they stopped seeing each other. When the blue agent noticed the green agent's presence, it was already too late. Using the default angle of 150° the two models behaved very similarly. But it is possible to see that the agent using GBM goes a bit farther in order to avoid the two obstacles. Using the angle of 270° it is possible to notice a new behavior for both models. When the blue agent noticed the approach from the green agent, it started to reduce its own speed preemptively or to turn around in order to avoid future collision. If the red agent notice any movement on the blue agent, it will also move.

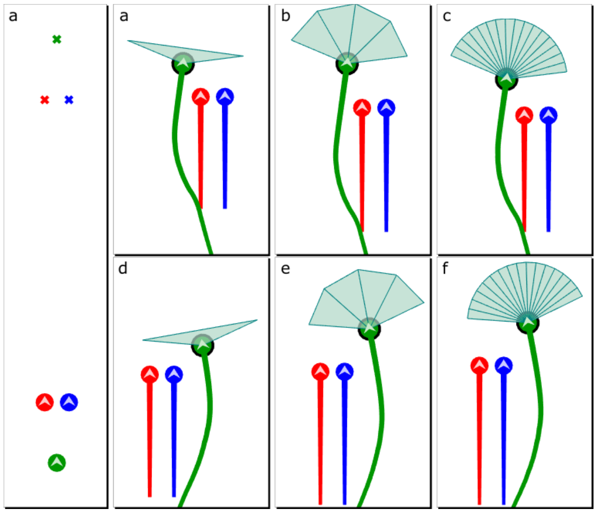

Figure 7. In the first column (a), the initial configuration: two agents (red and blue) walk in parallel, while a third agent (green) wants to move through them with a greater speed. The X mark shows the goals of the agents. By testing different number of slices (1, 4 and 16 from left to right), there are almost no changes in the behavior of the agent. (b), (c) and (d) were using OSV, whereas (e), (f) and (g) were using GBM.

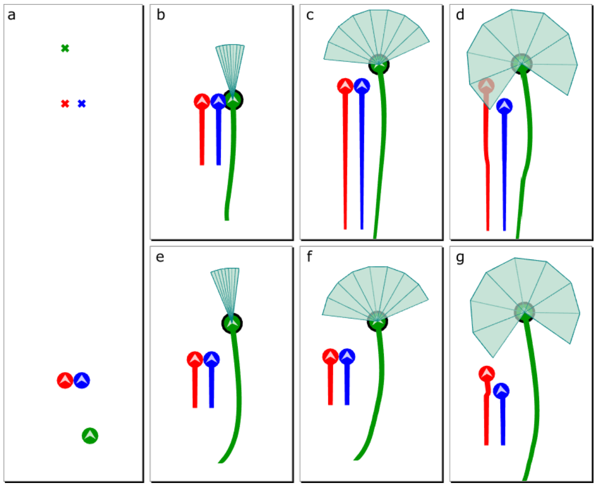

Figure 8. Multiple angles were tested for both models. (a) is the initial scene configuration. (b), (c) and (d) are using OSV. (e), (f) and (g) are using GBM. The horizontal field of view is 30° in (b) and (e), 150° in (c) and (f) and 270° in (d) and (g).

The next parameter studied was the vertical field of view of the agents, even not being modified by the addition of a panoramic vision. The default values for OSV and GBM are 80° of vertical opening and −40° of vertical rotation. This means that the top plane of the frustum is horizontal, whereas the bottom plane is almost parallel to the simplified agent mesh conical surface. Changing the angles to make the bottom plane go away from the agent's body is possible, but there would be some cases where the agent cannot see objects close to him, leading to collisions. The top plane angle was modified by changing the two angles. As expected, lowering the angle from the top plane causes the agent to lose anticipation capabilities. As we can see in Figure 9, in both models the narrow vertical angle caused the agents to start their avoidance movements quite later than when they were using the default angle values. The values of the vertical angles might be changed in order to simulate different situations, such as agents with vision impairment, different species of agents or event agents with different states of attention where the head position might be relaxed and looking down.

