# Navigation in Human Flows: Planning with Adaptive Motion Grid

Jacques Saraydaryan, Fabrice Jumel, Olivier Simonin

# Navigation in Human Flows : Planning with Adaptive Motion Grid

Jacques Saraydaryan[1,2], Fabrice Jumel[1,2] and Olivier Simonin[1,3]

***Abstract*— An important challenge for mobile robots is to navigate efficiently in human populated environments. In this context, we examine how human presence grids can be extended to model human motions, considering only embedded sensors. The proposed flow grid computes in each cell a discrete distribution of the human motion. The model is defined to take into account the most recent observations, so as to adapt to changes. More, it is expanded with a predictive motion pattern. Then we revisit the cost function of the A* path-planning algorithm to take into account the risk of encountering humans. We compare the standard A* with variants exploiting the human presence likelihood [1] and the proposed flow grid. Experiments in simulation show that the Flow grid A* is able to compute paths minimizing the risk of navigating against human flows, and to adapt to their variations. Experiments with a mobile robot confirms the ability of the model to map human flows and to optimize paths.**

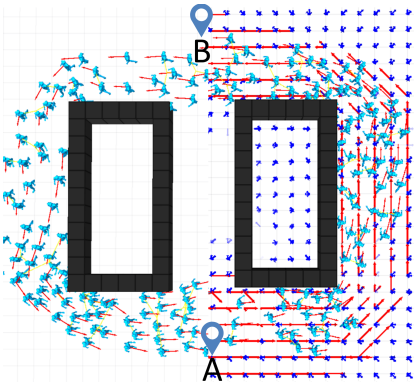


Fig. 1. Simulation of a 200 pedestrian flow. On the right side, the Flow Grid (red arrows) computed by a robot moving in the flow

## I. INTRODUCTION

In this paper, we aim to deal with the problem of robot navigation in dense-human environments, ie. crowded environments (as in pedestrian streets, platforms, public buildings). Generally, robots perceive their working environment from embedded and/or external sensors (eg. fixed cameras). All these information can be merged and shared by robots to build a representation of the current state of the environment or to learn the dynamics, as human flows [2].

The present paper aims to define a grid model able to map human flows and their changes, then to exploit such an information to compute efficient paths. We consider that robots know the static part of the environment (e.g., a metric map) and are able to locate. SLAM techniques allow to make such an assumption [3]. Contrary to many works, we do not consider that the environment is equipped with external fixed cameras or sensors. Therefore, robots have to move in order to cover the environment and to update their knowledge of the human activity. By detecting humans moves, robots can build a map of human presence and of human flows.

First, we examine how human presence grids can be extended to human flow grids, with a minimum of additional information. For this purpose, we compute in each cell a discrete distribution of human motion. We call *flow grid* such a model (illustrated in Fig. 1). Second, we define two A* path-planning algorithms based on cost functions exploiting different kind of information about human flows. The first one uses a human presence map, called affordance map in the work of Tipaldi et al. [1], containing the likelihood of encountering humans. The second model uses the proposed flow grid, containing the likelihood of human motions. We experiment these models in simulation, in order to evaluate their ability to map human flows and to compute paths able to exploit or avoid them. Then we experiment the more promising model with a robot evolving among people walking.

The paper is organized as follows. Section II presents related work to mapping of human flows. Section III introduces the flow grid as an extension of the human presence grids. Then Section IV revisits the cost function of the A* algorithm considering these different flow models. In Section V we present first evaluations of how the flow-grid-A* performs to adapt to changing human flows, in simuation and with a real robot.

