Proceedings of the 2003 IEEE/RSJ
Intl. Conference on Intelligent Robots and Systems
Las Vegas, Nevada · October 2003

Adaptive Patrol for a Group of Robots

Francois Sempé
France Telecom R et D
38 du Gal Leclerc
92131 Issy-les-Moulineaux, France
francois.sempe@lip6.fr

Alexis Drogoul
Laboratoire d'Informatique de Paris 6
4 place Jussieu
75252 Paris Cedex 05, France
alexis.drogoul@lip6.fr

***Abstract*— Patrolling is a basic task for a group of robots. All regions of an area are checked at regular interval and robots may look for intruders or garbage to collect. This work proposes a reactive and adaptive approach of this problem. In a virtual environment shared by robots, task data are propagated from place to place with modifications in order to represent locally distant-tasks value. A robot has access to task data situated in the region where it stands and follows a gradient that guides it to valuable regions. First experiments on a simulator are presented. In addition of being efficient in achieving the work, the proposed architecture shows interesting properties of adaptability concerning the group size, the environment size and type.**

I. INTRODUCTION

The patrolling problem consists in visiting at regular interval all places of given area and gathering information. It is a basic collective task as many other tasks like cleaning or surveillance rely on it. A good collective architecture should adapt itself to the robots group size and environment size and topology because some robots may break down and some areas may be temporarily restricted.

This work explores simple mechanisms to control robots and to propagate task data through a virtual environment shared by robots. The parameter controlling propagation range is tuned with the help of a feedback value and allows the system to adapt itself to the number of robots and different environments.

Next section will tackle the patrolling task problem. Then an overview of patrolling method is proposed. System architecture and experimental results take place in sections 4, 5 and 6. First, hand tuned parameter results are shown, and then an adaptive policy is proposed.

II. THE PATROLLING TASK

A. Different aspects of patrolling task

In its most common version, patrolling consists in visiting all places of an area at regular interval. The first goal of a patrol is to gather information about the environment, whether it is for security or cleaning purpose, looking for intruders or garbage to collect. In term of efficiency, the global knowledge of the environment should be maintained as up-to-date as possible, i.e. each place should be visited alternatively and quickly.

In order to manage the patrolling task, the environment can be divided in zone that we call region. One region is connected to its neighbouring region and the whole area can be described as a graph where nodes are regions and edges a path between two adjacent regions. To patrol is not just to walk around: information has to be collected and it takes time. Depending on the sensors used by robots, the aspect of the terrain and the type of information that is looked for, a robot will spend more or less time in visiting a region. A "fire robot" with a heat sensor seeking a starting fire in a corridor should just move without a stop. At contrary, when an object is searched with camera, a robot might for instance stop and do a panoramic. Moreover, robots might have a second task like collecting encountered garbage, which represents another delay. The time spent by a robot to gather information for a given region is called the visit duration, which is an important parameter that modifies system dynamic.

Another significant factor of the patrolling task is the visit homogeneity. The region weight represents the visit frequency required for one region relatively to other region. For instance, a weight-of-2 region will need to be visited twice more than a weight-of-1 region. At one extreme, for the higher homogeneity, all regions have the same weight and should be visited repeatedly with the same interval of time. The situation becomes more complex when weights and visit durations are different or, worse, if they change during the patrol. Unpredictable visit duration may require a relay between two robots, when the first robot has to stop its work because it needs to go and recharge its battery.

B. Software vs. hardware patrolling

In [3], we can find a comparison of different multi-agent architecture applied to patrol. Software agents have to visit the nodes of a graph. The authors emphasize that many environment can be patrolled, virtual or physical, from network to building. Software agents move from node to node at a light speed (one node per simulator step) and no physical interference can occurs. But interferences are known to be a critical problem in robotics [7] and robots concentration must be avoided. In addition, another robot constraint is not taken into account: agents do not need to charge battery, which might have severe consequences on the on the efficiency of a strategy. [8] explores a patrolling policy, which works fine when robots do not have to join a charge station, but fails even to cover all the area when energetic autonomy is limited.

C. Evaluation criterion and reference

As proposed in [3], the idleness is a simple and meaningful evaluation criterion. A region visiting value is the time elapsed since the last visit, eventually weighed for non-homogeneous patrolling. Instantaneous idleness is the sum of visiting values divided by the sum of weights. Final idleness - or idleness to be short - is the average of instantaneous idleness over a run.

In order to compare efficiency between different experimental situations, a normalized idleness is proposed. During its patrol a robot has 4 distinct activities: moving from last visited region to next region that will be visited - or visit seeking -, visiting a region, going to a recharge station and recharging its battery. The two last activities ratio is almost constant. A good patrolling implies to minimize visit seeking and to visit only the best-visiting-value regions. Our reference idleness assumes that only best-visiting-value regions are visited and a robot wastes only one region movement between two visits, in average. Then normalized idleness is idleness over reference idleness.

