Multi-knowledge Approach for Mobile Robot Identification of a Changing Environment

QingXiang Wu[12], T.M. McGinnity[1], Girijesh Prasad[1], David Bell[2] and Brendan Glackin[1]

[1] School of Computing and Intelligent Systems, University of Ulster at Magee, Londonderry, BT48 7JL, N.Ireland, UK tm.mcginnity@ulster.ac.uk

[2] Department of Computer Science, Queens University, Belfast, UK q.wu@queens-belfast.ac.uk

Summary. In this paper, a large environment is divided into sub-areas to enable a robot to apply precise localization technology efficiently in real time. Sub-area features are represented in a feature information system so that conventional machine learning or data mining approaches can be applied to identify the sub-areas. However, conventional representations with a single body of knowledge encounter many problems when the sub-area features are changed. In order to deal with changing environments, the multi-knowledge approach is applied to the identification of environments. Multi-knowledge is extracted from a feature information system by means of multiple reducts (feature sets) so that a robot with multi-knowledge is capable of identifying an environment with some changing features. A case-study demonstrates that a robot with multi-knowledge can cope better with the identification of an environment with changing features than conventional single body of knowledge.

1 Introduction

Current robot localization techniques include position tracking techniques [1, 2, 3], global localization techniques [4], Markov localization [5, 6, 7], landmark localization [8, 9, 10], and visual-based localization [11, 12, 13, 14]. These techniques aim to estimate the position of a robot within a given environment by means of sensor data and a map of environment. In order to estimate a robot's position with a high precision, a fine-grained discretization in the state space is usually applied. The spatial resolution is usually between 5 cm and 50 cm and the angular resolution is usually 2 to 5 degrees. For a medium-sized environment of 40 x 40 m^2, an angular grid resolution of 2^o, and a cell size of 10 x 10cm^2 the state space consists of 28,800,000 states. If a map of an environment is very large, localization techniques encounter a high cost in computation, and are very difficult to use in real time. In this paper, a strategy is proposed to address this problem. A large map may be divided into area

H.I. Christensen (Ed.): European Robotics Symposium 2006, STAR 22, pp. 171–180, 2006.

maps, and a robot can apply conventional machine learning approaches to recognize area environment features, for example, in a building, in a room, or in a car park. Therefore, the robot carries out a precise localization in a small area using the specific information on the area environment such as an electronic map. However, conventional single body of knowledge encounters problems in a changing environment. A novel multi-knowledge approach is proposed to solve this problem. In Section 2, an environment feature decision system is used to divide a large map into sub-maps so that conventional machine learning [15] and data mining methods [16, 17] can be applied to identify the environment. The problems with conventional approaches are analyzed in Section 3. In Section 4 the multi-knowledge representation is defined and it is demonstrated that a robot with multi-knowledge has the ability to identify its environment even if some features of the environment changed. The conclusion is given in Section 5.

2 Representation of Environment Features

2.1 Environment Features

In order to illustrate the application of multi-knowledge to robot identification of a changing environment, suppose that there is a robot in a building with six rooms shown in Fig. 1. Rooms from No.1 to No.3 are offices. The area marked 4 is the corridor and rooms No.5 and No.6 are laboratories. Features of the environment are selected from those that can be detected by the sensors in the robot. In this case, the features of the environment are Area, GroundColour, WallColour, CeilingHeight, and Illumination. After the robot had gone through all the rooms in the building, the environment feature information system was obtained. The colours are obtained from a colour camera. Illumination is detected by photosensitive sensors. Distances are detected by distance detectors.

