Modeling Multi-Robot Interaction Using Generalized Occupancy Grids, With Application To Reducing Spatial Interference

Mauricio Zuluaga Richard Vaughan
School of Computing Science, Simon Fraser University, Burnaby, BC, Canada
mzuluaga,vaughan@sfu.ca

***Abstract*— As part of a program to find methods of reducing spatial interference in multi-robot systems, we propose the *Interaction Grid (IG)*, a generalization of the Occupancy Grid that models the spatial distribution of interactions between robots. We describe alternative methods for building Interaction Grids, first by recording instances of actual robot-robot interaction, and then by a much faster approximation method. The resulting maps of interaction likelihood can subsequently be used to modify the robot's behaviour to avoid interference. For example, we show how to automatically generate an aggression map for input to the aggressive display behaviour we have previously shown to be effective in interference reduction.**

I. Motivation: Reducing Spatial Interference in Multi-Robot Systems

Interference in general can be characterized as competition for resources, for example, access to a charging station or use of a shared tool or sensor. Most commonly, robots simply get in each other's way during normal navigation about the environment. An acute version of this problem is getting two Pioneer–sized (0.5m) robots through a standard (0.8m) doorway from opposite directions; some symmetry–breaking mechanism is required to decide who goes first. This is a real–world problem for robot applications such as mail delivery, factory and warehouse AGVs, and assisted–operator wheelchairs. Another common scenario is shown in Figure 1, where two robots are driving in opposite directions down a narrow corridor, blocking each other's progress.

Previously in [1], [9]–[11] we have shown that a stereotypical aggressive display behaviour can be effective in reducing interference in robot teams, thus improving their overall performance. The goal was to maximize the total work done by a team of robots in a canonical transportation task in an environment in which robots frequently interfere with each other, as occurs in a standard office building. Because the robots had to work in the same space territorial methods [3] were not appropriate. In our method, when robots come into competition for floor space, each selects an *aggression level* and the competition is resolved in favour of the more aggressive robot. Choosing an aggression level for each robot at random is effective in breaking symmetry [9], but performance can be further improved using a systematic, rational approach.

Global-Investment (GI) maps the aggression of a robot to the "effort", measured by time, it has put into reaching a goal. The more effort invested the most aggressive the robot will be. This method was shown to have a small advantage over random aggression in environments where the measured effort was well correlated to the cost of losing an aggressive interaction, i.e.an environment containing many long, narrow corridoors. [1]

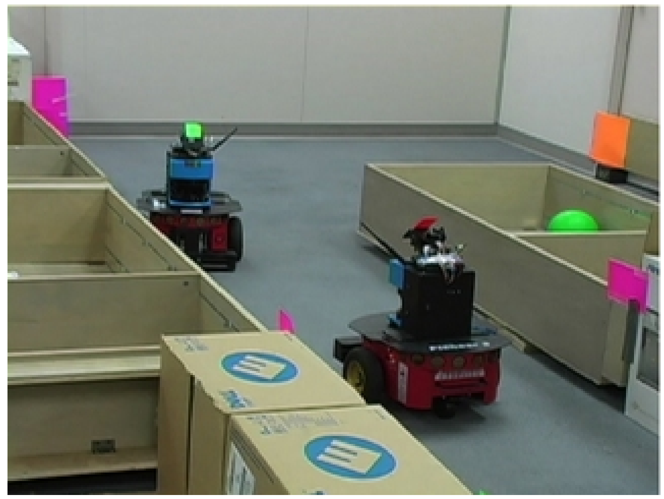

Fig. 1. Spatial-Interference: two robots block each others' way in a narrow corridor [11].

Local-Investment (LI) was designed to solve the problems with GI. In this method, aggression is proportional to *local* task investment only, i.e. aggression is proportional to the amount of time spent navigating a narrow corridor, but in more open areas aggression is set at random. This method, though more complex than GI, should be useful in a wider variety of environments, so we focus on LI hereafter [11].

Though LI could be used in many different situations and has the benefit of being scalable and not requiring the use of communication, it has a key parameter that must be tuned to the particular environment for maximum effectiveness. Ideally, the rate that aggression accumulates while traversing narrow corridors needs to be adjusted so that robots reach their maximum agressiveness at the end of the corridor and not before. In practice this means we need to identify all corridors and their traversal time in advance.