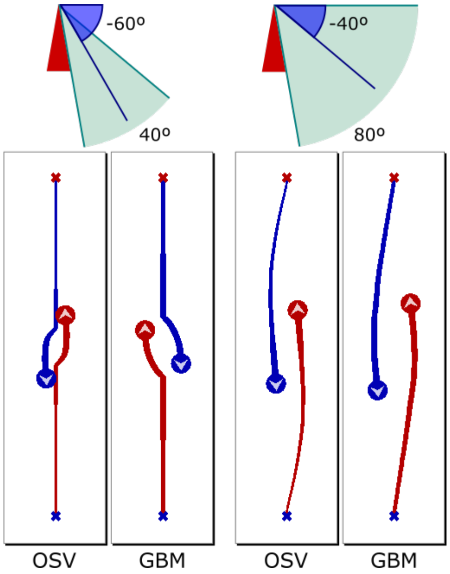

Figure 9. The image shows how the agents behave in a simple scene varying the top plane inclination. In the first two columns, it is shown the execution for the vertical angle of $40°$ and rotation of $-60°$, both for OSV and GBM. The last two columns represent the agent trajectories when the angles have the default values of $80°$ and $-40°$.

A. Performance

In order to conclude the study, an performance analysis was made by measuring the frames per second rate (*FPS*) of different parameters of the panoramic vision technique applied to OSV and GBM. The parameters studied were: number of agents in the scene, with values of 10, 50, 100 and 250; number of slices composing the vision rendering, with values of 1, 2, 4, 8 and 16; angle of the horizontal field of view, with values of 30, 90, 150, 270 and 360; and, the two angles that define the vertical field of view, with values of (40,−60), (80,−40) and (120,−20).

Three different scenes were used to testing the parameters. The first is the *circle scene*, where all the agents are put equidistant from a common point, facing each other. The objective of each agent is to cross the diameter of the circle and reach the opposite side. The second is the *swapping groups scene*. Agents are separated in two groups, and the groups must swap their positions by crossing at the center. Similarly, the *crossing groups scene* consists of two groups, but they must cross in a perpendicular path. These scenarios were a sample of the different scenes used in [1], [2] to evaluate the agents' behavior.

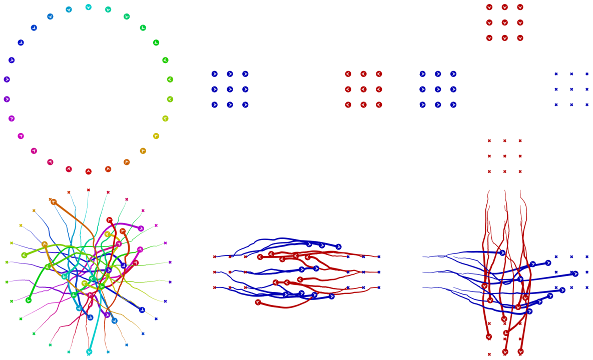

Figure 10. The different scenes used to testing the performance: circle (left), swapping groups (center) and crossing groups (right).

All possible combinations of the mentioned values were made, each one defined as a tuple (e.g. $(OSV, circle, 50, 150, 80, -40, 4)$). Then, they were sent to each algorithm for running the simulation for 10 seconds. The measured value of FPS of each execution was stored and then plotted to different graphs to compare how each parameter influenced the performance. The FPS of each frame is $\frac{1 frame}{time\_in\_seconds}$. Since the time is measured in milliseconds in the application, the following formula was used to measure the average FPS over all the execution:

$$FPS = \frac{1}{n}\sum_{i=0}^{n-1}\frac{1000.0}{time\_measure_i} \tag{1}$$

where n is the number of frames of the trial.

The analysis result is a dataset of tuples that contain all the parameters as well as the average FPS measured. Figure 11 shows how the different parameters affect the performance of the algorithms. The top graphic shows the performance impact caused by the number of slices in the vision of the agent. It is possible to see that the FPS is almost unchanged by the slices variation. Even if the technique makes the algorithm perform multiple rendering steps to generate a single image, the number of computations used to generate the image is proportional to the size of the buffer. Since the total buffer size does not change, the overall execution time is only changed by the overhead of calculating a new projection and setting a new viewport for each slice. The number of objects being rendered might also affect the performance of rendering each slice, but due to the clipping algorithm, objects that are outside the view frustum are left out of the computation, avoiding multiple calculations for the same object.

