Self Localization Using SeeAsYou Categorization System

Alexander Kazeka and Matthew St. John

Abstract

In this paper we prevent a novel robot localization technique using the TF*IDF metric coupled with an existing thalamocortical categorizer in the SeeAsYou vision system. We generate scores for all known locations with the correct location expected to receive the highest score. As we will show, this technique works very well with over a 90% success rate and can be expanded to a more complete SLAM system.

1. Introduction

Simultaneous Localization And Mapping, or SLAM, is one of the fundamental problems of robotics. Self-Localization describes the ability of an entity, human or robotic, to determine where it is located, either relative to known landmarks or even a complete map. Self-Localization requires sufficient information about the environment surrounding the entity, so it must be capable of efficiently gathering this information about locations in a manner that it can recall at a later time, the Mapping component of SLAM. Human beings have evolved this behavior, so humans design their world to utilize this skill. If robots are to interact with the human world without assistance, they must also be capable of SLAM. For our project, we have only implemented Self-Localization due to time constraints.

There are primarily two methods of tracking and storing information about the environment for SLAM. The first is to provide an exact map of the environment and a measuring instrument capable of gathering information at a matching precision level. For example, a topographical map combined with a GPS system can work very well for outdoor environments [1], although this system will of course be ignorant to its immediate surroundings. Such systems are also ill-suited to indoor use due to obstacles interference with the GPS system, but a matching level of precision relative to the size of the environment can be achieved through the use a laser rangefinder or sonar system and a three dimensional map of the rooms to be navigated through [2] [3]. Both of these systems provide very high levels of precision, but lack robustness in a dynamic environment. If an object is moved to a new position, the resulting map as plotted by the rangefinder or sonar system is altered dramatically. Moreover, sonar systems and most rangefinders are incapable of detecting a breakpoint between objects. A chair leg appears to be the same object as the floor on which is stands because the sensors used return an unbroken connection between them. We need a means of tracking the environment as a collection of objects, not a single continuous one.

A more robust system can be implemented through mapless SLAM. Cameras are usually the sensor of choice for these systems, both because they are well suited to differentiating between contiguous objects and because they are already present on robots that are to be used with human beings. Advanced laser rangefinders can also be used to gather both range data and grey-scale illumination data, but are vastly more expensive and have a lower resolution. Using a collection of images, either still snapshots or screengrabs from a video, and a segmentation technique, components of the environment deemed important can be stored. When the robot revisits the location, it will again locate important components of the environment and then match them with its database of the environment, providing it with a location [5]. Mapless SLAM generally cannot provide the same level of precision as the previous mechanisms due to the reduced precision of the instruments used and data stored, but a general localization, such as at the room-level of precision in an indoor environment, can be sufficient for the task.

The means in which the information about the environment can be stored varies quite a bit. A general description of the dominant properties of the environment can be used [4], but these properties are a sum of the entire environment, which is close to the whole-room perspective that sonar uses, and thus lacks robustness to a dynamic environment. Thus, the preferred method is generally to store the objects separately and probabilistically determine the current location based on the objects found [5]. Tracking them as landmarks can even allow for relative positioning, regaining some of the precision localization granted by more exact instruments [6].

The rest of the paper is structured as follows. Section 2 provides background information for our localization technique, which we then describe in section 3 along with the experiment design. Our results are summarized in section 4. We cover both our conclusions and potential expansions to the technique in section 5. At the end of the paper following the references we list various other sources we consulted while designing the system but were not included as references in this paper.

2. Background Information

We have implemented our Self-Localization technique using a modular robotic vision system called SeeAsYou, under development at Colorado State University. New components are added as one or more modules to the program, allowing unnecessary components to be disabled when they aren't needed.

To determine significant features in each image, we used a existing precategorizer module from SeeAsYou. The precategorizer first selects attention windows based on saliency as defined by intensity, sharp color changes, or changes in edge orientation. Feature vectors defining these properties are then extracted from the attention windows. Finally, the feature vectors are used to train a neural network so that attention windows that share similar feature vectors can be grouped together into categories. The output is both a collection of the feature categories found in the scene and the number of instances found. The precategorizer proved very reliable, and later in our testing we added a heuristic to bring in any outlier categories which had been miscategorized into a separate group for even better performance [7].

To assign weight to categories unique to a specific location, we used a system called Term Frequency Inverse Document Frequency, or TF*IDF. Although the technique is designed for information retrieval from text sources[8], the same technique is easily applied to searching for categories within an image[9]. TF rates the importance of a category to a location based on the ratio of the number of occurrences of the category at the location versus the number of occurrences of any category at that location.