Clearly we would prefer to relax this constraint, and allow the robots to identify the location and extent of areas of high interference probability in an unsupervised way. To do so, we introduce the concept of the Interaction Grid, a generalization of the Occupancy Grid.

II. Interaction Grids

We define the *Interaction Grid (IG)* as a metric grid of cells of equal size, each corresponding to a place in the world, whereby the cell contents are updated by interaction events at that location.

We intend "interaction" to be interpreted broadly. A grid accumulating events triggered by the interaction of the robot's body or sensors with a static obstcale is the classical Occupancy Grid [7]. A grid mapping the visibility of a robot to an observer is the Observability Map [2], [8]).

The interaction of interest in our application is one robot detecting of the presence of other robot along its desired path. The IG is therefore a matrix in which the value of each element is a function of the probability of an interaction event occuring at the corresponding small area of the world, and so the overall grid is an approximation of the probability distribution of robot interactions over space.

In the following sections we describe different methods for building IGs. Then we show how IGs can be used for building an unsupervised aggression strategy that increases the performance of a robot team in a transportation task.

III. Building Interaction-Grids

> "All models are wrong, some models are useful", George Box [5]

We all know that forecasting of city traffic conditions can only be done to a certain degree. There are many different variables that have an impact on it: time, day of the week, weather, etc. Not all variables are known, not all can be measured. Still some of them can be approximated, for instance we use rush-hour as a good indicator of tough traffic conditions, and though we also know that it is not perfectly accurate we still use it to plan our decisions.

In the case of robots sharing the same environment knowing where interactions will take place presents a similar problem. In the following sections we show different models for this problem. Experimental data shows that using these models we can improve the performance of a team of robots in a transportation task, which we briefly describe here.

A. Robot system

For convenience our experiments are performed using the well-known Stage multi-robot simulator [6], though we have previously validated the aggressive behaviour strategy in real robots [11]. Figure 2 shows the environment for these experiments and those described above. There are two rooms, containing the source and goal destinations respectively, and five mobile robots equipped with sonar and laser range sensors. The world measures 11 by 12 meters. All robot models are the standard models shipped with Stage, and approximate the ActivRobot Pioneer 3-DX and SICK LMS200 rangefinder.

All robots use the same controller which follows the architecture first described in [9] (Figure 3). Three behaviors are defined: *Navigate* is essentially a left-wall follower that for certain areas of the world can follow a heading instead, this is how the robots get into the rooms. *Fight* is a behavior triggered when robots compete for the right of way (i.e. in narrow sections of the world). *Panic* is oriented to take robots out of deadlock situations, for example when they are stalled or have been fighting for too long. A detailed description of this behaviour-based control architecture is given in [1].

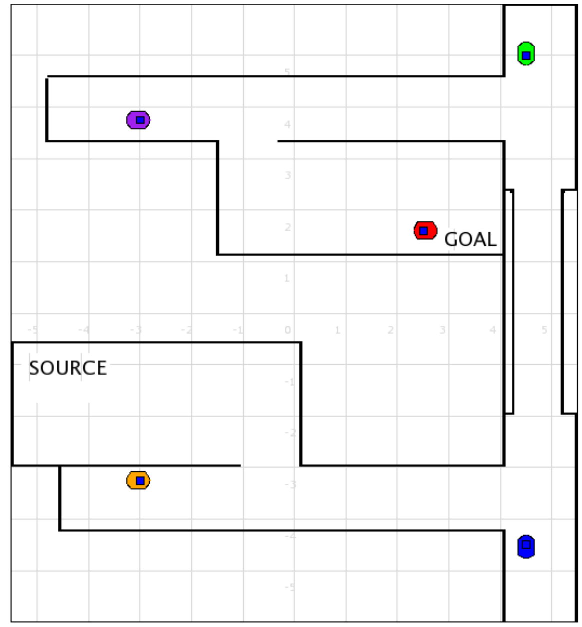

Fig. 2. The environment use in the simulation experiments (11x12 meters)

