Proceedings of the 2004 IEEE
Conference on Robotics, Automation and Mechatronics
Singapore, 1-3 December, 2004

An approach to Appearance-Based Simultaneous Localization and Map Building (SLAM)

Chun-Fan Lee and Arcot Sowmya
ARC Center of Excellence in Autonomous Systems (CAS)
School of Computer Science and Engineering
University of New South Wales
Sydney, Australia
cflee@cse.unsw.edu.au, sowmya@cse.unsw.edu

Abstract—**Current robotic localization and SLAM algorithms are restricted to simple geometric features such as lines and corners as landmarks. The richness of the information provided by visual perception has not been fully explored. This paper presents a vision based SLAM algorithm which utilizes visual information with minimal prior assumptions.**

I. Introduction

Over the past twenty years, building an autonomous mobile robot which is able to interact intelligently with the physical world is the goal for many researchers. Robotic mapping focuses on the problems of modeling the physical environment, sensor data acquisition and action planning. The robotic mapping problem is commonly known as Simultaneous Localization and Mapping or SLAM for short. Humans perceive 70% of world information through visual perception. From an image we can tell an object's color, texture, shape, size etc. Electronic visual sensors are robust, inexpensive, have high refresh rate and widely available. All these factors have made the use of vision in mobile robots very attractive. The localization technique that depends on visual perception is said to be appearance-based. Unlike many Extended Kalman Filter (EKF) tracking based localization techniques [4, 9, 10], landmarks do not need to be defined explicitly in the appearance-based approach, therefore there are fewer assumptions made about the environment and consequently the system will be more portable and robust.

II. Appearance Based Localization

Appearance-based systems attempt to provide "appearance to action" or "appearance to location" [2] mapping of the environment. The "appearance to location" type of navigation maps each appearance to a location in the global coordinate frame. The "appearance to action" type usually represents the environment with a view-sequence [1] structure. When localizing, the robot compares the current observation against the list of previous observations.

Matsumoto et. al. [1] proposed a model called the "View-Sequenced Route Representation". A sequence of images and the in-between frame action are memorized successively along a route. The current observation is compared with the view sequence to find the best matched frame. The robot navigates by following the view sequence; steering angle is adjusted to minimize the difference between the current observation and the expected observation. Generally speaking, a view-sequence represents a navigational path; it gives information about how to get from any point along the path to any other point on the path. Existing appearance-based methods require the map to be given a priori; or contain a map learning phase where the environment is "shown" to the robot by a tour guide. Therefore current appearance-based algorithms can only be considered as navigation and localization techniques, rather than true SLAM algorithms.

III. Localization as POMDP

There exist many localization techniques but what makes them different is the way in which they represent uncertainty in the localization process. The underlying framework universal for all of them can be seen as a Partially Observable Markov Decision Process (POMDP) [3]. The POMDP models the localization process in terms of states and probability which provides a theoretical background to analyze the problem mathematically. In order to understand the fundamental properties of the localization process, it is important to understand the POMDP.

A POMDP consists of a set of states $\mathbf{x} = \{x_0 \dots x_k\}$, observations $\mathbf{z} = \{z_0 \dots z_k\}$ and actions $\mathbf{u} = \{u_0 \dots u_k\}$ where the subscript denotes the time index. Each state contains a set of variables that we wish to model, such as the robot pose. Given $\mathbf{z}$ and $\mathbf{u}$, we calculate the belief state $\theta(x_t)$, which is defined to be $\theta(x_t) = P(x_t \mid \mathbf{z}, \mathbf{u})$. A recursive formula for calculating the latest belief state consists of three stages:

A. Prediction stage

In the prediction stage, a probabilistic model, which is called the transitional probability is built to model the error involved in robot movements:

$$P(x_{t+1} \mid x_t, u_t) \quad (1)$$

where x_t and u_t is the state vector and action executed at time t respectively. The prediction stage gives an estimate of the robot position after executing action u_t.

B. *Observation stage*

An observation of the true position is made in the observation stage. The current observation is compared with the previous observations to give the observation probabilities:

$$\forall x_k, P(z_{t+1} \mid x_k) \quad (2)$$

The observation probability calculates the likelihood of observing the current observation z_{t+1} from robot state x_k.

C. *Update stage*

The estimation from the prediction stage is combined with the observation probabilities through the Chapman-Kolmogorov equation:

$$\theta(x_{t+1}) = \alpha P(z_{t+1} \mid x_t) \int P(x_{t+1} \mid x_t, u_t)\theta(x_t)dx_t \quad (3)$$

where α is a normalizer. This equation shows how the belief state can be calculated recursively by incorporating the transitional and observation probabilities.