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

where $n_{i,j}$ is the number of instances of category i in image j, and the denominator is the sum of all categories in the image. Objects with a low TF score are likely either errors or are so rare they would be unreliably located anyways.

IDF rates the frequency of the category across all locations.

$$idf_i = \log \frac{|D|}{|\{d_j : o_i \in d_j\}|}$$

where $|D|$ is the total number of images in our training set, and denominator is the number of documents containing category i. Categories with a low IDF score appear in many of the images, so also have a high chance of being categorizer errors or camera noise. They may also simply be very ordinary categories such as a window.

The final TF*IDF score is

$$tfidf_{i,j} = tf_{i,j} \cdot idf_i$$

A high TF*IDF score requires both a category i that is common in image j, but rare among the other images.

3. Implementation and Experiment Design

The self localization module was implemented as part of the SeeAsYou system. The inputs to the module were sets of categories produced by the precategorizer; they were grouped by location, and when all the training locations' categories were received by the self localization module, a TF*IDF weight was calculated and assigned to each category. During testing the categories were compared to the ones stored for each location and the sum of TF*IDF weights was computed for all matches; as a result we had a separate TF*IDF score with respect to each training location.

To test our approach we gathered 360-degree panoramas consisting of 68 320 by 240 images in three locations – Robot Lab, North Lab and Lobby – all of them in the Computer Science building (see Figure 1 for sample images). The locations shared some similar features like the floor and the wall colors, but also had somewhat different furniture and computers. For example, the Robot lab had a series of cabinets along one while the North Lab had black computer cases on the floor. Four panoramas were collected from the Robot Lab, five from the North Lab, and four from the Lobby (see Figure 2). For most of these tests, the only difference between the runs was the sample location. We also introduced environment variables for the fourth Robot Lab image set, rearranging some of the chairs in the room, and for the fourth Lobby image set, turning on the overhead lights. Additionally two Control panoramas were collected in an empty narrow hallway adjacent to the North lab.

4. Results

TF*IDF proved to be a good measure for robot localization when used with the hierarchical thalmocortical categorizer in the SeeAsYou system. After selecting a combination of three 360-degree panoramas which produced the highest number of distinguishing categories (with TF*IDF weights greater than 0) for the training locations (namely sets #2 from Robot Lab and Lobby locations, and set #4 from North Lab location), we applied two similar approaches to compute TF*IDF sums for test locations and both produced favorable results.

Our first approach was to only sum TF*IDF values for the most specific leaf categories of the hierarchical categorizer (see Leaf TF*IDF section of Table 1). This produced correct localization for 9 of 10 test panoramas based on the largest TF*IDF sum with respect to each training location. Two Control panoramas taken in a narrow empty hallway produced maximum TF*IDF sums lower than any of the other test panoramas; this could allow for a clear rejection threshold (maximum TF*IDF sum less than 0.17 in our dataset) to eliminate locations that the system wasn’t trained to recognize, although this is not a technique we chose to implement for this project.

Our second approach was to use three levels of a category hierarchy and sum respective TF*IDF scores over all three levels. The same training panoramas were used since they still produced distinguishing categories on all the levels. Also, using the same training panoramas enabled a direct comparison of the two approaches. The maximum hierarchical TF*IDF sums correctly identified locations of all test panoramas; however, the scores for the control panoramas weren’t clearly lower than TF*IDF scores produced by other panoramas (see Table 1). Although no clear rejection threshold could be set, the added location recognition accuracy may be used in conjunction with short-term memory to eliminate locations that the system wasn’t trained to recognize: for example, if nearby test samples produce highest scores in different non-adjacent training locations, such test samples have high probability of belonging to a location that the system wasn’t trained to recognize.

5. Conclusions and Future Work

Overall, TF*IDF performed quite well throughout our testing and can be used effectively for robot localization. In our experiments we achieved over a 90% success rate both in the Leaf-only tests and in the 3 level hierarchy test. Many of these tests included random individuals walking by during image collection but the system was still able to determine its location without difficulty. The rearrangement of furniture in the Robot Lab and turning on the lights in the Lobby also had no effect on the ability of the system to correctly identify those locations.

An obvious extension to the system would be to recognize more locations, store more panoramas, and utilize training sets per location. Although the panorama locations we chose for each of the three locations were selected specifically because they were representative of the various categories to be found there, they cannot capture all of the information in just one pass. By having more samples total, we could have more samples to use for training, which would improve the robustness of the system and allow it to more accurately identify more locations. It would also be interesting to see how the localization system performs outdoors where category distinction is more difficult.

