USING COLOURED SNAPSHOTS FOR SHORT-RANGE GUIDANCE IN MOBILE ROBOTS

S. Gourichon*, J.A. Meyer*, P. Pirim**

*AnimatLab – LIP6, 8 rue du Capitaine Scott, 75015 Paris, France,
{stephane.gourichon,jean-arcady.meyer}@lip6.fr
**BEV suite 23 69, 9 Chester Mews, SW1 London, UK, ppirim@club-internet.fr

Abstract

Studies of searching behaviour in bees by Cartwright and Collett led to a computational model of short-range insect guidance that has been successfully implemented on real robots. Still, its reliability depends crucially on arriving at a good match between landmarks observed in the goal place and those in any nearby place within the environment. This paper describes an application of this model in a standard office environment, with unprepared landmarks that may occasionally become invisible or that are easily confused. The corresponding approach calls upon a visual chip that perceives colour and the whole height of the visual field, and upon a matching algorithm that uses colour and proceeds globally, using dynamic programming. Together, they lower the risk of spurious landmark matchings and enhance the performance of the algorithm significantly, allowing it to work without a full 360° panorama and to cope with object disappearance. The performance with respect to the original model of Cartwright and Collett is assessed, both in simulation and in experiments with a real robot. Improvements over previous robotic applications of this model, or its variants, are emphasized. Directions for future improvements are indicated.

Keywords: visual homing, autonomous robot, snapshot model, landmark navigation, vision chip.

1 Introduction

Among the various navigational abilities that were afforded to animals by evolution, short-range guidance is one of the simplest [1, 2]. Basically, this capacity makes it possible to characterize a given place by the relative positions of the natural landmarks that can be seen or detected from it, and to use such information to return to this goal when displaced from it.

To endow mobile robots with similar capacities, engineers typically use artificial landmarks that they place optimally in the surroundings of a given target they want their robots to be capable of reaching. Such is typically the case when a charging station is characterized by infrared beacons, as is the case in [3] for example.

In some circumstances, especially when autonomy is a desired quality in the mobile robots considered, the need to call upon such prepositioned artificial landmarks is a serious drawback. Indeed, an autonomous robot should cope with whatever landmarks are present in an environment it discovers for the first time, and should associate these landmarks with a goal place it might be desirable to reach again in the future, without any human intervention. An obvious possible solution to such a problem is to draw inspiration from the way animals solve it – a specific instance of a more general *biomimetic approach* that is becoming more widespread in robotics [4, 5, 6, 7, 8, 9, 10].

This has been done here, capitalizing upon a guidance model described by Cartwright and Collett [11] that is based on experimental results on visual (or scene-based) homing in bees. Nevertheless, because the objective of this reference model was not to elaborate on the early visual input processing by the bee's nervous system, but rather to produce a simple and effective geometrical homing algorithm based on landmark bearings, its assumptions cannot be directly transposed to a mobile robot navigating in a standard office environment.

To tackle this problem, we used a visual chip allowing the perception of coloured regions, and taking into account the whole height of the visual field. We replaced the original matching algorithm by another, based on dynamic programming, which uses colour and proceeds globally.

In Section 2 we describe the original Cartwright and Collett model, together with some related work directly relevant to our experiments. In Section 3, we describe the modifications we made to this model and how they compare with related approaches. In Section 4, we describe the experiments we have performed and the corresponding results, which are then discussed in Section 5. Several directions for future work are also outlined.

2 Previous work

2.1 Cartwright & Collett

Cartwright & Collett [11] tested five simple computational models affording short-range guidance capacities [1] to a simulated bee. Of these models, the one the most often cited is referred to as "the snapshot model" or "CC model" and assumes that, when the bee encounters an interesting location, it takes a snapshot of the environment. Such a snapshot is represented by a set of thick and thin sectors that correspond respectively to landmarks and to gaps between landmarks in a circular panorama (see Fig. 1). To guide the bee back to that location, the model continuously outputs a movement vector that tends to reduce the discrepancy between the memorized snapshot and the panorama observed in the bee's current location. The bee then flies in the direction indicated by this vector until, hopefully, the length of this vector becomes zero, indicating that the goal is reached.

The movement vector is computed as illustrated on Fig. 1, left, in which the bee's vision system is modelled as a one-dimensional circular retina sensitive to luminosity. First, each region in the snapshot is paired with the nearest region of the same type (landmark or gap) in the current view, assuming that both images are aligned with an external compass direction. Then, a corresponding contribution vector is computed. If the region seen is shorter (respectively larger) than the region memorized, a *radial* unit vector pointing outward (resp. inward) is considered. If the centre of the region seen is on the left (resp. right) of the centre of the region memorized, a *tangential* unit vector is added, which points toward the left (resp. right). The sum of all those individual contribution vectors gives the resultant movement vector that indicates the direction to go. In other versions of this model [12], the sum was transformed into a weighted sum, where radial contributions were scaled by a factor 3 before summing.