IV. Appearance Based SLAM

Equation 3 was derived from probability principles, suggesting how we should formulate the problem and integrate information together. The equation consists of three parts, the previous state estimate, the transitional probability and the observation probability.

In our proposed algorithm, the environment is represented by a set of states and links. Each state represents a location in the environment, and stored with each state is the observation made at that location. States are connected together by links, where each link represents the inter-relation and traversability between two states.

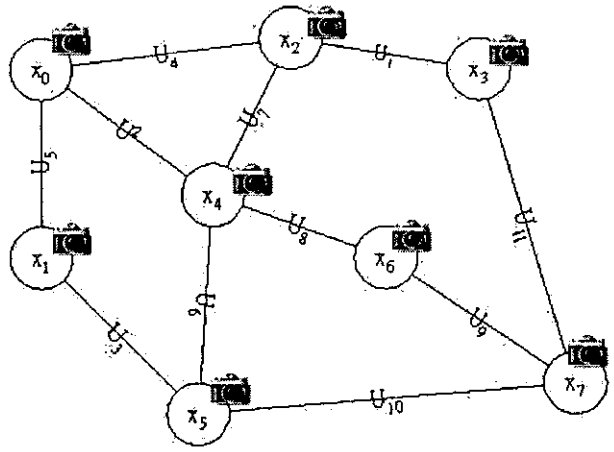

Figure 1 - Environment model. x = State, U = link, camera = snap shot taken at state.

Unlike many current approaches based on distance tracking, the algorithm does not reference a single global coordinate frame. Building a map in one global coordinate frame is an inherently difficult task. It requires precise measurements and all measurements made at different times and locations must be consistent with each other. In contrast to robotics navigation, humans tend to remember the surrounding features of a particular location and the local inter-relation between different locations [8]. We can generalize our daily experience into a number of hypotheses:

It is not necessary to reference all the information in a single coordinate frame.

We can relate different pieces of information together by local inter-relationships

An implicit map representation containing only local location information and inter-relationships between all locations is sufficient to guide us to go from one location to the other, provided there exists a path within the inter-relationships between the two locations.

The technique closest to such a navigation technique to this biologically inspired SLAM strategy is the Appearance-based navigation. An early attempt in using the "appearance to location" model revealed the difficulty in using the POMDP (Kalman filter or similar) model along with appearance-based technique as a SLAM solution. Existing landmark based SLAM algorithms update the robot position by triangulation, the robot position can be accurately pin-pointed within a few iterations if providing the landmarks, and the observations are accurate. This is not the case for appearance-based methods. Since snap-shots are location sensitive, it is inaccurate to calculate observation probability from previous states using nearest-neighbor principle. If the robot is estimated to be at some unvisited position which cannot be verified this with the current observation, then we can only assume that the robot has the possibility to be at the current location. The important consequence is that if unvisited locations cannot be invalidate from the observation probability, the error of the robot position converges slowly while the divergence rate generated

from the motion model quickly overruns the convergence rate from observations. Therefore using a global reference frame for locational references in conjunction with appearance-based SLAM strategies is unlikely to succeed.

V. From Localization to SLAM

In our new approach, state estimation is done purely based on appearance comparison and transition estimation. Let us first define a few terms: a link is an action or state transition which brings the robot goes from one state to another upon execution and a path is a sequence of consecutive states. The algorithm consists of three stages:

A. *Prediction stage*

The algorithm keeps track of a set of candidate paths (path hypotheses). Each path is a hypothesis of where the robot has traveled up to current time t. When the robot is uncertain about its current position, each candidate path will splits into a number of children paths, with each covering one possibility leading from the candidate path (the parent path).

After execution of action u_t, the robot reaches some unknown state. There are two possibilities; the unknown state can either be an unvisited state or a state which has already been visited. For each possibility, a link is created to connect from the end of the candidate paths to the unknown state. The executed action model u_t which is equivalent to the transitional probability function in POMDP is stored with the link. See Figure 2, this is hypotheses generation and calculating probabilities of each happening.

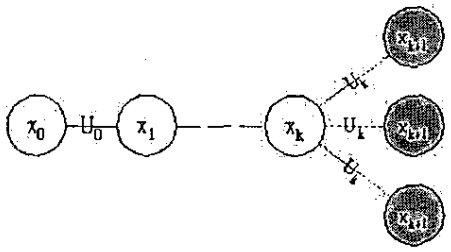

Figure 2 - The path from state X_0 to X_k splits into three children paths after action u_k is executed.