For a complete SLAM system, we must implement a mapping mechanism. Currently, new locations must be manually identified and added as training sets for the self localization module. By adding a short term memory to store several of the previous locations and an adjacency map for the known locations, the robot can track its movement. If the top scoring location changes, for example from the Robot Lab to the Lobby, the robot can check if where it was is adjacent to where it now believes it is. If the two locations are not adjacent, then its current location is in fact a new one and should be added to the training set.

Figure 1: From the top left moving clockwise, select images taken from the Robot Lab, North Lab, Lobby, and Control.

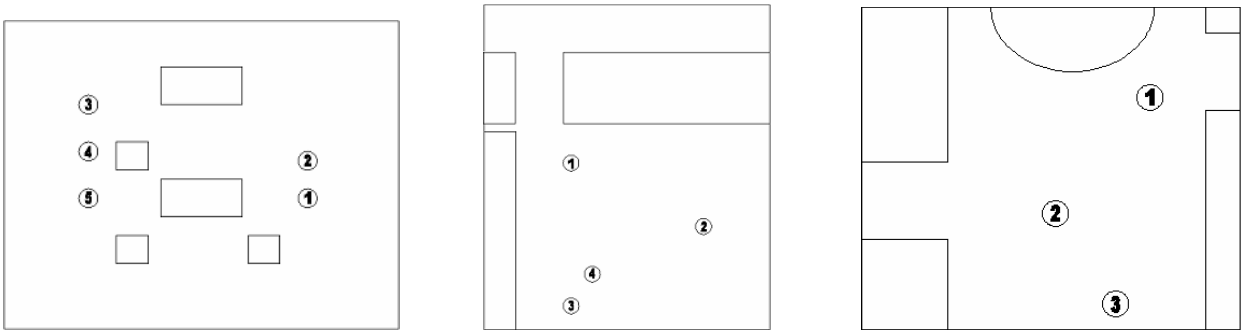

Figure 2: Our three testing environments: the North Lab, Robot Lab, and 2nd floor Lobby respectively. Location 4 in the Robot lab also had some of the chairs in the room rearranged. Location 4 in the lobby was the same as location 3 but with the overhead lights turned on.

	Leaf TF*IDF			Level 1 TF*IDF Hierarchy			Level 2 TF*IDF Hierarchy			Sum Over Three Hierarchy Levels		
	Robot Lab Score	**North Lab Score**	**Lobby Score**	**Robot Lab Score**	**North Lab Score**	**Lobby Score**	**Robot Lab Score**	**North Lab Score**	**Lobby Score**	**Robot Lab Score**	**North Lab Score**	**Lobby Score**
Robot Lab #1	0.238296	0.215522	0.191677	0.0210241	0.031094	0.033645	0.0381723	0.0304721	0.0379584	0.2974924	0.2770881	0.2632804
Robot Lab #2	0.395891	0.0414852	0.0614729	0.0210241	0	0.0155285	0.0435976	0.00124376	0.0198419	0.4605127	0.04272896	0.0968433
Robot Lab #3	0.250053	0.0827483	0.168775	0	0.031094	0.0181165	0.00871951	0.0317158	0.0247675	0.25877251	0.1455581	0.211659
Robot Lab #4	0.304339	0.198432	0.164991	0.0210241	0.031094	0.033645	0.0354597	0.0367708	0.0388211	0.3608228	0.2662968	0.2374571
North Lab #1	0.0994997	0.207257	0.0762207	0	0.031094	0.0181165	0.00400459	0.038897	0.0236542	0.10350429	0.277248	0.1179914
North Lab #2	0.132666	0.308578	0.177431	0	0.031094	0.0181165	0.0107218	0.035527	0.02243	0.1433878	0.375199	0.2179775
North Lab #3	0.17665	0.307536	0.250813	0	0.031094	0.0181165	0.02243	0.035527	0.0332547	0.19908	0.374157	0.3021842
North Lab #4	0.0820942	0.425808	0.11215	0	0.031094	0.0181165	0.0020023	0.0451957	0.0189792	0.0840965	0.5020977	0.1492457
North Lab #5	0.117067	0.328096	0.178293	0	0.031094	0.0181165	0.00500574	0.0367708	0.0253795	0.12207274	0.3959608	0.221789
Lobby #1	0.203323	0.150711	0.279055	0.0210241	0.031094	0.033645	0.027031	0.0292283	0.0511206	0.2513781	0.2110333	0.3638206
Lobby #2	0.0750861	0.0827099	0.500869	0.0210241	0.031094	0.033645	0.0280322	0.0304721	0.0575209	0.1241424	0.144276	0.5920349
Lobby #3	0.175744	0.10391	0.145651	0	0.031094	0.0181165	0.00942984	0.0376533	0.0262422	0.18517384	0.1726573	0.1900097
Lobby #4	0.108572	0.0687056	0.219476	0.0210241	0.031094	0.033645	0.0250287	0.0309133	0.0385705	0.1546248	0.1307129	0.2916915
Control #1	0.163663	0.0605205	0.0499785	0.0210241	0	0.0155285	0.0277413	0.00505496	0.0181165	0.2124284	0.06557546	0.0836235
Control #2	0.0688502	0.0129587	0.0454144	0	0	0	0.00400459	0	0.00345077	0.07285479	0.0129587	0.04886517