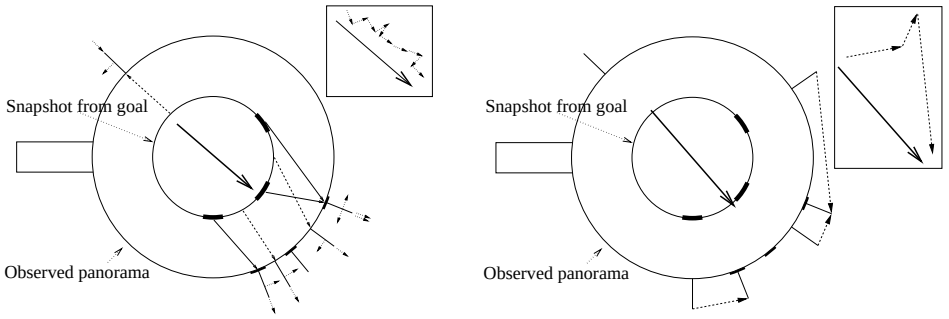

Figure 1: Left: Principle of Cartwright & Collett's CC model, see 2.1 for explanation. Right: the DV variant, see 2.5. The rectangular boxes show the concatenated contribution vectors (dotted lines) next to the resultant movement vector (heavy line).

2.2 Rüchti

In simulation, Rüchti explored a number of issues related to the CC model and its variants, notably in the case where an external compass reference is missing [13]. He also emphasized the difference between a directional and an aligned agent. A directional agent keeps a constant body orientation,

even if the direction of its trajectory varies. An aligned agent keeps a body orientation aligned with its current motion (i.e., the same part of the body is always pointing in the direction of the next move).

2.3 Hong et al.

Although Hong et al. [14] did not mention Cartwright and Collett's work, they implemented the same model. Its input was a 360° panorama provided by a camera pointing up at the bottom of a spherical mirror. The location signature was made up of zones of monotonically increasing or decreasing intensity in a grey-levelled one-dimensional panorama. Fifteen points of maximum intensity change were selected, and a small window around each was considered for "feature matching". This matching was performed with a normalized correlation function using the mean and an approximation of the standard deviation of intensities in the snapshot and current view windows. The derivation of the direction to go called upon a tangential contribution only. The article briefly described a successful run, in which a robot successively reached several different goal places in a hall.

2.4 Röfer

Röfer [15] applied a modified algorithm to the autonomous motion of a wheelchair in an indoor environment. Constraints on the moves of such a four-wheeled vehicle where taken into account. As in Hong et al.'s setting, the input was an image from a camera pointing up at the bottom of a spherical mirror. The location signature was a one-dimensional array of RGB (red, green, blue) values extracted along a circle in the two-dimensional colour image. The matching algorithm relied on optical flow computations, through an iterative process inspired from Kohonen's self-organizing feature maps [16]. The model estimated the wheelchair rotation and the amount of displacement between the current position and the snapshot position. It was also used to direct the robot along a predetermined path.

2.5 Möller et al.

Another application of the CC model to a real-world robot was performed by Möller and his colleagues [17, 18] in the context of a desert landscape with vertical black cylinders used as sharply contrasted artificial landmarks that showed up as dark regions against a bright panorama. A 360° vision field was emulated by a camera pointing up at a conical mirror.

These authors also introduced and discussed a few variants of this model, namely the PV (proportional vector) model, the DV (difference vector) model, and the ALV (average landmark vector) model. The PV model considers, instead of constant length contribution vectors, vectors with lengths that are proportional to the discrepancies they are related to. The DV model, depicted in Fig. 1, right, replaces the tangential contribution vectors by secant vectors that join the memorized and current landmark positions. As for the ALV model, instead of using a snapshot image, it only calls upon the average landmark vector of the goal location. The image-matching procedure is therefore simplified to a subtraction of the average landmark vectors of the current and the goal locations.

2.6 Weber et al.

Weber, Venkatesh and Srinivasan [19] have experimented some alternative matching algorithms to Cartwright and Collett's original PV model, in varied situations (first mathematically idealized point landmarks without occlusions, with both unlimited and limited range of view; then extended circle landmarks with occlusions) along with a thorough analysis and statistical results. They compared the performance of their algorithm with the more computationally expensive warping method of Franz et al. [20].

They also performed some real robot experiments, and compensated for their robot's inability to perceive a full 360° panorama by having it physically turn a complete tour on itself every time a snapshot of the environment is to be obtained.

They conclude that "in its present form, the proposed algorithm is only proficient at homing with respect to a single constellation of landmarks, where there is no significant perception horizon problem".

3 Our approach

The major shortcoming the work just reviewed revealed is the high sensitivity of both the performance and reliability of the corresponding snapshot models to spurious matchings. As our approach focuses on the applicability of these models to the case of a real robot in a standard office environment and because, instead of using a 360° unidimensional vision field, we directly use two-dimensional images from a panning camera (see 3.1), the spurious matching problem becomes an even more crucial issue. In particular, due to the robot's limited vision field, some landmarks may become invisible even when not occluded. As a consequence, the matching procedure may have to cope with sets of landmarks that are not necessarily identical in the snapshot and in the current view.

To remedy such difficulties, two major adjustments have been made to the PV version of the snapshot model:

- to better discriminate landmarks, we call upon as much available visual information as possible. For this purpose, we use a custom biologically-inspired visual perception chip, the GVPP[1], which submits the whole two-dimensional colour visual field to early on-board and real-time processing.
- to minimize spurious matchings and to enlarge the catchment area (i.e., the region in which the robot succeeds in getting to the goal, see [12]), a dynamic programming algorithm (see 3.3) taking into account all coloured regions together is chosen, instead of using a local matching algorithm based upon individually-paired regions that are processed separately.