## II. RELATED WORK

Recent works have considered the problem of learning human flows from one or several static cameras or LIDAR. In particular we can mention the work of *Tipaldi and Arras* [1] defining the **spatial affordance map**. The model learns the spatio-temporal distribution of events, from the observations in each cell representing the environment. The main limitation of the approach is to only model human presence, i.e. without information about the motion/velocity of humans. Then *Kucner et al.* introduced Conditional Transition Maps [2], which model the probability of human transition between neighbouring cells of a grid representing the environment. Each new observation requires to determine the cell of the observed person, the cell from where he/she arrives and the cell to where he/she goes. This combination of transitions requires to learn **64 parameters per cell**, which requires a

[1]CITI Lab. Inria Chroma team, INSA Lyon 6 avenue des Arts, 69680 Villeurbanne cedex, France

[2]CPE Lyon, Domaine scientifique de la Doua, 69100, Villeurbanne, France

[3]INSA Lyon, Université de Lyon, 20 Avenue Albert Einstein, 69100, Villeurbanne, France

lot of observations to learn dynamics over the whole environment. An extended approach [4] allows long-term spatial correlations while reducing the neighboring transitions to only **4** directions (north, south, east, west).

In [5] and [6] *Wada et al.* have shown that human activity, in particular walking, can be observed and mapped from embedded sensors and with SLAM techniques. After collecting observations from each region of the environment they can generate a human motion grid, giving in each cell the statistics of walking direction in every 15 degree [6].

Gaussian process have been exploited for their ability to predict the motion pattern of people at arbitrary locations in [7]. However this approach exploits a pior map derived from expert knowledge about the environment. Moreover, Gaussian based techniques are known to be computationaly expensive.

Our approach supposes no prior knowledge about the environment and does not exploit any information about human destinations. We examine how presence grids can be extended with motion traces, while weighting the most recent observations to be able to adapt to the flow changes. Then we revisit the cost function of the A* algorithm to exploit this human flow modeling (we discuss path-planning techniques in Sec. IV-A).

## III. Human flow modeling and learning

We adopt a grid representation of human presence and flow in the environment, as for instance in [1], [2]. In such grids, each cell holds the likelihood of encountering humans (ie. presence) or a discrete distribution of human motion direction.

### A. Motion and presence grids

In each cell $c_{x,y}$, we consider a discrete distribution of the motion directions. This defines a set of $K$ directions, noted $k_i$. In practice we discretize in 8 directions.

We note $Z^t$ the set of observations performed on a cell up to time $t$. **An observation at time t**, of a cell $c_{x,y}$, consists in identifying a human direction if a person is present. By hypothesis, only one human can occupy a cell at a given time. In practice, we consider a cell size of $60cm$x$60cm$.

More formally, we note each observation as following:

$$O^t_{c_{x,y},k} = \begin{cases} 1 & \text{if a person is moving in direction } k \\ 0 & \text{otherwise.} \end{cases}$$

In this paper, we consider that sensors are perfect, e.g. we do not model their uncertainty.

Now we note $M_{c_{x,y},k}(Z^t)$ the likelihood to observe a human moving in direction $k$ in cell $c_{x,y}$. The objective is to learn this value from all the observations carried out by the robots on this cell.

As a first approximation, we can compute this likelihood as a standard average of the observations, $\overline{M}_{c_{x,y},k}$, corresponding to a counting model as defined in [8].

It is possible to derive the human presence likelihood, $M_{pres}$, which consists in merging the $k$ directions :

$$M_{pres_{c_{x,y}}}(Z^t) = \sum_{k \in K} \overline{M}_{c_{x,y},k}(Z^t) \qquad (1)$$

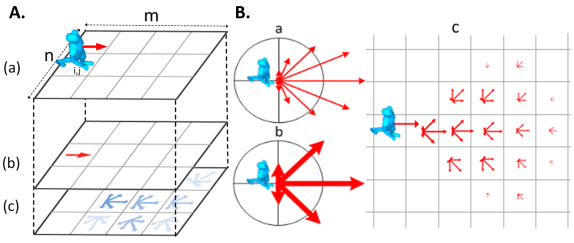


Fig. 2. A. Two layers grid model of the human flows (flow grid and prediction) B. Possible motions from Von Mises Gaussian circle probability distribution: (a) sample of Von Mises probability distribution with $\mathtt{C} = 2$, (b) normalized distribution with $|K| = 8$, (c) prediction direction grid according a given observation.

Merging all the observations allows to find the affordance-map proposed by [1].