III. REACTIVE PATROLLING

A. The need for task data propagation

The following review takes into account papers about the coverage problem, which is in fact a one-shot patrol.

The simpler patrol strategy is random displacement. It is commonly used for ant-like foraging problem [2] but offers poor efficiency in term of idleness [3]. Robots can rely on local data to avoid recently visited regions and to reach best-visiting-value one's. In this case, robots communicate through the environment in tagging visited regions. Flags [3] or timestamps [10], [8] of neighbouring regions are perceptible by robots, which move toward the most interesting region. Those strategies demonstrate to be efficient when no battery charge is necessary and with null visit duration. If the latter two conditions are not fulfilled the most distant areas from recharge stations may be not visited at all as shown in [8]. It happens that freshly charged robots waste time in visiting close-to-station regions and have to join again a station before reaching far regions.

One way to overcome this problem is to propagate task data from region to region.

B. Data propagation and gradient following

Data propagation has been applied to robotics for different purposes like foraging [1] and path finding [6]. More closely related to patrol problem, [5] makes an intensive use of data propagation for military units to select targets and find the best path. Various artificial pheromone flavours are used, one for each enemy unit type (target, air defense...) and may be attractive or repulsive. The range propagation trade-off, that will be discussed later, is solved using simultaneously many pheromones threads with different dynamics. Mobile agents (bombers and fighters) send "ghosts" that travel at network speed to plan units next step.

Patrolling can be seen as a target or task allocation where each robot has to choose one region to visit. However the situation exposed in [5] offers some significant distinctions to ours: the environment is an open space, interferences and refuelling are not taken into account, targets are scattered all over the simulated battlefield and agent types are many. We have adopted a more bottom-up approach. In our case, the information relative to a task is the visiting value. It represents what would be gained at the global level (idleness criterion) if a given region were visited. Propagated task data produce a gradient that guide robots to valuable regions. Robots are informed of neighbouring task values, climb the gradient, from region to region, and visit the first encountered vertex-gradient region. Once visited, the visiting value of a region drops to zero and gradient following can start again.

IV. SYSTEM ARCHITECTURE

How to propagate data and make it perceptible by robots? A hardware approach can be found in [4], [5]. It consists in scattering all over the working area electronic devices, which are used as propagation relays. One relay exchange data to propagate with neighbouring relays and can provide local data to neighbouring mobile agents through wireless communication. The software-based approach simulates propagation on a virtual working area shared by robots. We have adopted the latter solution that provides much more flexibility in term of installation.

Robots have localization and local navigation abilities and can estimate their remaining autonomy. Through a wireless communication network they send to the propagated data world (PDW) their localization and remaining autonomy and inform it when a visit begins and ends. The PDW runs on a remote computer and computes the task strength (TS) for each region in the following way.

$$TS_i = MAX(OTS_i, MAX_{N_i}(TS - (DLS * PLR)))$$

N_i is the neighbouring regions set of region i. Own task strength (OTS) refers to the task strength produced by a region itself and is equal to its visiting value as described in section 2. PLR is the Propagation Loss Rate and control the propagation range. Distance Loss Rate (DLR) is proportional to the distance between centres of a given region and a neighbouring region. It has been set to the

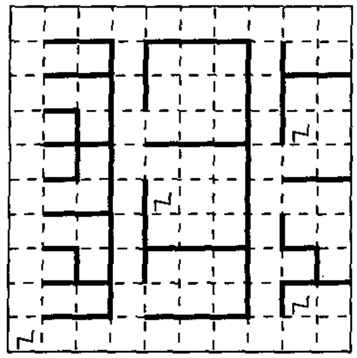

Fig. 1. The large office environment with location of the 4 charge stations. It is a 20x20m area divided in 100 square regions

travel duration between centres at average robot speed (0.2m/s).

Robots are driven by propagated data. They climb the task strength gradient and visit the first encountered gradient-vertex region. A region is a gradient vertex if its own task strength is above the propagated task strength.

V. EXPERIMENTAL SET-UP

As mentioned above, energy management is a key point in mobile robotics. There are a lot of ways to approach this problem, from heavy and highly autonomous robots that work all day long and get recharged by night to lighter robots that need to go and charge their battery at short intervals. But in the most general case, robots have to interrupt their useful activity to get some energy from time to time.