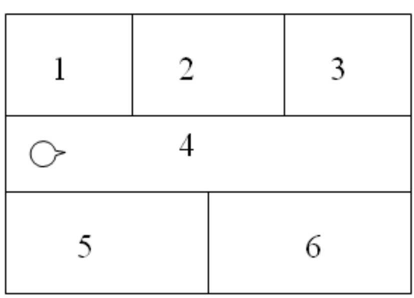

Fig. 1. A robot in a building with six rooms

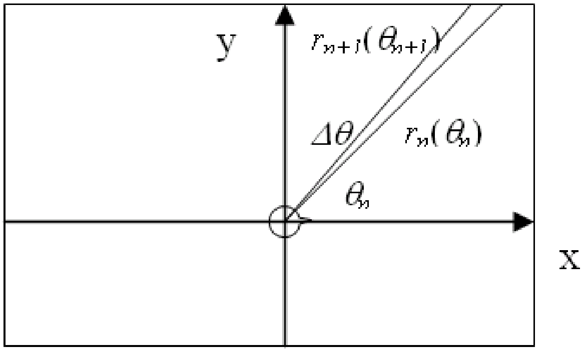

Fig. 2. A robot detects a room area

Area is calculated from the distance data. As shown in Fig. 2, let r(θ) represent distance from the robot to a wall along the direction θ. The distance r(θ) can be detected by distance detectors in the robot. The area can be calculated as follows.

$$s = \frac{1}{2}\int_0^{2\pi} r^2(\theta)d\theta \tag{1}$$

For simplicity, r(θ) is represented by a set of distances r_0, r_1, r_2, ..., r_N with angular interval $\Delta\theta = 2\pi/N$. The area between r_n and r_{n+1} can be calculated by using the small triangle if $\Delta\theta$ is very small.

$$s_n = \frac{1}{2}r_n r_{n+1}\Delta\theta \tag{2}$$

The room area can be calculated by

$$s = \sum_{n=0}^{N-1} s_n = \sum_{n=0}^{N-1} \frac{1}{2}r_n r_{n+1}\Delta\theta \tag{3}$$

where $r_0 = r_N = r(\theta = 0)$. Note that the robot can detect the area with this formula at any point in the room. This is an approximate formula. The error depends on the angular interval $\Delta\theta$ is. The other features can be detected by the corresponding sensors in the robot. An example environment information system is shown in Table 1.

2.2 Environment Feature Instance System

Following [17], let $I =< U, A \cup D >$ represent an *instance system*, where $U =$ u_1, u_2, ... ,u_i, ..., $u_{|U|}$ is a finite non-empty set, called an instance space or universe, and where u_i is called an *instance* in U. $A =$ a_1, a_2, a_3, ..., a_i, ..., $a_{|A|}$, also a finite non-empty set, is a set of *attributes* of the instances, where a_i

Table 1. Room Feature Decision System S:Area,GC:GroundColor, WC:WallColor, CH:CeilingHeight, B:Illumination, R:Room

U	S	GC	WC	CH	B	R
1	9	yellow	yellow	2.6	1	1
2	12	yellow	white	2.6	2	2
3	9	blue	white	2.6	1	3
4	13	gray	white	2.8	2	4
5	15	gray	yellow	3.0	3	5
6	15	yellow	white	3.0	3	6

is an attribute of a given instance. D is a non-empty set of decision attributes, and $A \cap D = 0$.

For every $a \in A$ there is a domain, represented by V_a, and there is a mapping $a(u) : U \rightarrow V_a$ from U into the domain V_a, where $a(u)$ represents the value of attribute a of instance u and is a value in the set V_a.

$$V_a = a(U) = a(u) : u \in U for\ a \in A \tag{4}$$

For a decision system, the domain of decision attribute is represented by

$$V_d = d(U) = d(u) : u \in U for\ d \in D \tag{5}$$

For example, Table 1 is regarded as an environment feature decision system in which the information is detected by a robot that has gone through all the rooms in the building shown in Fig 1. The attribute set A =Area, GroundColor, WallColor, CeilingHeight, Illumination. The decision attribute is D=Room. According to Equation (4), $V_{Area} = 9, 12, 13, 15$. So $|V_{Area}| = 4$. By analogy, we have $|V_{Ground}| = 3, |V_{Wall}| = 2.|V_{Ceiling}| = 2, |V_{Light}| = 3$, and the size of decision attribute domain $|V_{Room}| = 6$.