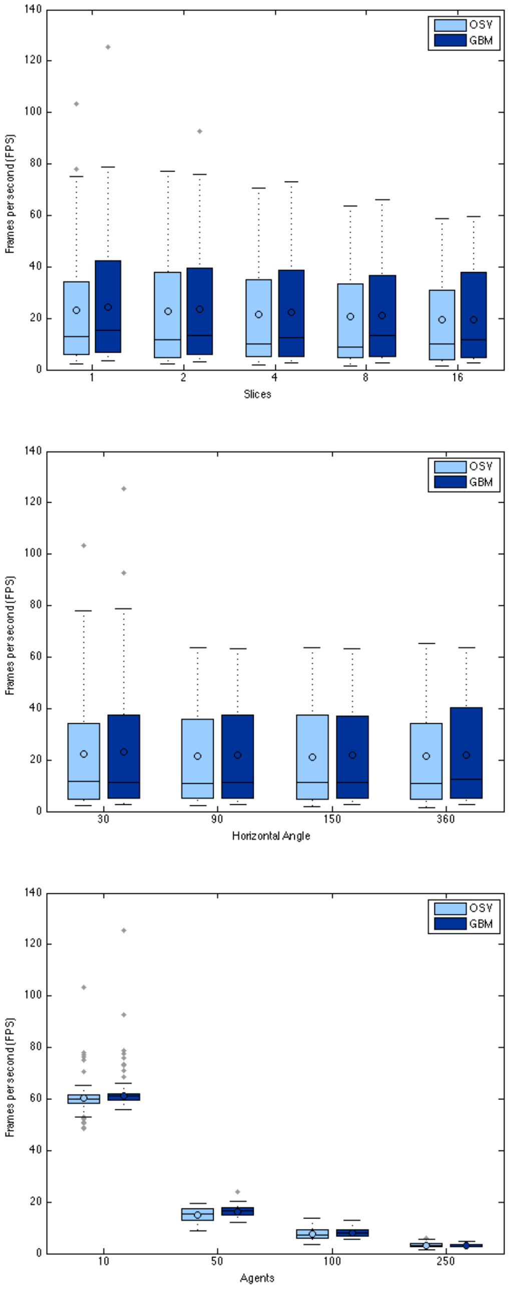

Figure 11. Results for the performance test. The graphics show the influence of the number of slices of the vision (top), the horizontal angle (middle) and the number of agents (bottom) on the FPS for each model.

The plot at middle shows how the horizontal angle affects the performance. As expected, just changing the angle value causes a minimal performance variation. The time of rendering each frame might be affected by the amount of vertexes contained in the scene, but since this step is being computed in parallel by the GPU, the change caused almost no variation on the plot.

The bottom graphic represents the FPS variation caused by the amount of agents on the scene. Since for each agent a new image of the scene must be rendered, it is expected that the change of this parameter causes a greater impact on the performance. It is possible to see that there is a small variation between the samples from the same amount of agents. This means that this dimension is the one that defines the wide range of frames per second measured. If all the data had been read with a fixed number of agents, the variance achieved would have been lower.

The tests were performed on a desktop machine with the following configurations: *Intel(R) Core(TM) i5-2400 CPU @ 3.10GHz, 8GB of RAM and Video card GeForce GTX750 2GB SC DDR5.*

VI. Conclusion

In this work, the influence of the field of view on the trajectory of autonomous agents was explored. The study was performed with two models for simulating crowds based on synthetic vision: OSV [1] and GBM [2]. In order to realize the study,the visual system used by the agents on both models had to be upgraded so as to handle horizontal angles greater than 180°. For that reason, a panoramic vision algorithm [35] was integrated to the agent's visual system. The two biggest contributions of this work were: the use of the panoramic vision by autonomous agents in vision-based algorithms for crowd simulation, and the analysis of the influence of the field of view in such type of simulation.

For the study, experiments were executed with some samples of the variables used to define the field of view of the model. From the experiments, the conclusion was that the value of the horizontal field of view used in both OSV and GBM is appropriate to avoiding collisions, given that two agents in a collision route are able to see each other at time for avoiding a collision in the near future, when having an horizontal angle greater than 120°. It became evident that the chosen values for the vertical angle and vertical rotation were adequate for the specific situation, since with 80° and $-40°$ respectively, the agent was capable of seeing everything close enough from its body, and below an horizontal plane over its head. Since all obstacles in the simulation are contained in that region, the agent is capable of seeing everything within its horizontal field of view. However, it is worth to mention that changing the vertical angle allows new situations with restricted vision to be simulated. Notwithstanding, it was also observed that increasing the number of slices did not cause noticeable losses on the performance of the application. With a slice count up to 16 it is possible to have minimal decreases in performance and keeping a good panoramic vision quality up to 360°.

