Image Processing for Visual Navigation of Mobile Robot using Fluorescent Tubes

Fabien LAUNAY Akihisa OHYA Shin'ichi YUTA

Intelligent Robot Laboratory
University of Tsukuba
1-1-1 Tennoudai, Tsukuba, Ibaraki 305-8573 JAPAN
Tel: +81-298-53-5155, Fax: +81-298-53-6471
E-mail: {launay,ohya,yuta}@roboken.esys.tsukuba.ac.jp

Abstract: This paper presents an image processing method for an autonomous mobile robot indoor navigation system using fluorescent lights. The self-localization of the vehicle is done by detecting the position and orientation of fluorescent tubes located above it's desired path thanks to a camera pointing to the ceiling.

A map of the lights based on odometry data is built in advance by the robot guided by an operator. During the teaching of the environment, the robot performs autonomously lights detection and adds appropriate information for each landmark to the lights' map it is building. Then a graphic user interface is used to define the trajectory the robot must follow with respect to the lights. While the robot is moving, an image processing algorithm similar to the one used during the teaching step is used in order to compare the position and orientation of the detected lights to the map values, which enables the vehicle to cancel odometry errors.

1. Introduction

When a wheel type mobile robot navigates on a two dimensional plane, it can use sensors to know its relative localization by summing elementary displacements provided by incremental encoders mounted on its wheels. The main default of this method known as odometry is that its estimation error tends to increase unboundedly[1]. For long distance navigation, odometry and other dead reckoning solutions may be supported by an absolute localization technique providing position information with a low frequency.

Absolute localization in indoor navigation using landmarks located on the ground or on the walls is sometimes difficult to implement since different objects can obstruct them. Therefore a navigation system based on ceiling landmarks recognition can be thought as an alternative to this issue.

The navigation system we developed consists in two steps. In the first step, the vehicle is provided with a map of the ceiling lights. Building such a map by hand quickly becomes a heavy task as its size grows. Instead, the robot is guided manually under each light and builds the map automatically. The second step consists in defining a navigation path for the vehicle and enabling its position and orientation correction whenever it detects a light recorded previously in the map. Both first and second steps require fast and robust image processing algorithms.

Since the map built by the robot is based on odometry whose estimation error grows unboundedly, the position and orientation of the lights in the map do not correspond to the reality. However, if the trajectory to be followed by the vehicle during the navigation process is defined appropriately above this distorted map, it will be possible for the robot to move along any desired trajectory in the real world. A GUI has been developed in order to facilitate this map-based path definition process.

We equipped a mobile robot with a camera pointing to the ceiling. During the navigation process, when a light is detected, the robot calculates the position and the orientation of this landmark in its own reference and thanks to a map of the lights built in advance, it can estimate its absolute position and orientation with respect to its map.

We define the *pose* of an object as its position and orientation with respect to a given referential.

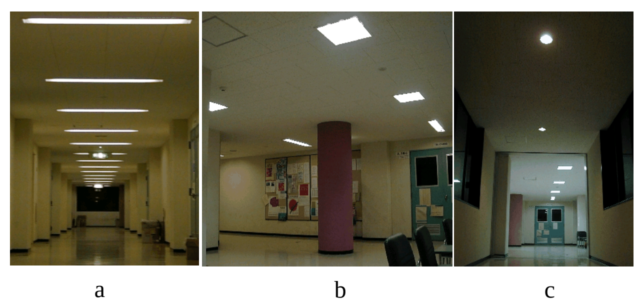

Figure 1. Target environment consisting of lights of different shapes in a corridor exposed to luminosity variations due to sunning.

2. Fluorescent tube detection

2.1 Fluorescent tube model

It is natural to think of fluorescent tube as a natural landmark for a vision-based process aimed at improving the localization of a mobile robot in an indoor environment. Indeed, problems such as dirt, shadows, light reflection on the ground, or obstruction of the landmarks usually do not appear in this case.

One advantage of fluorescent tubes compared to other possible landmarks located on the ceiling is that once they are switched on, their recognition in an image can

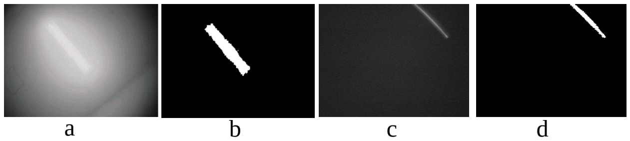

Figure 2. (a),(c) Typical camera images of the ceiling of a corridor containing a fluorescent light. The axis of the camera is perpendicular to the ceiling. (b),(d) Binarized images.

be performed with a very simple image-processing algorithm if we suppose that they are the only bright elements that are permanently found in such a place.