2.3 Condition Vector Space

The condition vector space, which is generated from attribute domain V_a, is denoted by

$$V_{\times A} = \underset{a \in A}{\times} V_a = V_{a1} \times V_{a2} \times ... \times V_{a|A|} \tag{6}$$

and the size of the space is

$$|V_{\times A}| = \prod_{i=1}^{|A|} |V_{a_i}| \tag{7}$$

For Table 1,

$$|V_{\times A}| = \prod_{i=1}^{|A|} |V_{a_i}| = |V_{Area}| \times |V_{Ground}| \times |V_{Wall}| \times |V_{Ceiling}| \times |V_{Light}| \quad (8)$$

$$|V_{\times A}| = 4 \times 3 \times 2 \times 2 = 48 \quad (9)$$

This means that there are 48 condition vectors in the condition vector space $|V_{\times A}|$. Every condition vector corresponds to a combination of attribute values. Every instance corresponds to a vector in the vector space by its attribute values.

$$\mathbf{A}(u) = (a_1(u), a_2(u), ..., a_{|A|}(u)) \quad (10)$$

For example, $\mathbf{A}(Room1) = (9, yellow, yellow, 2.6, 1)$. If $\mathbf{A}(u) = \mathbf{A}(v)$ for $u \in U$ and $v \in U$, instance u and instance v have the same condition vector. Instance u and instance v are indiscernible. $\mathbf{A}(U)$ represents a set of vectors which exist in the decision system.

$$\mathbf{A}(U) = \{\mathbf{A}(u) \colon u \in U\} \quad (11)$$

If $|\mathbf{A}(U)| = |V_{\times A}|$, the system is called a *complete instance system or complete system*. In the real world, training sets for decision-making or feature classification are rarely completed systems. Table 1 has only 6 existing condition vectors, while the size of condition space is $|V_{\times A}| = 48$. Thus Table 1 is not a complete instance system.

The decision vector space is represented by

$$V_{\times D} = \underset{d \in D}{\times} V_d = V_{d1} \times V_{d2} \times ... \times V_{d|D|} \quad (12)$$

V_d is equivalent to $V_{\times D}$ in cases in which there is only one decision attribute.

2.4 Conventional Knowledge Representations

Knowledge can be represented in many forms such as rules, decision trees, neural networks, Bayesian belief networks and other methods [15]. In general, knowledge can be defined as a mapping from condition vector space to decision space.

$$\varphi_A : V_{\times A} \rightarrow V_{\times D} \quad (13)$$

Let $B \subset A$. We have

$$V_{\times B} = \underset{a \in B}{\times} V_a = V_{a1} \times V_{a2} \times ... \times V_{a|B|} \quad (14)$$

and

$$|V_{\times B}| = \prod_{i=1}^{|B|} |V_{b_i}| \tag{15}$$

$|V_{\times B}|$ is called a subspace of $|V_{\times A}|$. The conventional approach in machine learning or data mining is to select a "best" subset B to get a mapping φ_B based on the subset B.

$$\varphi_B : V_{\times B} \rightarrow V_{\times D} \tag{16}$$

φ_B is then applied instead of φ_A to make decisions or in feature classification. Different mappings can be obtained because the training set is an incomplete system. For example, the decision tree, which is shown in Figure 3, is obtained from Table 1 by means of information entropy.