The transitional probability is used to estimate the possibility of such a transition having been taken. It is always possible that the current position corresponds to an unvisited state; therefore a path will always be generated for the hypothesis of the unknown state being a new state. It is also possible that the current observation matches with a number of previous states (perceptual aliasing). In this case, the transitional probability serves as a constraint so that new links are split only for those states that are possible to reach from the state after executing the current action.

In the case of the unknown state being a previous state, the transitional probability $P(x_{prev} \mid x_{cur}, u_{cur})$ is calculated by tracing forward in time from the previous state x_{prev} towards to the current state x_{cur}, and all the sequence of actions leading from x_{prev} to x_{cur} is used in the calculation.

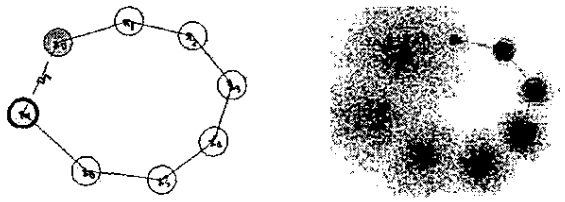

Figure 3- (left) Transitional probability calculation. Thick circle: current state, gray circle: previous state. (right) the estimate location after each action.

Figure 3 shows the location estimation overlaps the previous state; this suggests that the current transition is possible.

B. *Observation stage*

An observation z_{t+1} of the current state is made. The observation probability is calculated by comparing the current observation against the previous observations. In our implementation, we employ the gradient histogram [6] as our feature. For each snap-shot, the gradient value is calculated with the Sobel operator; the values range from $[-\pi, \pi]$ are discretized into k bins. Two observations are compared by comparing their gradient histogram using the χ^2 distance measure:

$$\chi^2(h_i, h_j) = \sum_k \frac{(h_i(k) - h_j(k))^2}{h_i(k) + h_j(k)} \tag{4}$$

where h_i and h_j are the two histograms respectively. The χ^2 distance is a measure of overall fit of the model to the data commonly used in statistical analysis. The observation probability is calculated by taking the inverse of the χ^2 distance:

$$P(z \mid x) = \frac{1}{(\chi^2)^\beta} \tag{5}$$

Beta is a constant to adjust the sensitivity of the denominator term.

C. *Update stage*

The observation probability is augmented with the transitional probability to give the updated belief state value.

Since each candidate path forms a set of children paths by splitting itself independent from other path hypotheses, the belief states of the children are only dependent on its parent. The belief state update equation becomes:

$$\theta(x_{t+1}) = \alpha P(z_{t+1}) P(x_{t+1} \mid x_t, u_t) \theta(x_t) \quad (6)$$

α is a normalizer to ensure the belief states sum up to 1 after update.

The following is the pseudocode of the Appearance-based SLAM algorithm:

```
For each path hypothesis, p {
Create new state x, link from p to x
For each existing state in p {
Calculate transitional probability
Calculate observation probability
Update belief state
}
```

Figure 4- Pseudocode of the Appearance-based SLAM algorithm

A tree structure will result from this process. Each branch split represents an uncertainty, a path from the root node to any leaf node is a possible robot path, the number of leaf nodes corresponds to the number of path hypotheses.

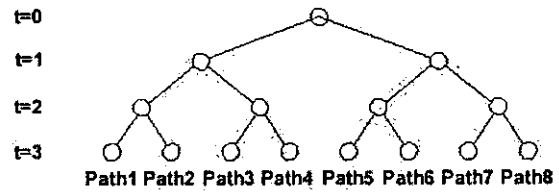

Figure 5- Structure of path hypotheses.

The number of path hypotheses will keep increasing upon uncertainty unless we can determine which of the paths are invalid and hence remove them. One possible path hypotheses invalidation scheme is to remove any path for which the current state estimation is less than a particular confidence threshold. As more information gathers in the future, we can hopefully be able to distinguish the true path from the false ones.

Similar to the multiple hypotheses tracking approach [7], our algorithm keeps track of a set of path hypotheses. The set of hypotheses are expanded only when additional coverage is needed, therefore the resource required is kept minimal. Hypotheses which have confidence less than a certain threshold are removed.

VI. Experimental Results

The algorithm was tested in a simulated environment generated by an OpenGL simulator. The simulator performs a walkthrough of a 3D indoor environment. The camera view is set to have an angle of view of 60°. Figure 12 shows a number of sample robot observations.