### B. The flow grid model

As human flows are not stationary processes, we propose to compute an exponential moving average ([9]) in order to give more importance to the latest observations :

$$M^t_{c_{x,y},k}(Z^t) = \alpha.O^t_{c_{x,y},k} + (1-\alpha).M^{t-1}_{c_{x,y},k}(Z^{t-1}) \qquad (2)$$

$\alpha$ is a parameter in $]0,1[$, then $M_{c_{x,y},k}$ remains in $[0,1]$. A high value of $\alpha$ will give more importance to new observations, while a low value will give more importance to the history of observations. As in occupancy grids [10] we update the likelihood at each new observation. One significant difference is that all $M^t_{c_{x,y},k}$ values are initialized to 0.

As human flows can change rapidly and radically, we need to update the cells information even they are not observed. In other word we let the possibility to vanish the information over time. For this purpose, we compute at a fixed frequency an update of all the cell values $M^t_{c_{x,y},k}$ :

$$M^t_{c_{x,y},k}(Z^t)' = \gamma.M^t_{c_{x,y},k}(Z^t) \qquad (3)$$

$\gamma$ is a parameter in $]0,1]$. One advantage of having $\gamma$ different from 1 is to remove information in area which were not visited for a long time, and so to force their revisit.

We note $M_{flow} = \forall x \forall y \forall k : M_{c_{x,y},k}$ the grid of human motion likelihood in every direction $k$ of each cell $c_{x,y}$.

### C. Expanding the flow grid with a human motion pattern

Building the flow grid from the local robot observations can take a long time, as requiring to visit and revisit the whole environment. In order to accelerate the flow modeling, we add a human motion pattern around the last observations, only in cells without information.

It has been shown that predicting the motion of a pedestrian by a velocity-based linear projection [11] is known to be a reasonable approximation for short term behavior [12]. In our context, we adapt the Von Mises model [13], by considering that a human can take a maximum rotation speed of $90°.s^{-1}$ with an average linear speed of $1m.s^{-1}$.

The main idea of the computation is to propagate the pattern from cells with maximum probability presence towards

direction with maximum probability. Details of the model can be found in [14].

Figure 2.B(c) shows the result of a Von Mises pattern computation. This distribution is normalized according to the number of possible directions ($|K| = 8$ in Fig. 2.B(b)). Then arrow length represents the probability value (Fig. 2.B(c)).

Note that, as soon as a robot observes a human at a given location $c$, its predicted flow is cleared if it exists (for each $k$, $M^{pred}_{c,k} = 0$), then the flow grid is computed.

### D. Illustration of the Flow Grid model

Figure 1 illustrates a first experiment of the Flow Grid computed by one robot (the simulator is presented in Section V-A). The environment contains 200 persons walking around two small rooms. On right, the flow grid is built by the robot crossing the flow during less than 1.5 minutes (the robot and pedestrian nominal speed is $1m.s^{s-1}$). Red arrows show the human flow computed from robot observations and blue arrows show the predicted flow. **The video[1] illustrates such a simulation**.

## IV. Revisiting the A* path-planning algorithm

### A. Cost estimation in human populated environment

In this paper we refer to the well-known A* algorithm widely-used for robot path-planning [15]. A* is a classical algorithm to find an optimal path to a goal using a heuristic to control the order of exploration of the cells. At each iteration of its main loop, A* determines which of its partial paths to expand into one or more longer paths. This is based on an estimate of the cost still to go to the goal node. A* selects the path that minimizes the function $f(n)$. $f(n) = g(n) + h(n)$ where $n$ is the last node on the partial path, $g(n)$ is the cost of the path from the start node to $n$, and $h(n)$ is a heuristic that estimates the cost of path from n to the goal without knowledge. The heuristic used in this paper, $h(n)$, is the classical euclidean distance from $n$ to the goal without obstacle.

To take into account the presence of humans, we consider three different map-distances to compute the cost function $g(n)$ in A* planning :

- the standard A*, i.e. the non informed model, where human presence is equiprobable for each cell.
- the affordance-map [1], estimating human presence.
- the flow-grid map, estimating human motions.