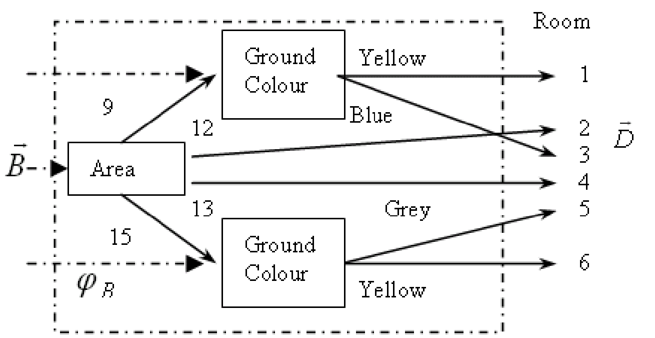

Fig. 3. Knowledge φ_B represented by a decision tree

The advantage of this approach is that it enables a robot to identify the rooms in the building with only two attributes – Area and Ground.

3 Problems with Single Knowledge

A single body of knowledge encounters problems in cases of dealing with changing attributes, incomplete attributes, and unseen instances. Issues arise when using a single body of knowledge in a changing environment. For example suppose that the features of the rooms in Table 1 have been changed. The new features are shown in Table 2.

Using the decision tree in Fig. 3 consider a robot that goes through the rooms with the changed features as shown in Table 2. When the robot enters room 1, it obtains the condition vector (9, yellow, yellow, 2.6, 2). As the decision tree only tests Area and GroundColor, it does not care that Illumination has changed.

Table 2. Changed Room Feature Values S:Area,GC:GroundColor, WC:WallColor, CH:CeilingHeight, B:Illumination, R:Room

U	S	GC	WC	CH	B	R
1	9	yellow	yellow	2.6	2(1)	1
2	12	yellow	white	2.8(2.6)	2	2
3	9	brown(blue)	white	2.9(2.6)	3(1)	3
4	13	blue(gray)	yellow(white)	2.8	2	4
5	14(15)	gray	yellow	3.0	3	5
6	13(15)	yellow	white	3.0	3	6

By using the decision tree, the robot can determine that it is in room 1. For the other instances from U2 to U6, the results are listed as follows.

U2: (12, yellow, white, 2.8, 2) → decision tree → *Room2*.

U3: (9, black, white, 2.9, 3) → decision tree → *Unknown*.

U4: (13, blue, yellow, 2.8, 2) → decision tree → *Room4*.

U5: (14, gray, yellow, 3.0, 3) → decision tree → *Unknown*.

U6: (13, yellow, white, 3.0, 3) → decision tree → *Room4*.

In many cases, the decision tree will give a wrong answer or cannot give any answer. This problem is not only encountered by decision trees, but other methods as well when using single body representations.

4 Environment Identification Based on Multi-knowledge

A multi-knowledge approach can be applied in the machine-learning domain to solve different problems. For example a combination of multi-knowledge and the Bayes Classifier have been used successfully to improve the accuracy of a classification system [18]. Here we apply the concept of multi-knowledge to a robot to identify the changed environment. A single body of knowledge representation usually does not need all attributes in an instance system. For example, the decision tree in Figure 3 contains only two attributes-Area and GroundColor. This attribute set is called a reduct. In general, there are many reducts in a decision system. In a conventional data mining method, firstly, a good reduct(or a set of good features) is selected and then extracted to form a single body of knowledge based on this reduct. The multi-knowledge approach however encourages finding as many reducts as possible. Every reduct can contribute to a single body of knowledge. A set of these single bodies of knowledge is called the multi-knowledge. *Definition:* Given a decision system $I = < U, A \cup D >$. *Multi-knowledge* is defined as

$$\Phi = \{\varphi_B | B \in RED\} \tag{17}$$

where φ_B is a mapping from the condition vector space $V_{\times B}$ to the decision space $V_{\times D}$. RED is a set of reducts from the decision system. Reducts RED can be found by the algorithm in [18]. For Table 1, there are 5 reducts. RED={{Area, Wall}, {Area, Ground}, {Ground, Illumination}, {Ground, Ceiling, Wall}, {Wall, Illumination, Ceiling}}