The following simulated experiments assume that a complete charge provides a 30-minute autonomy. As one run lasts 3 hours, 6 battery chargesper robot are necessary. A robot stops patrolling as soon as its remaining autonomy drops under 10 minutes. Then it is guided by a station gradient to the best station relatively to distance and station business. Charge speed is set to 6 (6 minutes of autonomy gained during one minute of charge) as provided by the recharge station we developed. Natural alternation of work/charge cycles is enough to manage charge stations sharing [9].

Visit duration has been set to 40 seconds, which represents a slow 360-degree panoramic. Reference visit duration is rounded to 70 seconds, taking into account the 20% of time dedicated to recharge phase (joining a station and recharging) and the 20 seconds necessary to travel through 2 regions (4m at 0.2m/s with 2m x 2m regions).

A set of experiments has been realized on a simulator with various group sizes, environment sizes (25 regions in 100m2 and 100 regions in 400m2) and topologies (office

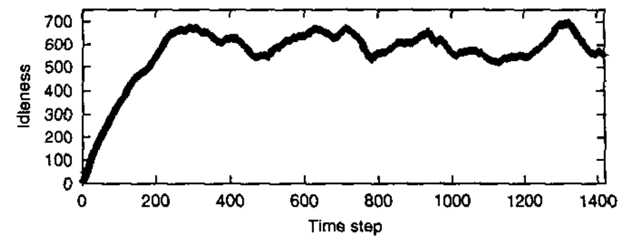

Fig. 2. Typical instantaneous idleness curve.

Environment	*Robot*	*Idle.*	*NI*	*PLR*	*GVR*
Large Office	2	2021	0.86	35	0.45
	4	1099	0.80	20	0.47
	8	589	0.74	15	0.49
Small Office	4	263	0.82	10	0.55
Large Open	4	990	0.88	25	0.45

TABLE I

HAND-TUNED PLR RESULTS. NI REFERS TO NORMALIZED IDLENESS.

and open environment). The 100-region office environment with four charge stations is shown on figure 1. Each experiment is repeated 3 times in the same conditions. Thus, results exposed in the next section are means over 3 runs.

VI. EXPERIMENTAL RESULTS

A. Hand tuned PLR

The PLR has been identified as the most significant parameter and will be the only one to be modified.

In the first set of experiments, PLR were hand tuned and fixed during a run. Figure 2 shows a typical evolution of instantaneous idleness. As all visiting values start to zero, there is a first phase of high rate increasing idleness that last until the end of exploration (when all regions have been visited at least once). The duration of exploration phase is closely correlated with the group and environment sizes. Therefore, we will take into account only the second phase for results.

With a hand-tuned PLR, we obtain satisfying results (table 1). Idleness is around 80% of reference idleness, which represents a very efficient patrolling. One can observe slight and expected efficiency decreasing per robots as the group size grows. Because distance between regions is shorter and robots scattering is easier, the open environment (no wall) scores the most.

B. Understanding PLR effect

How PLR modifies the group behaviour and finally the idleness? Figures 3 to 5 show instantaneous task gradient profiles where a grid node represents a region. Those

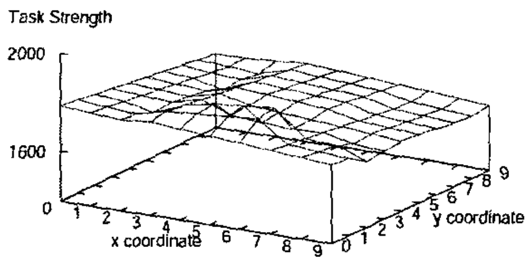

Fig. 3. Task strength profile with too low PLR (1).

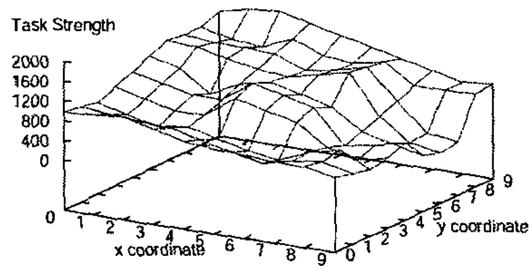

Fig. 4. Task strength profile with right PLR (15).

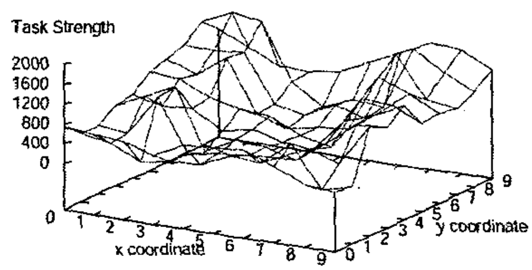

Fig. 5. Task strength profile with too high PLR (30).

figure are taken from 8-robot patrols in a large office environment with 3 different PLR: 1, 15 and 30. When the PLR is set to 1 (lower PLR), the region with the higher visiting value becomes the only gradient vertex of the whole environment. Its propagated OTS covers all others OTS. Therefore all robots rush to the same place and spend a lot of time in following the gradient uselessly. Number of visits is low and regions are visited one by one always in the same order. At the opposite, a too high PLR (30 in this case) makes local and weak regions more attractive than any other. In consequences, number of visits is high but very irregular between regions.