The algorithm starts by making an observation of the initial position z_0. After each action u_t (forward 100units, right/left turn 90°), an observation is made at the current position. 3% error is injected to each action. The robot walks in a cyclic path to test its ability to re-localize itself with respect to the previously visited states. The path is 6000 units long in total. The accuracy of the algorithm is measured by the average error between the true robot position and the path hypotheses. Since we are testing on the algorithm's re-localization ability, the hypothesis for the current position being a new state will be excluded from the measurements. Figure 6 and Figure 7 shows the robot navigation path and the actual robot path with 3% motion error respectively.

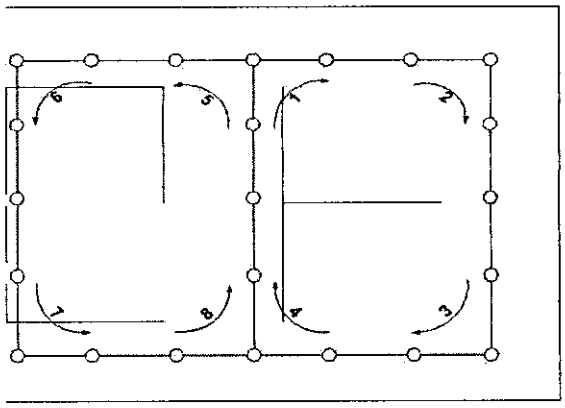

Figure 6 Robot navigation path.

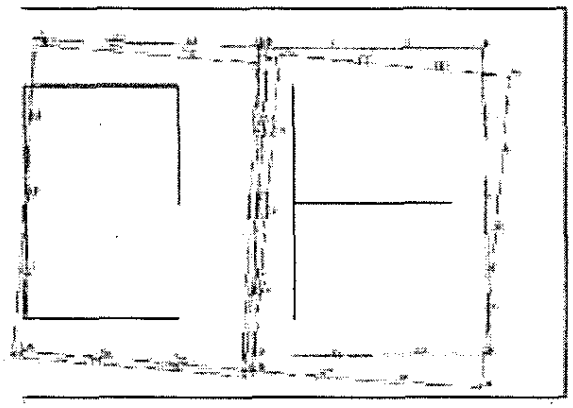

Figure 7 Actual robot navigation path.

Figure 8 shows the number of path hypotheses being tracked during the navigation, the robot starts the second cycle at t=36 and localizes itself again at t=50 and t=72. The error measurements (Figure 9, Figure 10, Figure 11) showing the path hypotheses are kept close to the true robot position. Although there can be a number of hypotheses existing at the same time, the low error rate indicates that the hypotheses do agree on the current robot position. This is an important indicator when we use the resulting map to perform path planning. As long as the hypotheses do agree upon a certain action, knowing a rough estimate of the current robot position is sufficient to decide on the action reliably.

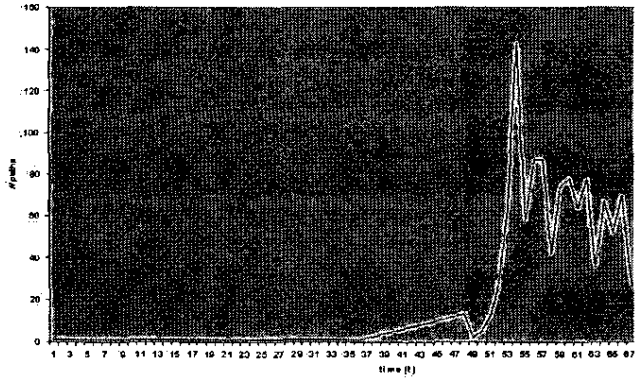

Figure 8 - Number of path hypotheses during the navigation.

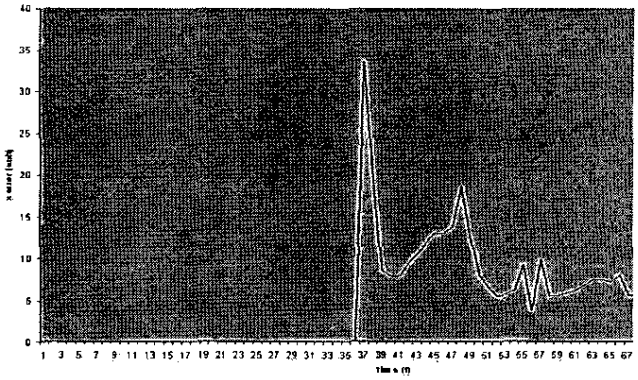

Figure 9- Error in the x direction.