*1) Standard A* algorithm:* The cost between 2 neighbors cells $n$ and $(n-1)$ is defined as:

$$g(n) = \|n, (n-1)\| \times (1 + F) + g(n-1) \qquad (4)$$

where $\|n, (n-1)\|$ is the distance cost between $n$ and $(n-1)$, F is a positive factor to represent human presence. As the disturbance due to humans is not known it is considered as equiprobable in all cells. Then $F$ is set to a constant (1 in the experiments).

[1]https://youtu.be/VeJ11GqwIPU

*2) A* algorithm with affordance-map:* An affordance map estimates the human presence likelihood in each cell as a spatial Poisson process [1]. In each cell $n$, the human presence estimation, noted $Poisson(n)$, is computed as follow: $Poisson(n) = M_{pres_n}$ (see Eq. 1 Section III-A).

This allows to define the following cost function :

$$g(n) = \|n, (n-1)\| \times (1 + F.Poisson(n)) + g(n-1) \quad (5)$$

We add to the distance a cost directly linked to the likelihood of meeting a human when moving to cell $n$, i.e. $Poisson(n)$. The factor F allows to tune the weight of the human disturbance in the cost function.

*3) A* algorithm with Human flow estimation:* The human flow grid and its predicted flow are used to determine the navigation cost.

Let $flow(n)$ be the human flow cost in cell $n$. We note $k$ the direction from $(n-1)$ to $n$ and $\theta$ its relative angle (e.g $k =$'$NE$', $\theta = \frac{\pi}{4}$). We note $\theta_{k_i}$ the angle relative to the direction $k_i$. Then $flow(n)$ is computed as:

$$flow(n) = \sum_{i=1}^{|K|} (1 - \cos(\theta_{ki} - \theta)) \times M_{n,k_i}(Zt) \qquad (6)$$

To penalize the impact of crossing a flow, the factor $1 - \cos(\theta_{ki} - \theta)$ gives a cost depending on the angle between the flow and the path direction. The value of $M_{n,k_i}(Zt)$ is doubled in case of moving in opposite direction and ignored in case of moving in the same direction. The sum of the penalties from all the directions determines the cost of moving from $n-1$ to $n$.

In the case where $M_{n,k_i}$ is not defined, the predicted flow grid $M^{pred}_{n,k_i}$ is used.

This leads to the following cost function :

$$g(n) = \|n, (n-1)\| \times (1 + F.flow(n)) + g(n-1) \quad (7)$$

The factor F allows to tune the weight of the perturbation caused by the risk of crossing a flow of humans. Currently, F is tuned experimentally to allow paths going through the flows if directions are compatible (see next section).

## V. Experimental evaluation

### A. Simulation : Adaptation evaluation

We developed a simulator based on PedSim, the 2D simulator of pedestrian crowd proposed by [16]. In addition, [18] proposes a 3D implementation of this simulator under the ROS framework, allowing to add a mobile robot and to teleoperate it. We extended this simulator to provide robot navigation along waypoints, computed with one of the A* based algorithms above.

Figure 3 presents the 'changing' scenario where we evaluate the ability of the different grid models and their relative A* variants to adapt their paths. Here a robot has to go from spot A to spot B, then to return from B to A, and repeat this process. Meanwhile, a crowd of people is following the corridors in the anti-clockwise direction, before to change for the opposite direction at time 500s. Figure 4 shows that

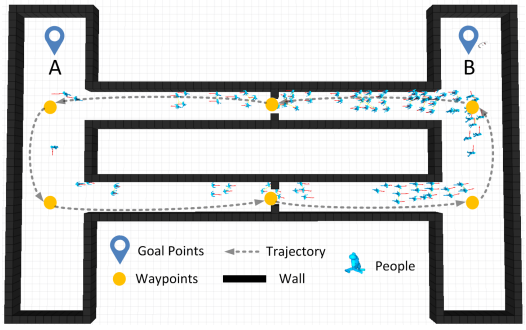


Fig. 3. Snapshot of the changing scenario : change of the Human flow direction after few minutes.