Applying these 5 reducts, 5 single bodies of knowledge can be obtained. For example, let reduct B={Area, Wall}. The existing condition vector space is

$$\begin{aligned}\mathbf{A}_{\{Area,Wall\}}(U) = \{&(9, yellow), (12, white), (9, white),\\ &(13, yellow), (15, yellow), (15, white)\}\end{aligned} \tag{18}$$

Extracting rules in this condition vector space from Table 1 and generalizing the rules, we have

$$\varphi_{(Area,Wall)} = \begin{cases} Room1 & if(Area, Wall) = (9, yellow) \\ Room2 & if(Area) = (12) \\ Room3 & if(Area, Wall) = (9, white) \\ Room4 & if(Area) = (13) \\ Room5 & if(Area, Wall) = (15, yellow) \\ Room6 & if(Area, Wall) = (15, white) \\ Unknown\ for\ other\ cases & \end{cases} \tag{19}$$

where $\varphi_{(Area,Wall)}$ is presented by using the existing condition vector space $\mathbf{A}_{(Area,Wall)}$ instead of $V_{\times(Area,Wall)}$ because the decision system is not a complete system. $\varphi_{(Area,Ground)}$, $\varphi_{(Ground,Illumination)}$, $\varphi_{(Ground,Ceiling,Wall)}$ and $\varphi_{(Illumination,Ceiling,Wall)}$ can be obtained by analogy. Every instance may get multiple decisions from multi-knowledge. In order to merge these decisions, a decision support degree, which is denoted by $Sup(d_i)$, is defined by a probability as follows.

$$Sup(d_i) = P(d_i | d_i = \varphi_B) \quad for\ \varphi_B \in \Phi \tag{20}$$

where $d_i \in V_d$ is a decision in the decision space, φ_B is a single body of knowledge among the multi-knowledge Φ. The final decision is made by

$$d_{Final} = arg \max_{d_i \in V_d} Sup(d_i) \tag{21}$$

Now consider a robot has multi-knowledge Φ extracted from Table 1 and it enters the changed environment shown in Table 2. For instance U3 in Table 2,

$$\begin{array}{ccccc} \varphi_{(Area,Wall)} & = & \varphi_{(9,white)} & = & Room3 \\ \varphi_{(Area,Ground)} & = & \varphi_{(9,brown)} & = & Unknown \\ \varphi_{(Ground,Illumination)} & = & \varphi_{(brown,3)} & = & Unknown \\ \varphi_{(Ground,Ceiling,Wall)} & = & \varphi_{(brown,2.9,white)} & = & Unknown \\ \varphi_{(Illumination,Ceiling,Wall)} & = & \varphi_{(3,2.9,white)} & = & Unknown \end{array} \tag{22}$$

So Sup(Room1) = 0, Sup(room2) = 0, Sup(Room3) = 1, Sup(Room4) = 0, Sup(Room5) = 0, Sup(Room6) = 0. The final decision is as follows.

$$d_{Final} = arg \max_{d_i \in V_d} Sup(d_i) = Room3 \tag{23}$$

It is possible, according to this algorithm, to calculate d_{Final} for all the instances in Table 2, The results are shown in Table 3. The results show that multi-knowledge can cope with a changing environment much better than any single body of knowledge.

Table 3. Results for multi-knowledge to identify the changed environment

Φ	U1	U2	U3	U4	U5	U6
$\varphi_{(Area,Wall)}$	Room1	Room2	Room3	Room4	Unknown	Unknown
$\varphi_{(Area,Ground)}$	Room1	Room2	Unknown	Room4	Unknown	Unknown
$\varphi_{(Ground,Illum)}$	Room2	Room2	Unknown	Unknown	Room5	Room6
$\varphi_{(Ground,Ceiling,Wall)}$	Room1	Unknown	Unknown	Unknown	Room5	Room6
$\varphi_{(Illum,Ceiling,Wall)}$	Unknown	Unknown	Unknown	Unknown	Room5	Room6
d_{Final}	Room1	Room2	Room3	Room4	Room5	Room6