When a 256 grey levels image such as the one shown in Fig.2.a containing a fluorescent tube is binarized with an appropriate threshold $0 \leq T \leq 255$, the only element that remains after this operation is a rectangular shape (Fig.2.b). If we suppose that no more than one light at a time can be seen by the camera located on the top of the robot, a fluorescent tube can be modeled by a given area N corresponding to the number of pixels brighter than a thresholded appearing in the image of the ceiling. This hypothesis which is validated in most traditional corridors leads to simple image processing algorithms as we will see below.

2.2 Image thresholding

When the robot is guided manually in a new environment in order to build a map of the lights, it has to look by itself for possible landmarks located on the ceiling. It does so by taking pictures of the ceiling regularly and analyzing each of them to detect if a light appears in the image or not. The first step of the image processing consists in finding the appropriate threshold that would lead to a binarized shape corresponding to a light eventually present in the image.

If we compare the histogram of a ceiling image containing no bright object with a ceiling image containing a fluorescent light, we notice that both of them contain a peak corresponding to the predominant grey levels of the image. See Figure 3.a,b for an illustration of both cases. Yet, the number of pixels brighter than the upper limit of the peak is very small in the case of an empty ceiling image whereas it is greater in the case of a light image. In the latter case, these pixels define the shape of the fluorescent light included in the image. Figure 3.e and f are a zoom along the vertical axis of Figures 3.c and d. They show clearly this property.

Therefore determination of the threshold and the detection of the presence of a light in the image can be performed as follows :

- Compute the image histogram.
- Select the threshold as the grey level where the decreasing half part of the peak goes under a certain pixels number value. This pixels number value was set to 50 in the experiment we made and does not have to be set with a great precision as we will justify in next section.
- Compare the number of pixels brighter than the threshold to a given value. If this number is greater than the value, then the image contains a bright object and further image processing is done. If not, we consider that no light is contained in the image, the image is ignored and another image is taken before performing the same light detection algorithm.

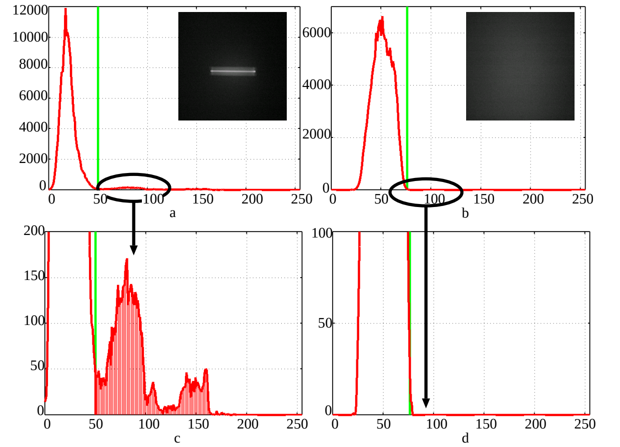

Figure 3. (a) Histogram of a ceiling image containing a light, (b) Histogram of a ceiling image without light. (c) Zoom of (a), (d) zoom of (b). Horizontal axis : grey level (0-255), Vertical axis : pixels number.

3. Light pose computation for map building

3.1 Incomplete light elimination

In order to take advantage of lights detected during the learning step for the navigation process, the robot has to store in the map the *pose* of a new light whenever it detects it. Since it is not possible to determine completely the *pose* of a light not fully included in an image without making hypothesis concerning its shape and size, images containing lights touching the border must be filtered at first. Figure 2.c shows a picture taken by the robot during the learning process which lead the light detection algorithm to keep these images for further image processing, but which can not be used to determine the position of the landmark.

3.2 Distortion correction

Once binarized images containing lights touching the border are eliminated, an algorithm is used to correct image distortion before determining the *pose* of the landmark in the image. Distortion correction enables to compute the *pose* of the light in the image with a better precision, especially when the binarized shape is not centered in the image. This image processing step involved in our system requires more time than the other algorithms. However, it should not penalize the teaching, nor the navigation process.

The transformation between a pixel location in a non-distorted image and its location in a distorted one is approximated by polynomials of degree 3 whose coeffi-

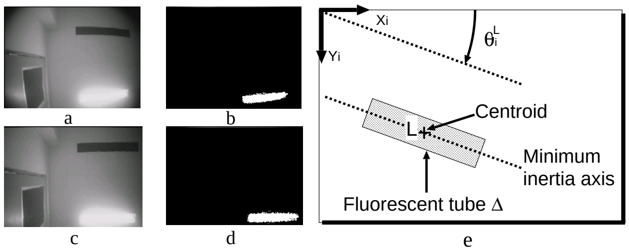

Figure 4. Fluorescent light located on the border of the image. (a),(b) original and binarized images before distortion correction, (c),(d) after distortion correction. (e) Moment-based features of a fluorescent tube.

cients are computed in advance. In order to avoid the computation of the polynomials giving the coordinates a pixel in a normal image would have in a distorted one for each pixel whenever an image is taken, the x and y offsets for each pixel are computed in advance and stored in memory. By doing so, the gain of time on the system we developed is about 11.8 seconds for each image.