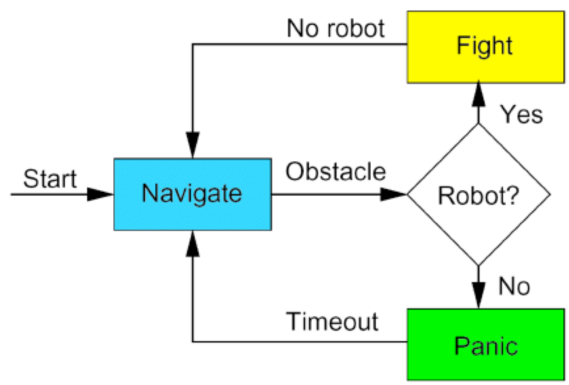

Fig. 3. Control Architecture

B. Method 1: Accumulating Interactions

When the experiment is started all robots start in the configuration shown in Figure 2. The robots task is a type of transportation task in which they are asked to pick resources at a source location and take them to a "goal" location. In the experiments the robots do not actually transport anything, they just navigate between "source" and "goal". After 10 minutes we stop the simulation.

For the IG we have chosen a cell size of 50 cms, for our environment (11x12m) this gives us a grid of 22x24 cells.

All robots share the same IG. At the beginning of the experiment we initialize all cells of the IG to zero.

The interaction event we record is the transition from the

Navigate state to the Fight state in the robot controller. This indicates that the robot can no longer navigate and make progress on its task; i.e. it has suffered from interference.

After the robots start moving through the world, all fight-starting events are simply accumulated in the corresponding grid cell (note that this assumes the robots are globally localized).

C. Results

Figure 4 (top row) shows a plot of the values stored in the IG at 30,60,120,240 and 480 seconds. White pixels are zero, darker pixels indicates higher values. Comparing the plots with the map of the environment (Figure 2), we observe that interference encounters between robots have been recorded only in the narrow corridor on the right. The IG provides evidence that this is an area of the world in which the robots often interfere with each other.

Note that the controller used by the robots in simulation could also be used by real robots without any modification. There are only two additional requirements: communication and localization. For communication it is possible to use a wireless network. Even if this is not feasible our method can still be used though every robot would have its own personal IG. For localization standard AMCL can be used, this has been proven to work well on many real situations. [4]. Also consider that the IG generating method does not require any knowledge of the details of the robot's navigation system.

This simple accumulation method can be seen to produce potentially useful maps of the world indicating areas of likely interference. But where interference events are relatively rare - a desirable condition - the maps take a long time to produce. However, these maps are absolutely correct in the sense that interference is known to occur at the non-zero grid cells.

IV. Method 2: The Bread-Crumbs Approximation method

We seek to speed up the aquistion of interaction grids, so we propose the following aapproximation method. The basis of the approximation is that spatial interference can only occur when two or more robots are co-located, so we record locations where that *could* occur.

Definitions:

K	Number of Robots
N	Number of Cells
$P(C_i^j)$	Probability of cell i being occupied by robot j
$P(C_i)$	Probability of cell i being occupied by any robot
B_i^j	Bread-crumbs in cell i left by robot j
B_i	Bread-crumbs in cell i
$S2G$	Robot's path from Source to Goal
$G2S$	Robot's path from Goal to Source

The values we store in the IG correspond to the probabilities that a cell may be occupied by any of the robots ($P(C_i = 1)$). Our simplifying assumption is that the probability of interference is a function of the probability of co-location.

We can write $P(C_i)$ in terms of $P(C_i^j)$:

$$P(C_i) = \sum_{j=1}^{K} P(C_i^j) \tag{1}$$

Because all robots follow the same path the $P(C_i^j)$ values are the same for all robots. This we simplify Equation 1 to:

$$P(C_i) = K \times P(C_i^1) \tag{2}$$

$P(C_i^1)$ is the probability that cell i is occupied by robot 1.

Conditioning C_i^1 over the route followed by the robot we obtain:

$$P(C_i^1) = P(C_i^1|S2G)P(S2G) + P(C_i^1|G2S)P(G2S) \tag{3}$$

$P(G2S)$ and $P(S2G)$ are roughly the same because it takes approximately the same time to go from source to goal than from goal to source. And also after the experiment has run for some time the number of trips between goals is very similar (at most one more trip from S2G). We can further simplify to:

$$P(C_i^1) = 0.5 * (P(C_i^1|S2G) + P(C_i^1|G2S)) \tag{4}$$

$P(C_i^1|S2G)$ and $P(C_i^1|G2S)$ are quantities that are not the same. They are defined by the trails the robots leave when going in one direction or the other.