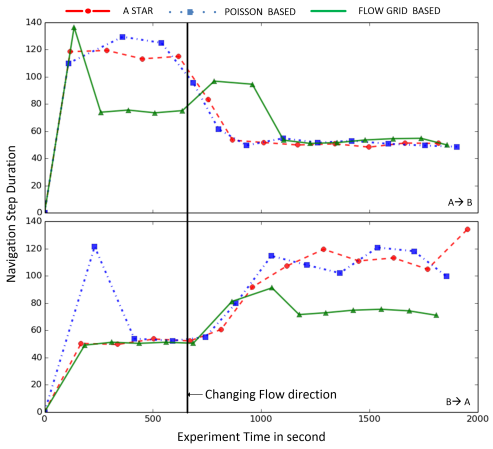


Fig. 4. Time of $A \Leftrightarrow B$ paths, for each A* strategy, in the changing flow direction scenario (Figure 3)

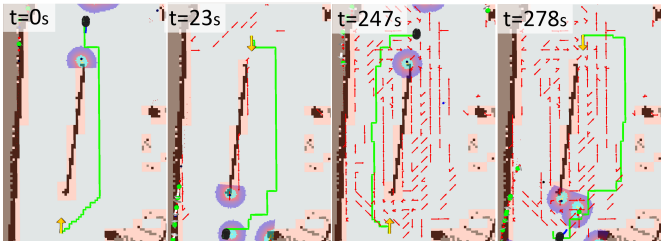


Fig. 5. Snapshots of the experiment showing the robot trajectory evolution (green lines) according to the Flow Grid Map update (red arrows).

before time 500s the Flow-grid A* model is the only one able to learn to use the bottom corridor to go from A to B. The robot follows the human direction then it obtains shorter travel durations (∼78). In the same period all models have equivalent travel durations from B to A, as the human flow use the top corridor with direction B to A. However, when the flow direction change, one can see that only the Flow-grid A* model is able to revise its flow model and then to compute paths going through the bottom corridor (green line, reaching 78 after time 1100s).

### *B. Experiments with Real Robot*

We experimented the Flow grid model and the A* revisited algotihm with a mobile robot evolving among walking persons. We use a Turtlebot 2 platform enhanced with an Rplidar A2 (resolution of 1° angular, omnidirectional and range of 6m). The People Ros package provides a toolbox for detecting human (face detection, leg detection) and estimating their location and velocity. The leg detector was used to detect humans location. After filtering location and velocity data we compute human directions and update the Flow grid Map.

Experiments were conducted in an area of $100m^2$ where a wall separates two connected corridors, see Fig. 5 on left (walls are pink areas). The robot starts on one end of the wall, then it processes a round trip between the two ends of the wall. After a first round trip, three persons begin to walk around the wall into anti-clockwise direction.

**A video of the experiment is available here** [2]**.**

The figure 6 shows the Flow grid map obtained (left side) composed of the observed human flow (red arrows) and the predicted flow (blue arrows). Moreover, the Flow-Grid-A* path-planning of the robot is displayed (green line). At the top right of Figure 6, the Turtlebot point of view shows the planned trajectory, a detected human (legs : red spheres, human location : green sphere and human velocity : white arrow) and the local cost map (used by the local planner to avoid collision). Then at the bottom right, the real robot and human situation is displayed.

The figure 5 shows two round trips of the robot at respectively 0 to 23s and 247 to 278s. After the first round trip, the robot has not met anybody, leading to an empty Flow grid (some false positives occurred close to obstacles). The computed path follows the wall with a safe distance. At time 247s, the robot has detected humans walking around the wall: the flow grid appears clearly as a field of arrows oriented into the anti-clockwise. One can see that the computed path from top to bottom follows the left corridor, which is the direction of the human flow. As it can be seen in the video the robot navigation is not obstructed by the persons and vice versa. The same result is obtained at time 278s, where the robot computes a path across the right corridor, which follows the flow direction.