C. *Adaptive PLR*

We are now looking for a feedback value in order to obtain an adaptive system. Idleness itself is too variable and offers too much inertia to PLR modifications. Number of

Environment	*Robot*	*Idle.*	*NI*	*Mean PLR*	*GVR*
Large Office	2	2052	0.85	32	0.41
	4	1120	0.78	21	0.43
	8	611	0.71	13	0.41
Small Office	4	277	0.78	7	0.44
Large Open	4	988	0.88	27	0.42

TABLE II

ADAPTIVE PLR RESULTS.

gradient vertexes is both reactive to PLR modifications and correlated with idleness. With hand tuned PLR, between 45% and 55% of regions are gradient vertexes (gradient vertex ratio or GVR). Therefore the following adaptation rules have been tested:

PLR start to 1. Then, after the exploration phase, each 15mn:

If GVR > 0.55, PLR += PLR Step

If GVR < 0.37, PLR -= PLR Step

PLR step start to 5, then after the first decreasing of PLR, it is set to 2.5 in order to obtain first a quick increasing to good PLR values and then a finer adaptation.

Table 2 shows that adaptive PLR obtain results close to hand-tuned PLR. The difference can be mostly explained by the adaptation phase when PLR increase from 1 to its efficient value.

D. *Experimentation on real robots*

Real world experimentations were conducted with 3 Pioneer 2DX robots from Activmedia equiped with a sonar rings and a camera.

VII. CONCLUSION AND FUTURE WORK

This paper explores a reactive approach to robot coordination through task data propagation in a shared virtual environment.

In addition of being efficient in achieving a patrol, the proposed architecture shows interesting properties of adaptability concerning the group size, the environment size and type. Further researches will focus on irregular patrolling with synchronization problems and experimentations on real robots. Additional propagated flavours and mechanisms should be necessary, like interactions between robots and task strength dynamic. For instance, the presence of a robot in a given region may cause a local PLR increasing in order to repel partners. This mechanism is effective in narrow corridor environment robots interfere a lot with each other's and partner's scattering becomes critical.

VIII. REFERENCES

[1] Alexis Drogoul and Jacques Ferber. *From Tom Thumb to the Dockers: Some experiments with foraging robots.* In Proc. of the Second International Conference on Simulation of Adaptive Behavior, pages 451–459, 1992.

[2] M. J. B. Krieger and J.-B. Billeter. *The call of duty: Self-organised task allocation in a population of up to twelve mobile robots.* Robotics and Autonomous Systems, 30:65–84, 2000.

[3] Aydano Machado, Geber Ramalho, Jean-Daniel Zucker and Alexis. *Drogoul Multi-Agent Patrolling: an Empirical Analysis of Alternative Architectures.* Proc. of Third int. Workshop on Multi-Agent Based Simulation. Bologna, Italy, 2002.

[4] Marco Mamei, Michael Mahan, Letizia Leonardi, Franco Zambonelli. *Motion Coordination for Ubiquitous Agents.* Workshop on Ubiquitous Agents on embedded, wearable, and mobile devices (at AAMAS 2002), Bologna, 2002.

[5] H. V. D. Parunak, S. A. Brueckner, J. Sauter, and J. Posdamer. *Mechanisms and Military Applications for Synthetic Pheromones.* Proceedings of Workshop on Autonomy Oriented Computation, 2001.

[6] D. Payton, M. Daily, R. Estkowski, M. Howard and C. Lee. *Pheromone Robotics.* Autonomous Robots, Vol. 11, No. 3, Kluwer Academic Publishers, Norwell, MA, Nov. 2001, pp 319-324.

[7] Miguel Schneider Fontan and Maja J. Mataric. *Territorial multi-robot task division.* IEEE Transactions of Robotics and Automation, 14(5), 1998.

[8] F. Sempé. *Allocation de tches et partage des ressources par coordination spatiale en robotique collective.* Technical Report - LIP6. Paris

[9] F. Sempé, A. Munoz and A. Drogoul. *Autonomous Robots Sharing a Charging Station with no Communication: a Case Study.* Proc. of the International Symposium on Distributed Autonomous Robotic Systems (DARS),2002.

[10] Israel A. Wagner, Micha Lindenbaum, Alfred M. Bruckstein. *Distributed Covering by Ant-Robots Using Evaporating Traces.* IEEE Transactions on Robotics and Automation October 1999, Volume 15, Number 5, pp. 918-933.