We assume the probability for a robot to be in any cell along its path to be uniform. If we allow a single robot to leave a trail of bread-crumbs while moving in the environment, the number of bread-crumbs left on every cell of the path would be the same. As a result $P(C_i^1)$ is proportional to the number of bread-crumbs that robot i leaves at cell i. If we accumulate bread-crumbs for all robots we can simplify Equation 2 to:

$$P(C_i) = K \times P(C_i^1) = \alpha B_i \tag{5}$$

Where α is a proportionality constant.

And so the probability of occupancy of a cell is proportional to the number of bread-crumbs found inside the cell.

A. Results

Figure 4 (bottom row) shows plots of the IG at 60,120,240 and 480 seconds measured using this method. We can see that the accumulation of bread-crumbs on the environment makes the narrow corridor emerge as an area of the world in which robot are more likely to be co-located and thus more interactions between robots are expected.

Comparing the IGs produced by the two methods, we see they are not identical, yet they both highlight the narrow long corridor section. Simple thresholding would render the results almost identical. But the bread-crumb co-location method produces information much more quickly than the direct event-recording method.

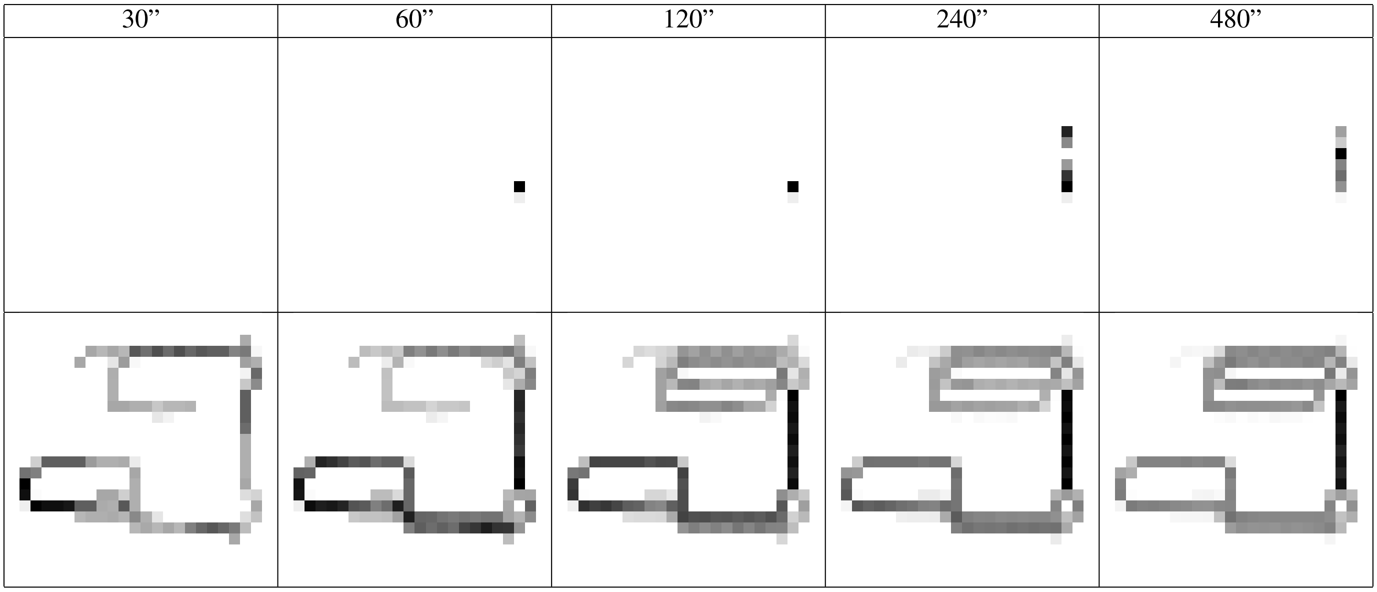

Fig. 4. Models of interaction likelihood

V. ϵ-SUPERVISED AGGRESSION ESTIMATION USING IGS

In this section we demonstrate that IGs can be used to automatically generate aggression strategies that do not require many parameters to tune and therefore reduce the need for *a priori* tuning.