3.1 Visual perception

GVPP is a biologically inspired visual perception chip that affords important advantages for a robotic implementation [21]. From a video signal, the chip perceives hue, saturation, luminance, and position – in the visual field – together with motion direction and motion velocity – in visual field referential. It also has some shape analysis capabilities. Moreover, the visual input is processed in real time because, instead of buffering images according to the usual frame-grabber approach, the processing is done on-the-fly within the time step of each arriving pixel [22].

3.1.1 Compass reference

To align snapshots and current views, instead of using the robot's compass that produces inaccurate measurements in the office where experiments are performed – due to the presence of many metallic and electrical installations in the surroundings – the movements of the robot are constrained. Thus it is assumed to move like a directional agent with a constant direction, while small unavoidable mechanical deviations are compensated for by direction estimates provided by the robot's built-in odometer.

3.2 Location signature

The views processed by the CC model and several of its sequels are supposed to be made up of successions of dark and bright regions. Although this is relevant when describing what a bee sees in a natural outdoor environment – where bright regions correspond to a sky background, and dark regions correspond to obstacles protruding from the ground [23] – a richer description of the environment is mandatory in the case of a robot in a standard office. This is why location signatures in our experiments are sets of coloured sectors in circular panoramas (e.g. Fig. 2). However, instead of using red, green and blue components (i.e., RGB colorspace) as Röfer did [15],

[1] GVPP is an acronym for Generic Visual Perception Processor (http://gvpp.org/).

we use hue, saturation and luminance (i.e., a HSV colorspace, where V stands for value). This approach makes it easier to recognize individual objects and discriminate between them, because the light coming from an object is never uniform, causing the RGB components that characterize it to accordingly be constantly changing. On the contrary, the measured hue remains nearly constant, thereby providing an excellent means for distinguishing objects. Additionally, we set a threshold on saturation, so that only those objects with enough colour are detected by the system (hue is noisy at low saturation, and undefined for grey objects).

3.3 Matching algorithm

Cartwright & Collett acknowledged the fact that the performance of their model decreases when the number of landmarks increases, because the chances of spurious matchings rise accordingly. Following Einsele & Färber [24], who reported a successful implementation of the dynamic programming paradigm in an application where several sets of laser-range measurements were matched, we apply such an algorithm to snapshot matching. To this end, we adapted an algorithm described in [25], that searches common sub-sequences in two sequences of characters. This entailed designing a function that scored the match between two individual sectors. It is given in (1), where Δ_{hue}, Δ_{sat}, Δ_{lum}, Δ_{az}, represent respectively the differences between the two sectors to be paired with respect to average hue, average saturation, average luminosity, and the azimuth of their centre, and where σ_{hue}, σ_{sat}, σ_{lum}, σ_{az} are the corresponding tolerance parameters:

$$s = \exp\left(-\left(\frac{\Delta_{hue}}{\sigma_{hue}}\right)^2 - \left(\frac{\Delta_{sat}}{\sigma_{sat}}\right)^2 - \left(\frac{\Delta_{lum}}{\sigma_{lum}}\right)^2 - \left(\frac{\Delta_{az}}{\sigma_{az}}\right)^2\right) \quad (1)$$

The output of such a function is 1 when the sectors have equal parameters, and tends toward 0 if they are very dissimilar. Hue, saturation, and luminosity are measured between 0 and 255, angles are in degrees. We use the following tolerance parameters: $\sigma_{hue} = 5$, $\sigma_{sat} = 15$, $\sigma_{lum} = 15$, $\sigma_{az} = 90$. This dynamic programming algorithm yields an overall match that maximizes the sum of individual matches. It has the important property that only one sector in the snapshot can be paired with a given sector in the current view. This algorithm is computationally of order $O(n^2)$ (n the number of landmarks) and thus in the same class as the algorithms tested in [19].

3.4 Direction to go

Once the global match is obtained, we use the PV variant (see 2.5) of the model to obtain a resultant movement vector, except that we choose to take into account only the coloured sectors of the visual field and ignore the gap sectors, as illustrated in Fig. 2.

Moreover, instead of calling upon both tangential and radial contributions to compute this resultant vector, only the former are used. Indeed, several preliminary comparisons indicated that this setting leads to good enough results, without the need to resort to sensitive weighting of both contributions. These choices are discussed further on.

4 Experiments and results

Our test case is a docking situation in which a snapshot is first taken at a charging station, then the robot is moved to some location nearby, from which it has to get back to the charging station. A run is considered successful if the robot physically connects itself in a way that makes recharging possible. The charging station is always behind the robot, so it never sees it and is committed to relying on other visual cues.

We compared the performance of the system in conditions that differed in terms of the perception and matching algorithm used, in order to assess the benefits of perceiving colours and of using the dynamic programming algorithm.

Setup 1 implements the PV model. The panorama is partitioned into object sectors and gap sectors, and no colour information is used. Two objects with two different colours are seen as a single object if no gap separates them. The matching algorithm pairs each object sector in the snapshot with the closest object sector in the current view.

