Integration of Metric Place Relations in a Landmark Graph

Wolfgang Hübner and Hanspeter A. Mallot

Eberhard-Karls University, Tübingen, Germany
`wolfgang.huebner@uni-tuebingen.de`

Abstract. This paper describes a graph embedding procedure which extends the topologic information of a landmark graph with position estimates. The graph is used as an environment map for an autonomous agent, where the graph nodes contain information about places in two different ways: a panoramic image containing the landmark configuration and the estimated recording position. Calculation of the graph embedding is done with a modified "multidimensional scaling" algorithm, which makes use of distances and angles between nodes. It will be shown that especially graph circuits are responsible for preventing the path integration error from unbounded growth. Furthermore a heuristic for the MDS–algorithm is described, which makes this scheme applicable to the exploration of larger environments. The algorithm is tested with an agent building a map of a virtual environment.

1 Introduction

Graphs are an efficient way to code environment maps for autonomous agents. A graph $G = (V, E)$ contains information only about salient places of the environment, represented by the node set V, and the topological structure of stored places coded in the edge set E. Compared with occupancy grids, graphs are a sparse sampled representation of the environment, where the sampling density can be adapted to the environment or the agent's needs. These advantages are paid for with the need for local navigation strategies allowing travel between nodes. The edge set of the graph contains information about routes through the environment which can be traversed by the agent without taking the risk of getting lost. Therefore the agent is only able to perform topological navigation [4].

The behavior of visual homing [2] is based on the ability of the agent to identify a place by the surrounding landmark–configuration. A possible technical way to capture the landmark configuration is a conical mirror vertically mounted above a camera, providing the agent with a 360° view. It has been shown [3] that an array of 72 pixels taken from the horizon line (see figure 1) is sufficient to guide the agent back to a known place from within a certain area around the recording position.

Experiments have shown that the portion of an environment that could be represented with places defined by the landmark configuration is mainly limited

J.R. Dorronsoro (Ed.): ICANN 2002, LNCS 2415, pp. 825–830, 2002.

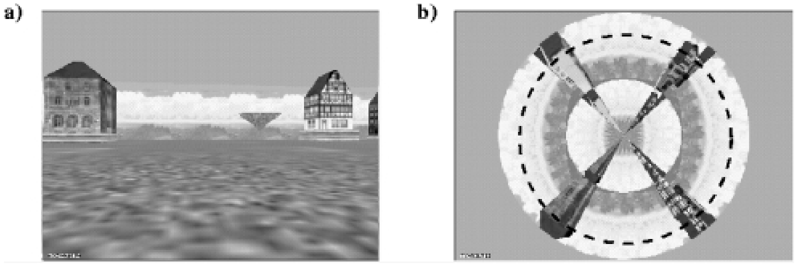

Fig. 1. a) Normal view of the environment. **b)** Panoramic view as seen from underneath the mirror. The dashed circle shows the location of the 72 pixels which are used by the visual–homing procedure.

by the ambiguity of the images. Ambiguous images are not only a result of the minimalistic visual approach. Increasing the resolution of the image may extend the size of the representable area by a certain amount. However this can't be a general solution for arbitrarily large environments.

A possible way to distinguish between similar views seen at different locations is to label the images with the recording position. Additional position information gives the agent the ability to perform survey navigation [4], i.e. the agent is able to find shortcuts apart from learned routes. This article describes a method which extends the topological graph information with metric relations between nodes. The following chapter describes the algorithm which calculates global position estimations from local movement measures obtained by path integration [1]. The last chapter shows results from an agent exploring a virtual environment and draws some conclusions about the map quality.