5 Conclusion

In this paper the problem of a robot identifying its location in a changing environment has been considered. A large environment is divided into small area environments. An instance information system is applied to represent environment features of the sub area environments. Conventional machine leaning approaches are applied to identify sub area environments. The problem for identifying a changing environment was analyzed, where a robot using conventional machine leaning mechanisms finds it difficult to solve the problem. In contrast a multi-knowledge approach was proposed to solve the problem. A case-study was presented to demonstrate that a robot with multi-knowledge copes with a changing environment much better than a conventional single-body knowledge representation.

References

1. Leonard JJ, Durrant-Whyte HF (1991) Mobile robot localization by tracking geometric beacons, IEEE Transactions on Robotics and Automation, 7(3): 376−382.
2. Feng L, Borenstein J, Everett HR (1994) "Where am I?" Sensors and methods for autonomous mobile robot positioning. Technical Report UMMEAM9412, University of Michigan, Ann Arbor, MI.

3. Jensfelt P, Kristensen S (2001) Active global localization for a mobile robot using multiple hypothesis tracking, IEEE Transactions on Robotics and Automation, Volume: 17(5): 748−760.
4. Dissanayake MW, Newman MG, Clark S, Durrant-Whyte, HF, Csorba M (2001) A solution to the simultaneous localization and map building (SLAM) problem. IEEE Transactions on Robotics and Automation, 17(3): 229−241.
5. Fox D, Burgard W, and Thrun S (1999) Markov Localization For Mobile Robots in Dynamic Environments. Journal of Artificial Intelligence Research,11: 391−427.
6. Fox D, Burgard W, Kruppa H, and Thrun S (2000) A probabilistic approach to collaborative multi-robot localization. Autonomous Robots, Special Issue on Heterogeneous Multi-Robot Systems, 8(3): 325−344.
7. Thrun S (2001) A probabilistic online mapping algorithm for teams of mobile robots. International Journal of Robotics Research, 20(5): 335−363.
8. Thrun S, (1998) Bayesian Landmark Learning for Mobile Robot Localization. Machine Learning, 33(1): 41−76.
9. Shimshoni I (2002) On mobile robot localization from landmark bearings, IEEE Transactions on Robotics and Automation, V18(6):971−976.
10. Briechle K, Hanebeck UD (2004), Localization of a mobile robot using relative bearing measurements; IEEE Transactions on Robotics and Automation, 20(1): 36−44.
11. Se S, Lowe D, Little J (2001) Vision-based mobile robot localization and mapping using scale-invariant features. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pp.2051-2058, Seoul, Korea.
12. Niall W, Jos SV (2002) Information Sampling for vision-based robot navigation, Robotics and Autonomous Systems, 41(2-3): 145−159.
13. Ayache N (1991) Artificial Vision for Mobile robots - Stereo-vision and Multisensor Perception, MIT-Press, page 342.
14. Kiyosumi K, Jun M, Yoshiaki S (2002) Autonomous visual navigation of a mobile robot using a human-guided experience, Robotics and Autonomous Systems, 40(2-3): 121−130.
15. Mitchell MT (1997) Machine Learning, McGraw Hill, Co-published by the MIT Press Companies, Inc.
16. Lin TY, Cercone N(eds) (1997) Rough Sets and Data Mining: Analysis for Imprecise Data, Boston, Mass; London : Kluwer Academic.
17. Polkowski L, Tsumoto S, Lin TY (2000) Rough Set Methods and Applications, New Developments in Knowledge Discovery in Information Systems, Physica-Verlag, A Springer-Verlag Company.
18. Wu QX, Bell DA, McGinnity TM (2005) Multi-knowledge for Decision Making, International Journal of Knowledge and Information Systems, Springer-Verlag London Ltd. 7(2): 246−266.