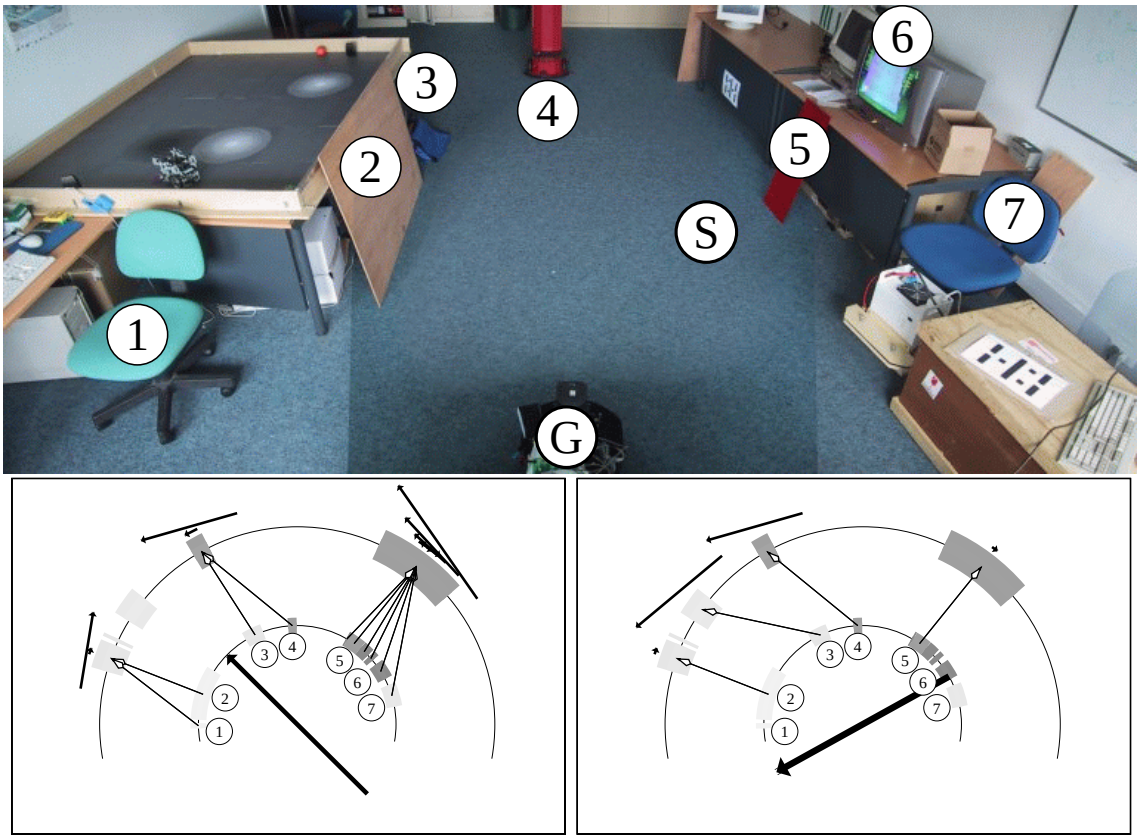

Figure 2: The upper part of this figure describes a standard office environment that has been used in simulation and in real experiments in this paper. Numbered labels show coloured objects that may be used for guidance. The spots labelled G and S indicate respectively the goal place and a start place. The lower part of the figure describes two matching procedures that are commented in section 4. Left and right: the inner arcs represent the snapshot memorized at the goal place, the outer arcs represent the current location signature acquired in the start place. Grey levels represent hue, the most discriminating parameter. Arrows between arcs represent matches. In each case, outers arrows are individual contribution vectors, while the central arrow is the resulting overall direction in which the robot should move. Left: with setup 1 (see section 4), due to spurious matchings, the resulting vector drives the robot to a nearby attractor. Right: with setup 3 (see section 4), colour is taken into account and matching is performed by the dynamic programming algorithm. The robot only matches the wooden board (2), the blue bag (3), the other robot (4), and the red board (5), and the resulting home vector correctly points to the goal place. Thanks to colour information, the robot correctly refrains from matching the green chair on the left (1), the TV screen (6), and the blue chair on the right (7) that are all visible from the goal but not from the current view.

Setup 2 implements the dynamic programming algorithm, but without using colour information.

Setup 3 uses both colour information and the dynamic programming algorithm. Two objects of different colours are distinguished even if no gap separates them.

We compared all three setups in simulation, and setup 1 versus setup 3 on a real robot.

4.1 Simulation

Fig. 3 illustrates the results that were obtained in simulation with setups 1, 2 and 3. In each case the same environment is used, in which a goal location and five landmarks are laid out for short-range guidance. The landmarks labelled a and c are of the same colour, while the two landmarks b near the centre of the arena are of a different colour. The simulated robot has a vision field of 200° and moves as a directional agent, always facing the right of the displayed arena. The figure shows the resulting movement vectors that are computed at many spots in this arena, together with the resulting catchment areas.