Here we will use the IG to generate a *Aggression Grid (AG)*, a similar array in which the cell values indicate the aggression level. A robot entering a fight would take the aggression value indicated by the cell corresponding to its current location.

A. From IG to AG

AGs are generated in the following way:

1) Clean IG: We are only interested in areas where interaction is likely, any cell of the IG which value is above a threshold we fix to the threshold value. We have set the threshold to be the mean of the IG before cleaning.
2) Log IG values along robot’s path: We send a robot from Source-to-Goal-to-Source. Every δ_{time} time we record the robot’s position and IG value.
3) Smooth and find Edges: We do edge detection with thresholding and Gaussian smoothing. We fix the threshold to 0.2 and sigma σ to 3. The edges found delimit the areas of the world where aggression should be minimum or maximum.
4) Use the sign of the gradient at the edges found to determine if aggression should be minimum or maximum.
5) Move the min aggression markers to the left, and max aggression markers to the right a δ_{path}. This has the overall effect of making the aggression function minimum a little before getting into the high-interaction area, and maximum a little after passing it.
6) Interpolate: We interpolate the aggression values between the min and max cells along the robot’s path.

Figure 5 shows the plots for the logged cleaned IG values, the min and max aggression points and the interpolated aggression values along the robot’s path.

All we have at this point is a one-dimensional aggression function. We still need to build a two-dimensional grid. The steps we show next provide a simple way of doing it.

1) Initialize AG: First we initialize/label all cells of the AG to a constant value of UNKNOWN.
2) Set Robot’s Path: Next we set all cells along the robot’s path with an UNDEFINED label. We make this distinction to later enable the use of heuristics for different areas of the world.
3) Finally set the values of the cells between min and max points to the values found in the one-dimensional aggression function.

Figure 6 shows the results of executing the previous steps on the previous IG data. The first column shows the IG we use for input. The second column shows the cell’s in the robot’s path. Darker cells are cells where the robot was long ago while Lighter cells are recent visited cells. This way we can infer the direction in which the robot moves (Source to Goal or Goal to Source). The third column displays the cells that mark the edges in the IG, they define the beginning and end of high-interaction areas of the environment along the robot’s path. Finally, the fourth column shows the computed AG. White cells are UNKNOWN areas, light-gray cells are UNDEFINED areas and the light to dark gradient areas indicate aggression values along the high-interaction areas, which in our case is along the narrow corridor on the right of the figure.

B. Why is this a good strategy?

We believe the AG found with the previous method to describe a good rational strategy to cope with spacial interference.