Table 1: Our results from all tests. Higher scores denote a better match.

References

---

[1] Michael H. Bruch, G.A. Gilbreath, J.W. Muelhauser, J.Q. Lum. Accurate waypoint navigation using non-differential GP

[2] Yoshiro Negishi, Jun Miura, and Yoshiaki Shirai. Mobile Robot Navigation in Unknown Environments Using Omnidirectional Stereo and Laser Range Finder

[3] Luis Montano José R. Asensio. Real-Time Robot Navigation in Unstructured Environments Using a 3D Laser Rangefinder

[4] Falk Schubert, Thorsten Spexard, Marc Hanheide, and Sven Wachsmuth. Active Vision-Based Localization For Robots In A Home-Tour Scenario

[5] Christophe Giovannangeli. Philippe Gaussier. Gaël Désilles . Robust Mapless Outdoor Vision-Based Navigation

[6] Michale Montemerlo, Sebastian Thrun, Daphne Koller, and Ben Wegbreit. FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem

[7] Tim Renner. Neural Sharpening in Object Recognition

[8] Juan Ramos. Using TF-IDF to Determine Word Relevance in Document Queries

[9] James Z. Wang, Yanping Du, RF*IPF: A Weighting Scheme for Multimedia Information Retrieval

Literature Consulted

Zender, H., Kruijff, G. M., 2007. Multi-Layered Conceptual Spatial Mapping for Autonomous Mobile Robots. AAAI Spring Symposium on Control Mechanisms for Spatial Knowledge Processing in Cognitive / Intelligent Systems.

Moroz, O., Jensfelt, P., Zender, H., Kruijff, G. M., Burgard, W., 2007. From Labels to Semantics: An Integrated System for Conceptual Spatial Representations of Indoor Environments for Mobile Robots. ICRA Workshop.

Wyeth, G., Browning, B., 1997. Cognitive Models of Spatial Navigation from a Robot Builder's Perspective. Adaptive Behaviour Journal.

Karlsson, N., Di Bernardo, E., Ostrowski, J., Goncalves, L, Pirjanian, P., Munich, M.E., 2005. The vSLAM Algorithm for Robust Localization and Mapping. ICRA

Thrun, S., Montemerlo, M., Koller, D., Wegbreit, B., Nieto, J., Nebot, E., 2004. FastSLAM: An efficient solution to the simultaneous localization and mapping problem with unknown data association. Journal of Machine Learning Research.

Thurn, S., 2002. Robotic mapping: A survey. In Exploring Artificial Intelligence the New Millenium. Morgan Kaufmann, 2002.

Thurn, S., 2000. Probabilistic Algorithms in Robotics. AI Magazine.

Fox, D., Thrun, S., Burgard, W., Dellaert, F., 2001. Particle Filters for Mobile Robot Lcalization. Sequential Monte Carlo Methods in Practice. Springer.

Se, S., Lowe, D., Little, J., 2002. Mobile Robot Localization and Mapping with Uncertainty using Scale-Invariant Visual Landmarks. International Journal of Robotics Research.

Se, S., Lowe, D., Little, J., 2001. Local and Global Localization for Mobile Robots using Visual Landmarks. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

Tversky, B., 2003. Structures of Mental Spaces: How People Think About Space. Environment and Behavior.

Rekleitis, I., 2003. A Particle Filter Tutorial for Mobile Robot Localization. Technical Report for Centre for Intelligent Machines, McGill University.

Franz, M., Mallot, H., 2000. Biomimetic Robot Navigation. Robotics and Autonomous Systems.