The three setups give similar results in the left-hand part of the arena, where all landmarks can be seen by the robot. Differing results are obtained in the right-hand part, where two, or even four, landmarks may become invisible. With the original matching algorithm (setup 1), an undesirable attractor point exists in the centre of the arena. In the right-hand part of the arena, there are fewer landmarks visible in that region than in the goal place. The corresponding matching process accordingly pairs several snapshot sectors with a single sector in the current view. This leads to movement vectors pointing toward landmark c, a point to which we will revert later on. This effect is suppressed in the cases of setups 2 and 3, because the dynamic programming algorithm doesn't allow several sectors of the snapshot to be paired with the same sector in the current view. As only tangential contributions are used, the robot is capable of aligning itself with landmark c but, in the absence of radial contributions, it has no clue about suitable moves toward or away from the goal.

Using dynamic programming without colour (setup 2) doesn't suppress the central attractor, but slightly enlarges the catchment area, whereas the joint use of dynamic programming with colour information in setup 3 enlarges this area still further.

Such results illustrate the benefit of distinguishing landmarks. In setups 1 and 2, the sizes of the catchment areas are limited by the attractor point in the centre of the arena, where the two landmarks b are wrongly paired with the two landmarks a in the snapshot. In setup 3, using colours makes better matching possible. The landmarks b are no longer wrongly paired with the landmarks a in the snapshot, but either correctly paired if visible, or ignored. This enlarges the catchment area.

It should be noted that the original matching algorithm (setup 1) generally cannot perform well with a field of view that does not cover the full 360° range. The different matching algorithms experimented in [19] could not perform as well as the dynamic programming with colour (setup 3), though this is not shown here. The ability to distinguish landmarks using additional information such as colour is necessary to avoid wrongly matching landmarks that appear in only one of the two matched views. This is crucial when the field of view is not 360°-wide.

4.2 Real robot experiments

Real experiments were performed using a Pioneer 2 robot, with on-board perception and computation, and with a panning camera the optical axis of which was 31 cm above ground. This camera was able to scan a panorama ranging from -100 to +100 degrees with respect to the robot's axis. The video images were analyzed continuously by the GVPP chip, and the relevant colour information was sent 25 times per second through a serial port to the computer in the robot.

In these experiments, there are two main reasons for object disappearances: objects seen from the goal may become invisible in other places – because they are behind the robot – and objects may become invisible when the robot comes too close to them – because they are too high to be detected by the camera (e.g., they lie on a desk).

We compared the catchment areas and the success rates of setups 1 and 3. As illustrated in Fig. 4, the catchment area of setup 3 is larger than that of setup 1.