Figures 4.a,b,c,d shows an image containing a fluorescent tube located on the border before and after distortion correction. Distortion correction can not be ignored unless the robot is supposed to navigate exclusively right under fluorescent lights.

3.3 Light pose in the image

The location of the tube's centroid L in the image as well as the orientation of its least moment of inertia axis θ_i^L are computed using the moment-based features of the binarized shape Δ, as shown in Fig.4.e according to the following equations.

$$\begin{cases} X_i^L &= \frac{1}{N}\sum_{(x,y)\in\Delta} x \\ Y_i^L &= \frac{1}{N}\sum_{(x,y)\in\Delta} y \\ \theta_i^L &= \frac{1}{2}\arctan\left\{\frac{2\mu_{1,1}}{\mu_{2,0}-\mu_{0,2}}\right\} \end{cases}$$

Where $\mu_{p,q} = \sum_{(x,y)\in\Delta}(x - X_i^L)^p(y - Y_i^L)^q$ stands for the (p,q) order central moments of the shape Δ (N is the pixel number of the binarized shape). This method based on the statistical properties of the image gives precise information without requiring edge detection, noise filtering or any time consuming algorithm. Furthermore, the threshold used to binarize the image does not have to be set precisely with this approach. In the case of the image given in Fig.3.a, a threshold located around 50 would only result in a translation of a few pixels of the shape centroid in the image and would not affect its orientation.

3.4 Shape recognition

Although it is possible to compute the light centroid position with the previous equations whatever the light's shape might be, non rectangular fluorescent lights require particular attention concerning the computation of their orientation. We chose to ignore the result of the equation giving light orientation in the case of square and round shaped lights as shown in Fig.1.b,c. The distinction between a rectangular and non-rectangular light is done by comparing the ratio $\frac{\sum_{(x,y)\in\Delta}[(x-X_i^L)\cos(\theta_i^L+90^o)-(y-Y_i^L)\sin(\theta_i^L+90^o)]^2}{\sum_{(x,y)\in\Delta}[(x-X_i^L)\cos\theta_i^L-(y-Y_i^L)\sin\theta_i^L]^2}$ to 1. A ratio close to 1 means the inertia of the shape with respect to the minimum inertia axis is the same as its inertia with respect to an axis perpendicular to the minimum inertia axis, hence that the shape is not rectangular.

3.5 Light pose in the robot referential

Once the *pose* of the landmark in the image is obtained it has to be converted in the robot referential. The orientation of the light can be easily obtained by taking into account the flattening rate of the image if necessary. Nevertheless the conversion between the light's centroid location in pixels in the image and its location with respect to the vehicle in meters requires to know the distance between the camera and the ceiling. The robot should be able to perform this operation during the learning process without being given in advance the ceiling height by an operator.

To achieve this purpose, two successive pictures of the ceiling are taken from two different places by the robot. By doing so, it is possible to compare how much the binarized shape centroid has moved in the image to the distance ran by the robot between the two shots. Once this conversion rate is obtained, it is possible to determine the light's location in the robot referential and finally in the global referential where the map is being built. The conversion rate is recorded as a map data for each light so that the robot can correct its position during the navigation process by taking only one shot of the landmark recorded in its map.

If a light is detected more than one time by the robot while it is taking pictures of the ceiling, the light is recorded only once in the map, provided the odometry errors made by the robot between the different observation places are not too big.

4. Lights map based navigation

When the teaching phase is completed, the robot holds a map of the lights that can be used later for the autonomous navigation process. The image processing algorithms involved in the navigation process are identical to the learning process algorithms but do not require two shots of the ceiling to determine the relation between the image metric and the robot's referential metric.

During the navigation process, the robot starts taking pictures of the ceiling whenever odometry data get close to the position of a light recorded in its map. When a light is found, the robot *pose* in the map referential is corrected by using the light *pose* in the map and the relative *pose* of the robot with respect to the observed landmark. Since the calculation of the robot's absolute *pose* estimation from an image is time consuming, retroactive data fusion with odometry data is necessary[3]. This function is achieved thanks to an algorithm developed previously in our laboratory[2].

5. Implementation and experiment

We implemented this system on the YAMABICO robot[4] developed in our laboratory. The sensors used by the robot to estimate its *pose* are optical encoders mounted on the wheels and a board CCD black and white camera facing the ceiling. The navigation program and the absolute *pose* estimation program based on fluorescent lights are implemented as independent modules. It is therefore possible to run simultaneously other *pose* estimation programs based on different landmarks and using different sensors without any modification of the existing vision-based module.