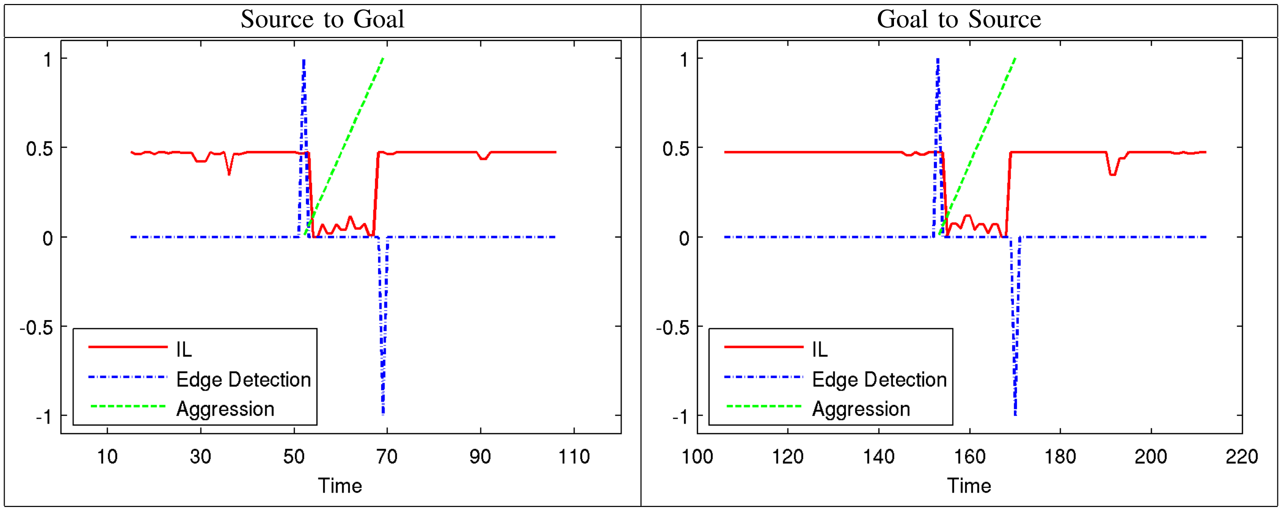

Fig. 5. Aggression Generation

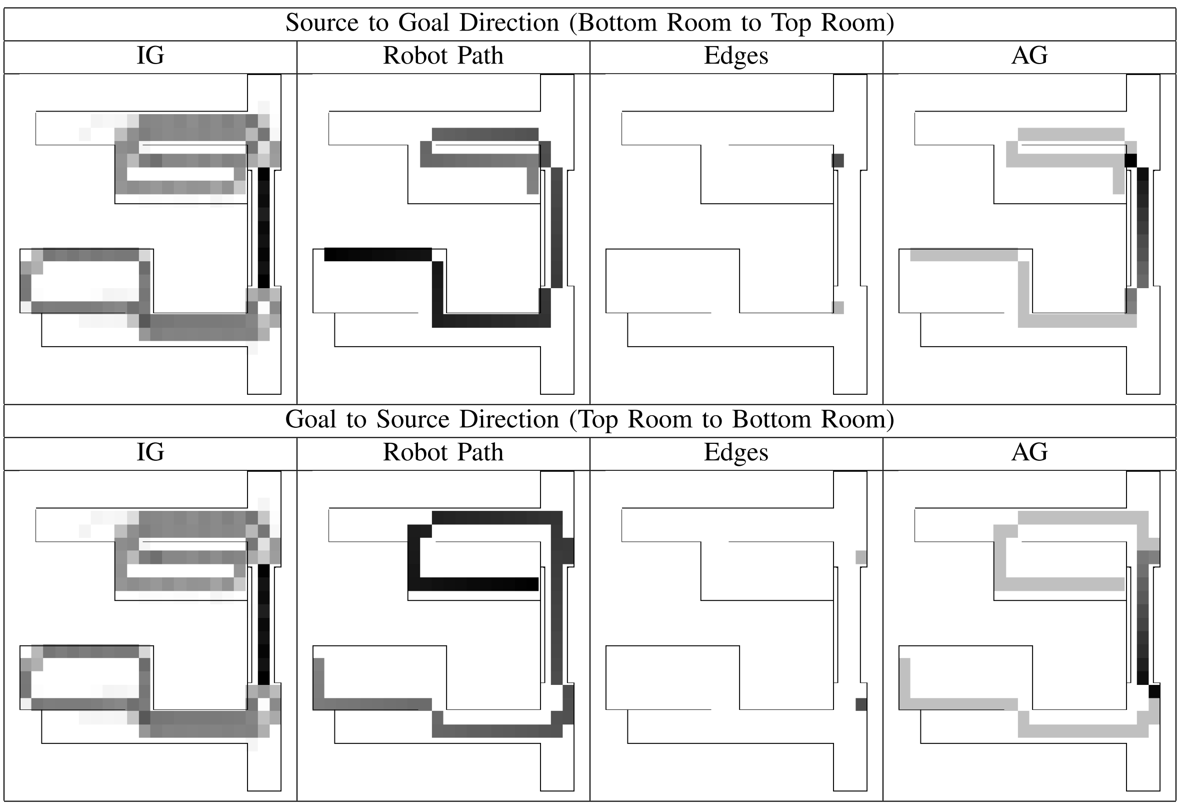

Fig. 6. Aggression Grid Generation

In a random strategy when two robots meet they both chose their aggression at random and the fight is solved in favor of the robot with the highest aggression. In [9] Vaughan et. al. introduced this technique and also found that many other strategies produced results that were not statistically different from the random case (e.g. hierarchies, amount-of-front-empty-space, etc).

Later in [11] we presented the LI strategy. LI is an strategy in which aggression is set based on local-effort. For example measuring how much effort has been put in passing along a narrow corridor. This strategy was shown to improve over previous strategies and specially over a random-only strategy.