2 Calculation of the Metric Graph

2.1 Movement in the Graph

An agent which uses visual landmark information and metric relations simultaneously can be modeled by a state vector containing the perceived view ($I_t \in \mathbb{N}^{72}$), the instantaneous position (x_t, y_t) and the instantaneous heading (ϕ_t):

$$S_t = (I_t, x_t, y_t, \phi_t) \tag{1}$$

The change in the state vector after a rotation Θ and a translation of length d is given by:

$$S_{t+1} = (I_{t+1}, x_t + \hat{d}\sin(\phi_t + \hat{\Theta}), y_t + \hat{d}\cos(\phi_t + \hat{\Theta}), \phi_t + \hat{\Theta}) \tag{2}$$

I_{t+1} is the perceived view after the movement is completed. A detailed mathematical description of image changes generated by a moving cone mirror is given

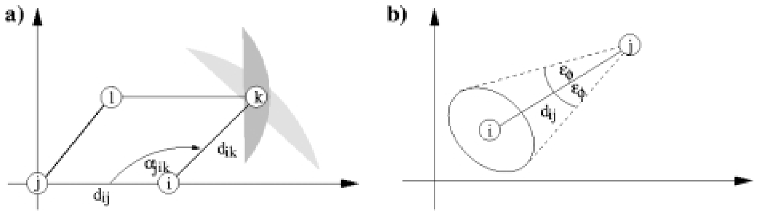

Fig. 2. **a)** Position estimates for k are affected by accumulated errors along the paths {j,i,k} and {j,l,k}. Calculating the position for k with respect to the whole network will bound the position error on k and consequently for all nodes linked to k. **b)** In order to calculate a reliable path from j to i, the error in the position estimate of the target node i must be limited independent from the distance d_{ij}. As a consequence the direction error ϵ_ϕ must decrease proportional to d_{ij}^{-1}

in [3]. The values $\hat{d}$ and $\hat{\Theta}$ are the intended translation and rotation plus a noise term. A node v_i in the graph contains the state vector at recording time S_i.

If the agent returns to an already known place by visual homing, a circuit in the graph is build. Considering the error in the movement measures, it is clear that a direct application of (2) will lead to erroneous position estimates along the path and to contradicting position estimates for the start node (see figure 2a). Instead of calculating path integration along single paths we use a graph–embedding procedure which takes all available routes into account and prevent the accumulation of errors.

2.2 Multidimensional Scaling

The distance between two nodes, or the length of one edge can be calculated from two successive integrator states stored for two nodes v_i and v_j:

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{3}$$

The angle between two edges sharing one common node are calculated from three states:

$$\alpha_{jik} = \pi - \arctan\left(\frac{y_j - y_i}{x_j - x_i}\right) + \arctan\left(\frac{y_k - y_i}{x_k - x_i}\right) \tag{4}$$

Graph embedding is mathematically equal to finding a function $f(V) \rightarrow \mathbb{R}^2$ which assigns a position $\tilde{\boldsymbol{x_i}}$ to the node v_i. The result of the graph embedding is a configuration of points $X = (\tilde{\boldsymbol{x_1}}, \tilde{\boldsymbol{x_2}}, \ldots, \tilde{\boldsymbol{x_n}})$, where pairs or triples of nodes should fulfill the geometrical constraints given by (3) and (4).

Such problems can be solved by *multidimensional scaling* methods [5]. A closed form solution exists if all pairs of distances are known, i.e. if the graph is fully connected [5]. Instead of using all distances, which is impractical for moving agents, we use the additional angle information. The resulting point configuration is unique if the initial integrator state $S_0 = (I_0, 0, 0, 0)$ is used to setup the coordinate system for all position estimates, i.e. $\tilde{\boldsymbol{x_0}} = (0,0)$ and $\tilde{\boldsymbol{x_1}} = (\tilde{x_1}, 0)$. Finding an appropriate point configuration is an optimization problem with two error functions. First a function which describes the mismatch error in the distance judgements:

$$E_d(X) = \sum_{(i,j)|d_{ij} \neq 0} \left(\| \tilde{\boldsymbol{x_j}} - \tilde{\boldsymbol{x_i}} \| - \frac{2 d_{ij}}{\max d_{ij}} \right)^2 \tag{5}$$

and second the error of the angular match[1]:

$$E_\alpha(X) = \sum_{(j,i,k)|\alpha_{j,i,k} \neq 0} \left\| \frac{\tilde{\boldsymbol{x_j}} - \tilde{\boldsymbol{x_i}}}{\| \tilde{\boldsymbol{x_j}} - \tilde{\boldsymbol{x_i}} \|} - R(\alpha_{jik}) \frac{\tilde{\boldsymbol{x_k}} - \tilde{\boldsymbol{x_i}}}{\| \tilde{\boldsymbol{x_k}} - \tilde{\boldsymbol{x_i}} \|} \right\|^2 \tag{6}$$

Finally, both functions are combined over a weighted sum:

$$E(X) = \lambda E_d(X) + (1 - \lambda) E_a(X), \quad \lambda \epsilon [0,1] \tag{7}$$

The weighting parameter λ could be used to compensate systematic errors in the agents path integrator. In the experiments λ is kept constant at 0.5.