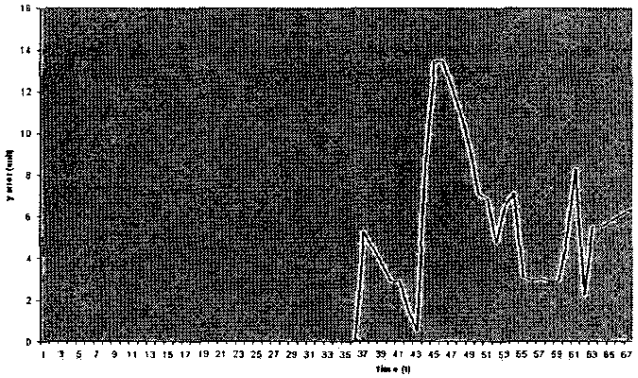

Figure 10 - Error in the y direction.

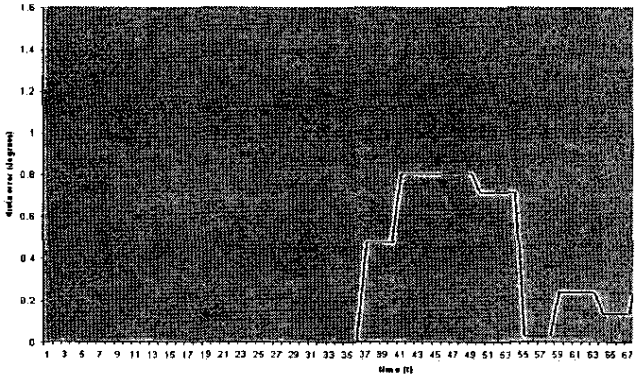

Figure 11- Error of the heading direction.

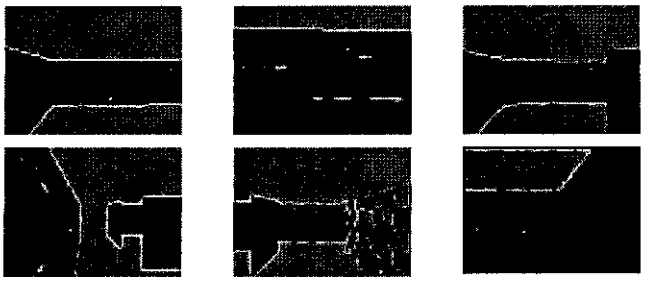

Figure 12- Sample snap-shots of the simulated environment.

VII. CONCLUSION

In this paper we have presented a possible approach to perform SLAM using appearance-based representations. Existing appearance-based methods only addresses the localization problem, where the system requires a map to be given a priori. We identified the short comings of reference in a single coordinate frame and the proposed algorithm relies on observations and their inter-relations, forming a network of observations. Experimental results show that the algorithm is able to re-localize to the existing states successfully.

REFERENCES

[1] Matsumoto, Y., Inaba, M. and Inoue, H. Visual Navigation using View-Sequenced Route Representation, In Proc. Of IEEE Int. Conf. on Robotics and Automation (ICRA 96), pp.83-88, 1996.

[2] Krose, B.J.A., Vlassis, N., Bunschoten, R. and Motomura, Y. A probabilistic model for appearance-based robot localization, Image Vision Comput. 19(6), pp.381-391, 2001

[3] Fox, D., Burgard, W. and Thrun, S. Markov Localization for Mobile Robots in Dynamic Environments, Journal of Artificial Intelligence Research (11) pp.391-427, 1999

[4] DeSouza, G. and Kak, A. Vision for Mobile Robot Navigation: A Survey, IEEE Trans. PAMI, Vol 24. No. 2, 2002

[5] Thrun, S. Particle Filters in Robotics, Uncertainty in AI (UAI) 2002, pp.511-518.

[6] Kosecka, J. Zhou, L., Barber, P. and Duric, Z. Qualitative Image Based Localization in Indoors Environments, Proc. CVPR Vol. 2, pp. 3-8, 2003

[7] Arras, K., Castellanos, J., Schilt, M., Sieqwart, R. Towards Feature-Based Multi-Hypothesis Localization and Tracking 4th European Workshop on Advanced Mobile Robots, Lund, Sweden, Sept. 2001.

[8] Biederman, I. Recognition by components: A theory of human image understanding. Psychological Review 94: 115-147, 1987.

[9] Clark S., Dissanayake G., Newman P. and Durrant-Whyte H., A Solution to Simultaneous Localization and Map Building (SLAM) Problem, IEEE Transaction of Robotics and Automation, Vol 17, No 3. pp. 229-241, 2001.

[10] Thrun S., Koller D., Ghahramani Z., Durrant-Whyte H. and Ng A.Y., Simultaneous mapping and localization with sparse extended information filters, Proc. WAFR, 2002.