With the method we propose in this paper if we choose to use random aggression for UNKNOWN or UNDEFINED areas of the world, and use the value from the AG in all other areas we get a strategy very much like the local-investment strategy. Figure 7 shows grids with the aggression values for both the old-LI method and the one we propose in this paper. The Aggression values inside the narrow corridor areas are similar. And so the strategy generated by the new method is

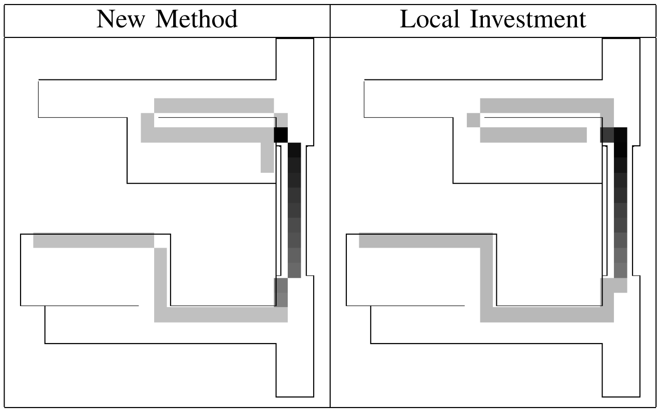

Fig. 7. Aggression Grids generated from IG information and using Local Investment

also better than a random strategy.

VI. Conclusion

In this paper we have described a generalization of the Occupancy Grid we call the Interaction Grid. We have demonstrated a direct and an approximation method for building IGs, and shown how to use the resulting map as input to an interference reducing behaviour.

We believe IGs are a potentially powerful device for modeling interaction between robots, or between robots and other agents, with many potential applications. To the best of our knowledge, this simple model has been overlooked to date. Indeed, when building traditional occupancy grid maps using range sensors, care is taken to avoid adding detections of robot teammates or other transitory objects to the static map. We are exploiting the data that is usually thrown away.

References

[1] Sarah Brown, Mauricio Zuluaga, Yinan Zhang, and Richard T. Vaughan. Rational aggressive behaviour reduces interference in a mobile robot team. *The 12th International Conference on Advanced Robotics ICAR2005*, 2005.

[2] Birgersson E., Howard A., and Sukhatme G.S. Towards stealthy behaviors. *IEEE/RSJ International Conference on Intelligent Robots and Systems IROS2003*, 2003.

[3] M. S. Fontán and M. J Matarić. Territorial multi-robot task division. *IEEE Transactions on Robotics and Automation*, 14(5):815—822, 1998.

[4] D. Fox, W. Burgard, F. Dellaert, and S. Thrun. Monte carlo localization: Efficient position estimation for mobile robots. In *National Conference on Artificial Intelligence (AAAI)*, 1999.

[5] Box George. *Robustness in the strategy of scientific model building, in Robustness in Statistics*. Academic Press, New York, 1979.

[6] Brian Gerkey, Richard T. Vaughan, and Andrew Howard. The player/stage project: Tools for multi-robot and distributed sensor systems. In *Proceedings of the 11th International Conference on Advanced Robotics (ICAR)*, Coimbra, Portugal, June 2003.

[7] H. Moravec and A. Elfes. High resolution maps from wide angular sensors. *International Conference on Robotics and Automation ICRA1985*, 1985.

[8] Ashley Tews Gaurav S. Sukhatme and Maja J. Mataric. A multi-robot approach to stealthy navigation in the presence of an observer. In *IEEE International Conference on Robotics and Automation, New Orleans, LA*, pages 2379–2385, May 2004.

[9] Richard T. Vaughan, Kasper Stoy, Gaurav S. Sukhatme, and Maja J Mataric. Go ahead, make my day: robot conflict resolution by aggressive competition. In *Proc. Int. Conf. Simulation of Adaptive Behaviour, Paris, France*. MIT Press, 2000.

[10] Yinan Zhang and Richard T. Vaughan. Ganging up: Team-based aggression expands the population/performance envelope in a multi-robot system. *International Conference on Robotics and Automation ICRA2006*, 2006.

[11] Mauricio Zuluaga and Richard T. Vaughan. Reducing spatial interference in robot teams by local-investment aggression. *IEEE/RSJ International Conference on Intelligent Robots and Systems IROS2005*, 2005.