Given an arbitrary configuration X, function (6) is limited by $E_\alpha(X) \leq \sum 4$. Contrary to function (6), function (5) has no upper limit, which could make the weighting factor λ obsolete and results in an overfitting of the distance values. This problem is solved by scaling down the measured edge distances by $\frac{1}{2} \max(d_{ij})$ so that the maximal distance equals 2. Selecting a start configuration randomly in a circle around the origin with a radius of $2|V|$ guarantees a solution for which function (5) and (6) are equally accounted. The resulting configuration is scaled up again by $\frac{1}{2} \max(d_{ij})$.

2.3 Application to Large Graphs

The optimization problem defined by (7) has a dimensionality of $2|V| - 3$. With a growing number of nodes a direct application of the MDS–method becomes impractical. Therefore the following heuristic is used to calculate the position estimations in real–time:

1. A subgraph around the current location v_c is selected containing all nodes which are less than ϵ edges away from the current node[2]: $G' \subseteq G$ with $V' = \{v | d(v, v_c) \leq \epsilon\}$.
2. The MDS–procedure embeds G' into a local coordinate system with $\boldsymbol{x_c}' = (0,0)$ as the origin, resulting in a point configuration X' for the subgraph.

---

[1] $R(\alpha_{jik}) \in \mathbb{R}^{2 \times 2}$ is a rotation matrix

[2] Using the graph distance as a selection method is one of many possibilities. It is e.g. also possible to use the smallest circuit containing v_c.

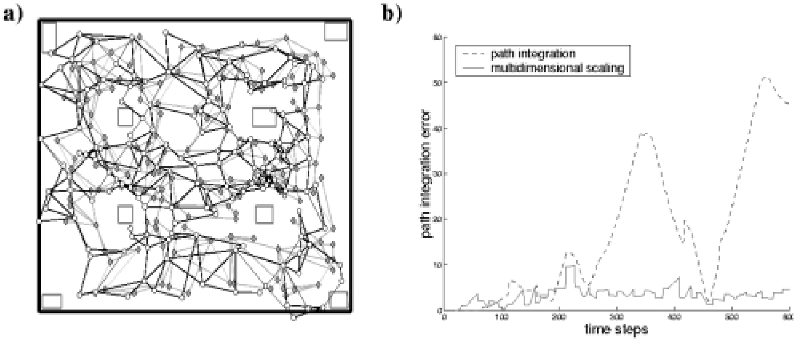

Fig. 3. a) Resulting map. Nodes marked by diamonds indicate the correct positions. Nodes marked by dots are the position estimations calculated with the MDS–method. Landmarks are shown as gray boxes. **b)** Error in the path integrator position over a time interval. Jumps in the mds–curve show points where the path integrator was recalibrated. Decreases in the path integration curve are only random.

3. G' is merged back into the global map G. The coordinate transformation is described by a rotation R and a translation T which minimizes the position difference between the corresponding point sets X and $RX' + T$. For this problem a closed form solution exists (see e.g. [6] for a detailed description).
4. The new position estimates are combined with older ones using a simple time filter in order to make the global map more stable. This becomes necessary especially as the subgraph selection could break graph circuits.

3 Results

For the experiments we use a simulation of a khepera–robot which explores an arena with a size of 180cm × 180cm. The arena contains 8 toyhouses as landmarks enclosed by a textured wall (see figure 1a). The agents path integrator was able to measure rotations and translations with a precision of $\pm 10\%$. The heading of the agent is assumed to be known within the visual resolution of the panoramic image, i.e. $\pm 2.5°$ with 72 pixels (see figure 1b).