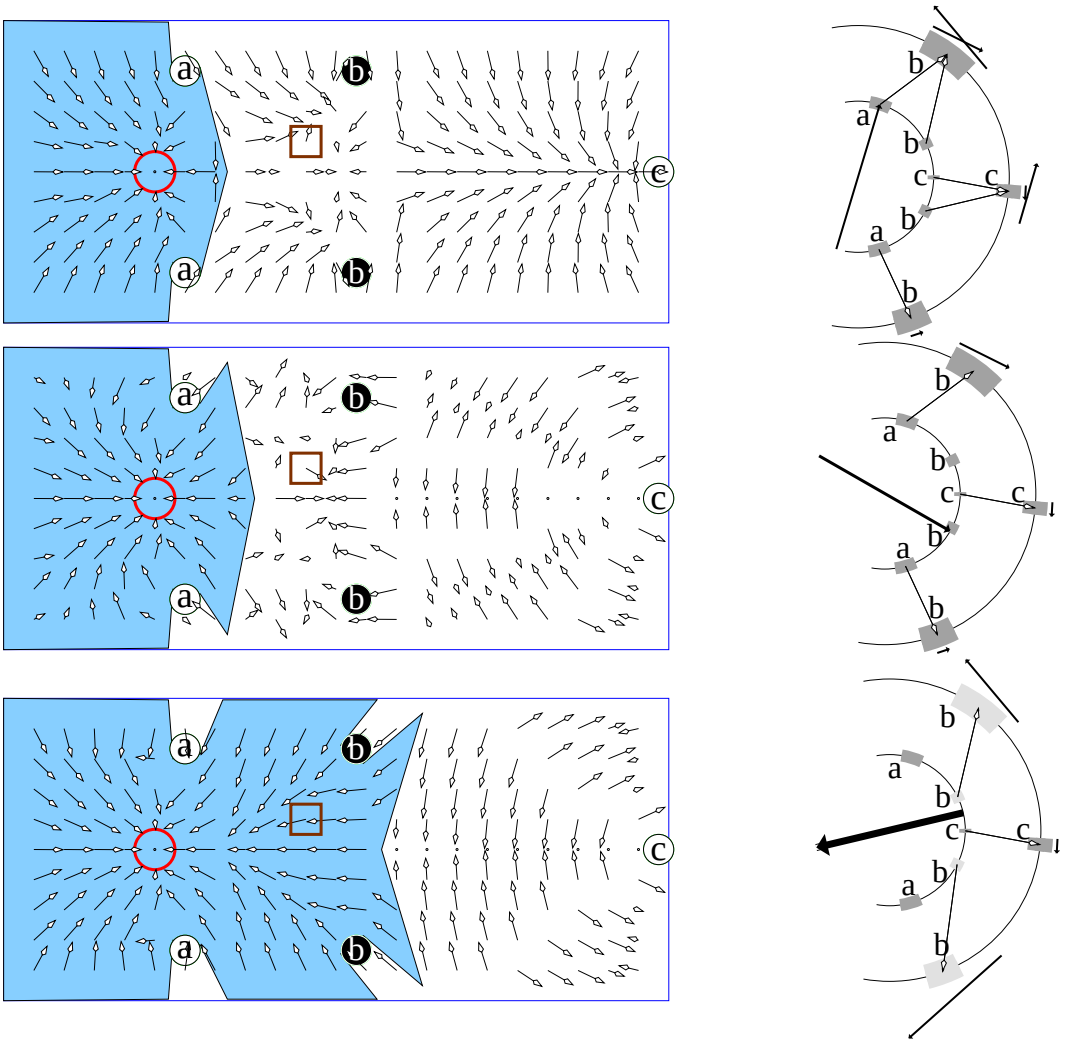

Figure 3: Simulation of the PV model in three experimental setups and comparison of the effects of landmark disappearances caused by the robot's limited angle of vision. Setup 1 (top): the original PV matching. Setup 2 (middle): dynamic programming matching algorithm, without colour information. Setup 3 (bottom): dynamic programming matching algorithm, in conjunction with colour information. Left: Vector fields within the arenas show the movement vectors calculated in specific places by each algorithm. For clarity, we limited the lengths of the vectors. The shaded areas approximately represent the catchment areas. Right: diagrams showing the results of the matching procedures performed in places marked by squares in the arenas on the left part of the figure. The corresponding contribution vectors and the resulting movement vectors are indicated (a thicker central arrow indicates a longer vector).

As in the simulation, the size of the catchment area in setup 1 is limited by the pairing of several snapshot sectors with the same view sector. Consequently, matchings in some places are consistently wrong. Instead of attracting the robot to the goal, where all contributions tend to zero, the model drives it to places where contributions are still important but where their sum is a null vector. We observed such an attractor near the TV and the red board, and another one near the bag and the other robot. It seems that these attractors are responsible for the triangular shape of the catchment area in Fig.4.

The factor limiting the size of the catchment area in setup 3 seems to be colour similarity. The blue chair is visible only near the goal, in the right-hand portion of the robot's view. When the robot moves away from the charging station, the blue door starts becoming visible on its left. The chair and the door have very similar colours so that, when no other information in the scene prevents it, the matching process pairs these landmarks as if they were the same object, thus attracting the robot further from the goal. Because dynamic programming takes the whole scene into account, the presence of other cues in the current view – such as the red board for example – makes it possible to restore proper matching. Thus, when the red board is correctly perceived, the blue door is no longer incorrectly matched with the blue chair.

5 Discussion

The results obtained both in simulation and in robotic experiments clearly indicate that short-range guidance is possible in a standard office environment and that it is considerably improved when the underlying matching procedure takes advantage of coloured signatures and of a global algorithm like dynamic programming.

Previous applications of the Cartwright and Collett's algorithm generally allowed several sectors in the snapshot to be matched with a single sector in the current view. As shown in the above simulations, this tends to attract the robot to this view sector, because of the combined effects of the corresponding contribution vectors. This may be a desirable property in cases where a few landmarks are clustered in an empty open space, which was usually the case in the above-mentioned variations of the snapshot model. In such circumstances, whatever the remote starting place, the resulting movement vector drives the robot closer to the cluster, until the changing distribution of landmarks in the visual field results in a different and more precise matching. In an unprepared environment with natural landmarks, however, the chances of detecting a lonely cluster of landmarks in an open space are small, and matching in the CC model becomes a real issue. Using coloured signatures helps to distinguish landmarks, while using a dynamic programming matching algorithm not only makes the pairing of several sectors of the snapshot with the same sector in the current view impossible, but also reduces the drawbacks caused by disappearing landmarks.

Likewise, using the GVPP chip made it possible to detect objects up to 20° above or below the horizontal plane of the camera, whereas other robotics implementations were blind to any object not crossing the horizon line.

There are several ways of improving the procedure that was used here. In particular, the vertical position of objects in the visual field, together with appropriate shape-recognition algorithms, may be used to better distinguish landmarks. Using a more robust compass could also serve to obviate the displacement constraints that were imposed on our robot, and to help change it from a directional agent to an aligned agent. Even the constraint of using a compass could be relaxed in the future, though in a different way from existing approaches [15, 20]. In particular, we are investigating the possibility of extracting more precise information from angular measurements, which could both improve the calculation of the direction to go and the estimation of the robot's current orientation. Likewise, in the current implementation, the system waits for the camera to mechanically pan the whole panorama before computing a new direction, which takes several seconds. This could be improved because movement vectors may be computed each time the camera points to a new direction.

Only object sectors and tangential contributions to the resulting movement vector were used in the experiments described above. This choice was suggested by the work of Franz et al. [20], who proved that, assuming error-free bearing measurements and matchings, when using tangential contributions each motion step reduces the distance from the goal because the angle between the

true direction to the goal and its estimate is less than 90 degrees. However, as observed by these authors, it is true that the performance of tangential contributions only can be impaired by very non-isotropic landmark setups, an effect that we observed from time to time in our experiments. Therefore, still other improvements might follow from a clever mix of tangential and radial contributions, when the latter will depend upon landmark size information that is easily provided by the GVPP chip.