Increasing the size of field of view allowed developing new ideas to making the agent's movement more believable. A normal human being possesses a central vision and a peripheral vision and the existing models still do not take that into account. A new model could then be developed taking into consideration the different regions of the human field of view. Other possibilities include studying the influence of the agent's head rotation, allowing it change the direction it is looking while keeping itself on its trajectory. Despite allowing more collisions to happen, it would be possible to model a system that is more credible, since collisions actually happen sometimes in a real crowded environment.

Finally, there are different aspects to be studied related to the simulation of visual perception. Other types of projection, like the fish-eye view, common in sportive photographic cameras due to its wide aperture, can also be explored as alternatives to the panoramic view used in this work. Furthermore, it would be interesting using the synthetic vision to simulate the perception of different species of agents, given that, for example, a human agent possesses a different field of view configuration from a bird agent [36], that has its eyes on the side of the head.

References

[1] J. Ondřej, J. Pettré, A.-H. Olivier, and S. Donikian, "A synthetic-vision based steering approach for crowd simulation," *ACM Trans. Graph.*, vol. 29, no. 4, pp. 123:1–123:9, Jul. 2010.

[2] T. B. Dutra, "Gradient-based steering for vision-based crowd simulation algorithms," Ph.D. dissertation, Fortaleza, 2015. [Online]. Available: http://www.mdcc.ufc.br/teses-e-dissertacoes/teses-de-doutorado

[3] N. Pelechano, J. M. Allbeck, and N. I. Badler, "Virtual crowds: Methods, simulation, and control," *Synthesis Lectures on Computer Graphics and Animation*, vol. 3, no. 1, pp. 1–176, 2008.

[4] D. Thalmann and S. Musse, in *Crowd Simulation*. Springer London, 2013.

[5] S. Ali, K. Nishino, D. Manocha, and M. Shah, in *Modeling, Simulation and Visual Analysis of Crowds: A Multidisciplinary Perspective*. Springer New York, 2013.

[6] S. Chenney, "Flow tiles," in *Proc. of the 2004 ACM SIGGRAPH/Eurographics symp. on Computer animation*, 2004, pp. 233–242.

[7] A. Treuille, S. Cooper, and Z. Popović, "Continuum crowds," in *SIGGRAPH '06*. ACM, 2006, pp. 1160–1168.

[8] R. Narain, A. Golas, S. Curtis, and M. C. Lin, "Aggregate dynamics for dense crowd simulation," *ACM Transactions on Graphics*, vol. 28, pp. 122:1–122:8, 2009.

[9] M. Kapadia, S. Singh, W. Hewlett, and P. Faloutsos, "Egocentric affordance fields in pedestrian steering," in *Proceedings of the 2009 symposium on Interactive 3D graphics and games*, ser. I3D '09. New York, NY, USA: ACM, 2009, pp. 215–223.

[10] S. Patil, J. van den Berg, S. Curtis, M. C. Lin, and D. Manocha, "Directing crowd simulations using navigation fields," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, pp. 244–254, February 2011.

[11] A. Lerner, Y. Chrysanthou, and D. Lischinski, "Crowds by example," *Computer Graphics Forum*, vol. 26, no. 3, pp. 655–664, 2007.

[12] E. Ju, M. Choi, M. Park, J. Lee, K. Lee, and S. Takahashi, "Morphable crowds," *ACM Trans. Graph.*, vol. 29, pp. 140:1–140:10, 2010.

[13] C. Reynolds, "Flocks, herds and schools: a distributed behavioral model," *SIGGRAPH Comp. Graphics*, vol. 21, no. 4, pp. 25–34, 1987.

[14] W. Shao and D. Terzopoulos, "Autonomous pedestrians," *Graphical Models*, vol. 69, no. 5-6, pp. 246–274, 2007.

[15] S. Singh, M. Kapadia, B. Hewlett, G. Reinman, and P. Faloutsos, "A modular framework for adaptive agent-based steering," in *Symp. Int. 3D Graphics and Games*, 2011, pp. 141–150.

[16] D. Helbing and P. Molnár, "Social force model for pedestrian dynamics," *Physical Review E*, vol. 51, no. 5, pp. 4282–4286, 1995.

[17] D. Helbing, I. Farkas, and T. Vicsek, "Simulating dynamical features of escape panic," *Nature*, vol. 407, no. 6803, pp. 487–490, 2000.

[18] C. Reynolds, "Steering behaviors for autonomous characters," in *Game Developers Conference 1999*, 1999, pp. 763–782.