The agent follows a simple exploration strategy in order to build a map of the environment. At each time step two distance measures between the instantaneous state S_t and the node states S_i are calculated. First the image similarity $d_I(I_t, I_i) = \max_i(\mathrm{corr}(I_t, I_i))$ with $\mathrm{corr}(I_t, I_i) = \max_n \sum_{m=0}^{71} I_t(n) I_i(m-n)$ and second the metric distance $d_M(\boldsymbol{x_t}, \boldsymbol{x_i}) = \min_i ||\boldsymbol{x_t} - \boldsymbol{y_i}||$. If d_I is below a certain threshold ($d_I \leq 0.98$) and d_M is greater than 5cm a new node is added and the agent selects a new exploration direction by rotating about a fixed angle of 90°. If $d_I \geq 0.98$ and $d_M \leq$ 5cm for the same node, i.e. $\arg\max(\mathrm{corr}(I_t, I_i)) = \arg\min(||\boldsymbol{x_t} - \boldsymbol{y_i}||)$ the agent tries to reach this node by visual homing. After a successful homing trail, a new edge is added. Finally, the position estimates are updated with the MDS–algorithm and the path inte-

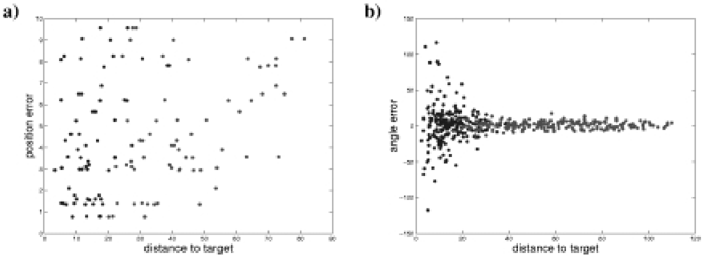

Fig. 4. The plots show that the MDS–method bounds the position error over the whole environment. Therefore the map could be used to plan paths between arbitrary pairs of nodes. **a)** Position error of the target node $\boldsymbol{x_g}$ with respect to the distance from start node $\boldsymbol{x_s}$. **b)** Error in the calculated driving direction ϵ_α with respect to the distance.

grator is recalibrated to the improved position estimation. The resulting graph is shown in figure 3a. The thresholds for d_I and d_M are used to adjust the mesh density so that the recalibration frequency is adapted to the path integrator noise.

The map quality is measured by the agent's ability to navigate reliably over larger distances apart from learned routes. Given a start node v_s and a target node v_g the direction to the goal is calculated by $\alpha = \arctan{(y_s - y_g)/(x_s - x_g)}$. If the error $\boldsymbol{\epsilon_i}$ in the position estimates is bounded for all nodes, i.e. $\forall_i : ||\boldsymbol{\epsilon}_i|| \leq$ *const.* the error in the calculated driving direction must decrease proportional to the distance $\epsilon_\alpha \sim ||\boldsymbol{x_s} - \boldsymbol{x_g}||^{-1}$ (see figure 2b), which is indeed the case for the resulting map (see figure 4a and 4b).

The results show that metrically embedded landmark graphs can be established from noisy data. Such graphs form a powerful representation of space for navigation, detouring, and other tasks in spatial cognition.

References

1. Benhamou S., Séguinot V.: *How to find one's way in the labyrinth of path integration models.* J. theor. Biol. **174** (1995) 463–466
2. Cartwright B.A., Collet T.S.: *Landmark learning in bees.* Journal of Computational Physiology **A 151** (1983) 521–543
3. Franz M.O., Schölkopf B., Mallot H.A., Bülthoff H.: *Where did I take that snapshot? Scenebased homing by image matching.* Biol. Cybernetics **79(3)** (1998) 191–202
4. Kuipers B.: *The spatial semantic hierarchy.* Artifical Intelligence **119** (2000) 191–233
5. Mardia K.V., Kent J.T., Bibby J.M.: *Multivariate Analysis.* Academic Press, Inc., (1982) 413–415
6. Lu F., Milios,E.: *Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans.* Journal of Intelligent and Robotic Systems **18** (1997) 249–275