It is obvious that the matching logic underlying any version of the snapshot model is liable to get into trouble in case of modifications in the environment. However, with the current implementation, it turns out that, if an object disappears (either by removal, or because it passes behind the robot), the dynamic programming algorithm will correctly use additional colour information to avoid wrongly pairing it with it another object, if the resulting situation is not too ambiguous. This is a clear benefit over previous work. However, in the case where a perceived object is moved so that its bearing changes relatively to the robot – without changing the order in which objects are perceived – the corresponding contribution vector is changed. As a result, the length of this vector doesn't equal zero at the goal place, thus driving the robot toward a nearby attractor, and lowering the homing precision. Possible solutions to this problem are under investigation. For example, one can assess the coherence of the set of contribution vectors and ignore matchings that create inconsistencies, taking inspiration from [26]. This would mean that objects with variable positions would be ignored by the navigation system as unreliable landmarks.

Finally, it is worth mentioning that the short-range navigation strategy that was implemented here did not involve dedicated sensors measuring distances, but rather relied on mere angular measurements. There is good reason to believe [27] that similar information could be efficiently used to implement longer-range navigation strategies, which commonly resort to sophisticated sensors like sonars or laser range-finders, and we plan to test such possibilities in the future.

6 Conclusion

The snapshot model of Cartwright and Collett may be used to implement insect strategies of short-range guidance in a mobile robot within a standard office environment, provided appropriate modifications are made to this model. In particular, because such environments offer numerous opportunities for losing sight of the various unprepared landmarks it contains, special care must be devoted to the risks of spurious matchings.

In this paper, such risks were lowered through the use of a dedicated visual chip and of a dynamic programming algorithm that improved the performance of other robotic implementations of the CC model cited in the text. The visual chip allowed perception of coloured objects in the whole visual field (not only the horizon) in real-time. The dynamic programming algorithm for matching proved to be more reliable than local procedures because it performed global matchings and took advantage of the colour information, while staying computationally cheap.

Several possible improvements of our approach to short-range guidance were suggested in the text and will be implemented in the future. Likewise, the potentialities of applying mere angular measurements to higher-level navigation strategies, like those calling upon topological or metrical maps [1], will be the subject of additional research.

Acknowledgements

This work has been partially funded by the interdisciplinary program Robea of the CNRS. S.G. gratefully acknowledges the support of a PhD scholarship granted by DGA.

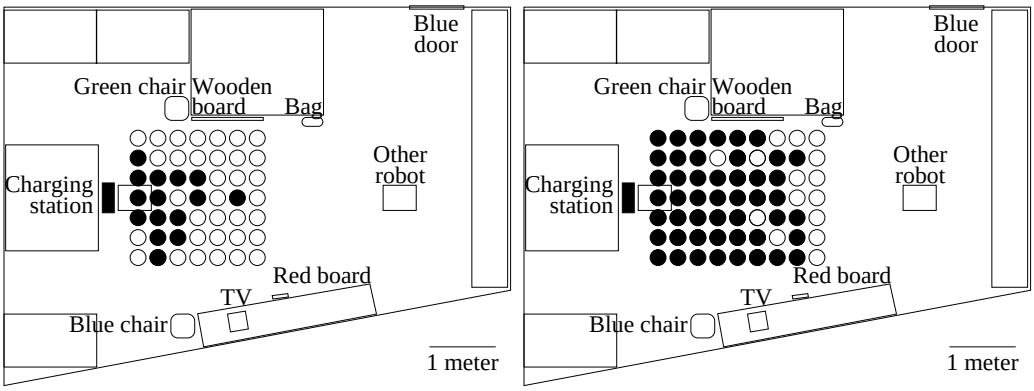

Figure 4: Comparison of catchment areas for setup 1 (left) and setup 3 (right) when experiments are performed on a real robot in the environment of Fig. 2. The robot recorded a snapshot at the charging station (solid black rectangle), then was driven to one of the start places that were aligned on a regular 30-centimetre-wide grid. Its task was to get back to the charging station and to physically connect to it. Solid black circles indicate successful runs, white circles correspond to failed runs. The grid was extended to the right until all runs starting in the same grid column failed.

References

[1] O. Trullier, S. Wiener, A. Berthoz, & J. A. Meyer, Biologically-based artificial navigation systems: Review and prospects, *Progress in Neurobiology*, *51*, 1997, 483–544.

[2] H. A. Mallot & M. O. Franz, Biological approaches to spatial representation – a survey, *Proc. of the 16th International Joint Conference on Articial Intelligence (IJCAI-99)*, in T. Dean (Ed.), Morgan Kaufmann, 1999.

[3] A. Z. Seungjun Oh & K. Taylor, Autonomous battery recharging for indoor mobile robots, *Proceedings of Australian Conference on Robotics and Automation (ACRA2000)*, August 2000.

[4] S. Hirose, *Biologically Inspired Robots: Snake-Like Locomotors and Manipulators.* Oxford Univ. Press., 1993.

[5] G. Bekey, Biologically inspired control of autonomous robots, *Robotics and Autonomous Systems*, *18*(1-2), 1996, 21–31.

[6] R. Beer, R. Quinn, H. Chiel, & R. Ritzmann, Biologically inspired approaches to robotics, *Communications of the ACM*, *40*(3), 1997, 30–38.