[19] S. Paris, J. Pettré, and S. Donikian, "Pedestrian reactive navigation for crowd simulation: a predictive approach," *Computer Graphics Forum*, vol. 26, no. 3, pp. 665–674, 2007. [Online]. Available: http://dx.doi.org/10.1111/j.1467-8659.2007.01090.x

[20] J. van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *IEEE Int. Conf. on Robotics and Automation*, May 2008, pp. 1928–1935.

[21] I. Karamouzas, P. Heil, P. van Beek, and M. H. Overmars, "A predictive collision avoidance model for pedestrian simulation," in *Motion in Games*, ser. LNCS. Springer, 2009, vol. 5884, pp. 41–52.

[22] J. Pettré, J. Ondřej, A.-H. Olivier, A. Cretual, and S. Donikian, "Experiment-based modeling, simulation and validation of interactions between virtual walkers," in *Proc. of the 2009 ACM SIGGRAPH/Eurographics Symp. on Computer Animation*, ser. SCA '09. New York, NY, USA: ACM, 2009, pp. 189–198. [Online]. Available: http://doi.acm.org/10.1145/1599470.1599495

[23] S. J. Guy, J. Chhugani, C. Kim, N. Satish, M. Lin, D. Manocha, and P. Dubey, "Clearpath: Highly parallel collision avoidance for multi-agent simulation," in *Proc. of the 2009 ACM SIGGRAPH/Eurographics Symp. on Computer Animation*, ser. SCA '09. New York, NY, USA: ACM, 2009, pp. 177–187. [Online]. Available: http://doi.acm.org/10.1145/1599470.1599494

[24] J. van den Berg, S. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Robotics Research*. Springer, 2011, vol. 70, pp. 3–19.

[25] J. Cutting, P. Vishton, and P. Braren, "How we avoid collisions with stationary and moving objects," *Psychological Review*, vol. 102, no. 4, pp. 627–651, 1995.

[26] W. Warren and B. Fajen, "Optic flow and beyond," L. M. Vaina, S. A. Beardsley, and S. K. Rushton, Eds. Norwell, MA, USA: Kluwer Academic Publishers, 2004, ch. From optic flow to laws of control, pp. 307–337.

[27] K. W. Rio, C. K. Rhea, and W. H. Warren, “Follow the leader: Visual control of speed in pedestrian following,” *Journal of Vision*, vol. 14, no. 2, 2014.

[28] A. R. D. Silva, W. S. Lages, and L. Chaimowicz, “Boids that see: Using self-occlusion for simulating large groups on gpus,” *Comput. Entertain.*, vol. 7, no. 4, pp. 51:1–51:20, Jan. 2010. [Online]. Available: http://doi.acm.org/10.1145/1658866.1658870

[29] O. Renault, N. Magnenat-Thalmann, and D. Thalmann, “A vision-based approach to behavioural animation,” *Journal of Visualization and Computer Animation*, vol. 1, no. 1, pp. 18–21, 1990.

[30] T. B. Dutra, G. Priem, J. B. Cavalcante-Neto, C. A. Vidal, and J. Pettré, “Synthetic vision-based crowd simulation: reactive vs. reactive planning approaches,” in *Proceedings of the 27th Conference on Computer Animation and Social Agents (CASA 2014)*, 2014.

[31] *Effects of field of view on human locomotion*, vol. 6955, 2008. [Online]. Available: http://dx.doi.org/10.1117/12.771950

[32] S. Jansen, “Obstacle avoidance behaviour during locomotion: Strategy changes as a result of visual field limitations,” 2012.

[33] J. Ardouin, A. Lecuyer, M. Marchal, and E. Marchand, “Navigating in virtual environments with 360o omnidirectional rendering,” in *3D User Interfaces (3DUI), 2013 IEEE Symposium on*, March 2013, pp. 95–98.

[34] ——, “Stereoscopic rendering of virtual environments with wide field-of-views up to 360o,” in *Virtual Reality (VR), 2014 iEEE*, March 2014, pp. 3–8.

[35] W. van Oortmerssen, “PanQuake,” 2015, http://strlen.com/gfxengine/panquake/.

[36] G. Martin, “Visual fields in woodcocks scolopax rusticola (scolopacidae; charadriiformes),” *Journal of Comparative Physiology A*, vol. 174, no. 6, pp. 787–793, 1994. [Online]. Available: http://dx.doi.org/10.1007/BF00192728