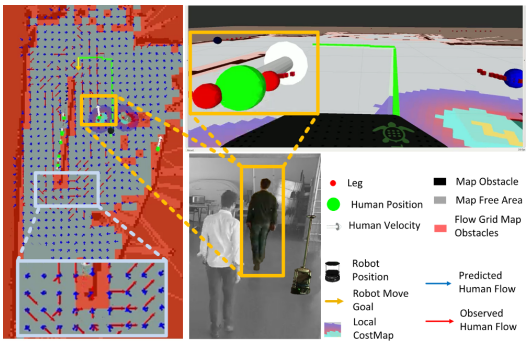


Fig. 6. Example of Flow grid (red arrows) and Prediction Flow grid (blue) obtained with a a real robot (using a Turtlebot 2 base and a Rplidar laser)

[2]https://youtu.be/WiBcO9S_Jmw.

## REFERENCES


[1] G. D. Tipaldi and K. O. Arras, "I want my coffee hot! learning to find people under spatio-temporal constraints." in *ICRA*. IEEE, 2011, pp. 1217–1222.

[2] T. Kucner, J. Saarinen, M. Magnusson, and A. J. Lilienthal, "Conditional transition maps: Learning motion patterns in dynamic environments," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013, pp. 1196–1201.

[3] C. Stachniss, *Robotic Mapping and Exploration*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, ch. Coordinated Multi-Robot Exploration, pp. 43–71.

[4] Z. Wang, P. Jensfelt, and J. Folkesson, "Multi-scale conditional transition map: Modeling spatial-temporal dynamics of human movements with local and long-term correlations," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015, pp. 6244–6251.

[5] T. Wada, Z. Wang, T. Matsuo, Y. Ogawa, Y. Hayashibara, Y. Hirata, and K. Kosuge, "Building human motion map for mobile robot in the indoor dynamic environment," in *Proc. of the 2010 IEEE International Conference on Robotics and Biomimetics*, 2010, pp. 543–548.

[6] T. Wada, Z. Wang, Y. Ogawa, Y. Hirata, and K. Kosuge, "Incremental human motion map system and human walking behavior representation in indoor environment," in *Proc. of the 2012 IEEE International Conference on Robotics and Biomimetics*, 2012, pp. 747–752.

[7] S. O'Callaghan, S. P. N. Singh, A. Alempijevic, and F. T. Ramos, "Learning navigational maps by observing human motion patterns," 2011.

[8] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, ser. Intelligent robotics and autonomous agents. MIT Press, 2005.

[9] T. Nescher and A. Kunz, *Using Head Tracking Data for Robust Short Term Path Prediction of Human Locomotion*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 172–191.

[10] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," in *Computer*, 1989, pp. 46–57.

[11] T. Ikeda, Y. Chigodo, D. Rea, F. Zanlungo, M. Shiomi, and T. Kanda, "Modeling and prediction of pedestrian behavior based on the sub-goal concept." in *Robotics: Science and Systems*, 2012.

[12] M. Seder and I. Petrović, "Dynamic window based approach to mobile robot motion control in the presence of moving obstacles," in *Proceedings of IEEE International Conference on Robotics and Automation - ICRA 2007, Roma, Italy, 10-14 April 2007*, 2007, pp. 1986–1991.

[13] P. E. J. K. V. Mardia, *Directional Statistics*. John Wiley and Sonsons Inc., 2000.

[14] F. Jumel, J. Saraydaryan, and O. Simonin, "Mapping likelihood of encountering humans: application to path planning in crowded environment," in *The European Conference on Mobile Robotics (ECMR)*, ser. Proceedings of ECMR 2017, Paris, France, Sept. 2017. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01588815

[15] S. M. LaValle, *Planning Algorithms*. New York, NY, USA: Cambridge University Press, 2006.

[16] C. Gloor, P. Stucki, and K. Nagel, "Hybrid techniques for pedestrian simulations." in *ACRI*, ser. Lecture Notes in Computer Science, P. M. A. Sloot, B. Chopard, and A. G. Hoekstra, Eds., vol. 3305. Springer, 2004, pp. 581–590.

[17] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Physical Review E*, pp. 4282–4286, 1995.

[18] "Ros packages for pedsim (pedestrian simulator) based on social force model of helbing et. al, https://github.com/srl-freiburg/pedsim_ros." [Online]. Available: https://github.com/srl-freiburg/pedsim_ros