[7] N. Sharkey & T. Ziemke, Biorobotics., *Connection Science*, *10*(3), 1998, 361–391.

[8] J. Ayers, J. Davis, & J. Rudolph (Eds.), *Neurotechnology for Biomimetic Robots.* The MIT Press, 2001.

[9] B. Webb & T. Consi (Eds.), *Biorobotics: Methods and Applications.* Cambridge, MA: MIT Press/AAAI Press, 2001.

[10] R. Duro, J. Santos, & M. Grana (Eds.), *Biologically Inspired Robot Behavior Engineering.* Springer Verlag, 2002.

[11] B. A. Cartwright & T. S. Collett, Landmark learning in bees, *J. Comp. Physiol*, *151*, 1983, 521–543.

[12] B. A. Cartwright & T. S. Collett, Landmark maps for honeybees, *Biol. Cybern.*, *57*, 1987, 85–93.

[13] S. Ruchti, *Landmark and compass reference in landmark navigation,* Master's thesis, Universität Zürich, March 2000.

[14] J. Hong, X. Tan, B. Pinette, R. Weiss, & E. Riseman, Image-based homing, *Proc. IEEE Intl. Conf. on Robotics and Automation*, 1991, 620–625.

[15] T. Röfer, Controlling a wheelchair with image-based homing, *Spatial Reasoning in Mobile Robots and Animals, AISB-97 Workshop*, Manchester University, 1997, 66–75.

[16] T. Kohonen, Self-organized formation of topologically correct feature maps, *Biological cybernetics*, *43*, 1982, 59–69.

[17] R. Möller, D. Lambrinos, R. Pfeifer, T. Labhart, & R. Wehner, Modeling ant navigation with an autonomous agent, *From Animals to Animats 5. Proceedings of the Fifth International Conference on Simulation of Adaptive Behavior*, in R. Pfeifer, B. Blumberg, J. Meyer, & S. Wilson (Eds.), The MIT Press/Bradford Books, 1998, 185–194.

[18] D. Lambrinos, R. Möller, T. Labhart, R. Pfeifer, & R. Wehner, A mobile robot employing insect strategies for navigation, *Robotics and Autonomous Systems, special issue: Biomimetic Robots*, *30*, 2000, 39–64.

[19] K. Weber, S. Venkatesh, & M. Srinivasan, Insect-inspired robotic homing, *Adaptive Behavior*, *7*(1), 1999, 65–96.

[20] M. O. Franz, B. Schölkopf, H. A. Mallot, & H. H. Bülthoff, Where did i take that snapshot? scene-based homing by image matching, *Biological Cybernetics*, *79*, 1998, 191–202.

[21] P. Pirim, Le mécanisme de la vision humaine dans le silicium, *Electronique, le mensuel des ingenieurs de conception*, *68*, 1997.

[22] P. Pirim, Method and device for real-time processing of a sequenced data flow, and application of the processing of digital video signals representating. Patent FR 2611063, February 13 1987.

[23] T. Collett & J. Zeil, The selection and use of landmarks by insects., *Orientation and Communication in Arthropods*, in M. Lehrer (Ed.), Basel: Birkhauser, 1997, 41–65.

[24] T. Einsele, Real-time self-localization in unknown indoor environments using a panorama laser range finder, *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS-97)*, 1997, 697–703.

[25] R. Cori & J.-J. Levy, *Algorithmes et programmation*, (Palaiseau, France: École Polytechnique, 1998, ch. 8, 179–183).

[26] M. Betke & K. Gurvits, Mobile robot localization using landmarks, *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2, 1994, 135–142.

[27] F. C. Dyer, Spatial memory and navigation by honeybees on the scale of the foraging range, *The Journal of Experimental Biology*, *199*, 1996, 147–154.

Stéphane Gourichon

Stéphane Gourichon works in the AnimatLab of the Laboratoire d'Informatique de Paris 6. He graduated in 1998 from École Polytechnique, France, and received his M.Sc. degree in artificial intelligence, pattern recognition and applications from University of Paris 6 (DEA IARFA). He is currently a doctoral student at the AnimatLab. His research interests are in the fields of situated artificial intelligence, perception (especially visual), efficient autonomous robots, and adaptive behaviour.

Jean-Arcady Meyer

Jean-Arcady Meyer is currently Research Director at the Centre National de la Recherche Scientifique (CNRS) and head of the AnimatLab in the Laboratoire d'Informatique de Paris 6 (LIP6). He is the founder of the journal Adaptive Behavior and a former Director of the International Society for Adaptive Behavior. He is also on the Founding Board of Directors of the International Society for Artificial Life. He was the main organizer, or co-organizer, of the series of seven international conferences on Simulation of Adaptive Behavior (SAB). He also organized two international summer schools on Adaptive Systems and Simulation. His main scientific interests are the interaction of learning, development, and evolution in adaptive systems, both natural and artificial.


Patrick Pirim

Patrick Pirim is currently chief research engineer at BEV, the company he founded to develop the GVPP chip. He graduated from ENSEA engineering school in 1977 and currently heads the research department at BEV. Since 1986, he has developed 7 industrial silicon chips which mimic biological perception and understanding. His interests are the design of efficient and computationally cheap artificial perception systems.