The validity of the proposed navigation system has been shown by making experiments in the corridor shown in Fig.1 at different times of the day. The robot was first guided under each light in order to build the landmarks map. Then it used the map and a path defined above it to navigate in the middle of the corridor until a goal point and come back to its starting point. The maximum speed of the robot was 35 cm/s and total distance on one way was about 50 meters. On the robot's path 24 fluorescent tubes of different shapes were present, separated by a distance varying from 2.2 meters to 4.5 meters.

The experimental results of one of those experiments are shown in Figure 5 where the bold line corresponds to the odometry data of the robot. When a light is found, the absolute *pose* of the robot is corrected after a certain delay represented by the distance between the marks '+' and '×' respectively. The table below gives average computing times for the different steps of the image-processing algorithm. All image-processing is done on board by a vision module developed in our laboratory which is equipped with a 20MHz Thomson T805 processor.

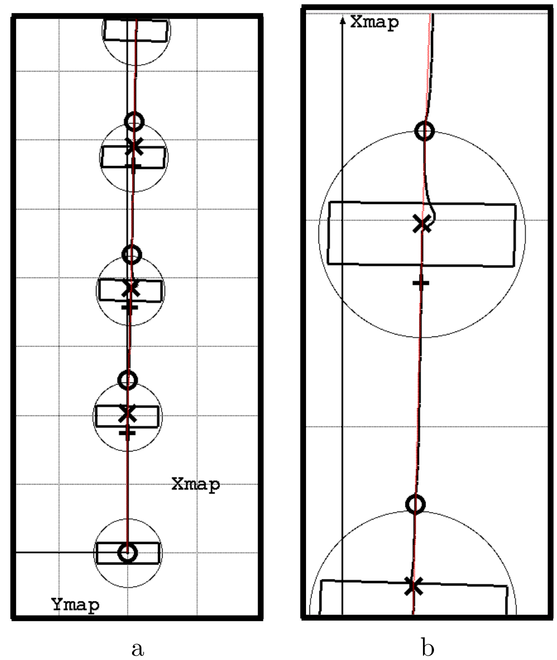

Figure 5. Odometry data of the robot correcting its trajectory using the detection of fluorescent lights in a corridor. (a) Zoom on the first lights, (b)zoom on one light. Rectangles correspond to lights and discs to their detection area. '+': light detected, '×': *pose* correction, '∘': the robot enters a new segment.

Table 1. Average computing time for the different steps of the image-processing algorithm. The image size is 756×238 pixels.

Capture	0.03 s
Distortion correction, thresholding	1.82 s
Borders scanning	0.19 s
Light position calculation	0.06 s
Light orientation calculation	0.13 s
TOTAL	2.23 s

6. Conclusions and future work

In this paper, we presented a complete navigation system that enables a mobile robot to achieve long distance indoor navigation thanks to the lights located above its trajectory.

In a first step, the robot builds in advance a map of these landmarks that can be detected easily. Once the map-building process is finished, the trajectory the vehicle has to follow is defined above the previous map thanks to a GUI. In the second step, the robot looks for the lights it has learnt and fuses its new estimated absolute *pose* with odometry whenever a landmark is detected during the navigation process.

Experiments show that it is possible for the robot to navigate with precision on a long distance without any other position or orientation sensing system than optical encoders and a black and white camera pointing to the ceiling.

Future work will address how to take advantage of loops in the map that may occur during the map building process. Because of the accumulation of odometry errors while the robot is guided for the first time throughout its environment, a light detected more than one time should in any case be recorded only once in the map. Since the *pose* of the light computed by the robot in a global referential will be different whenever the vehicle re-encounters a landmark during the learning process, further work has to be done to cope with this issue.

References

[1] C. Ming Wang, "Location estimation and uncertainty analysis for mobile robots", *Proc. of 1988 International Conference on Robotics and Automation* , pp 1230-1235.

[2] S. Maeyama, A.Ohya, S.Yuta, "Non-stop outdoor navigation of a mobile robot - Retroactive positioning data fusion with a time consuming sensor system", *Proc. of 1995 Intelligent Robots and Systems* , Vol.1, pp 130-135.

[3] A. Kosaka, M. Meng, A. Kak, "Vision-guided mobile robot navigation using retroactive updating of position uncertainty", *Proc. of 1993 International Conference on Robotics and Automation* , pp 1-7.

[4] S. Yuta S. Suzuki, S. Iida, "Implementation of a small size experimental self-contained autonomous robot – sensors, vehicle control, and description of sensor based on behavior–", in *Experimental Robotics II* , R. Chatila *et al.* , Eds. Berlin, Germany: Springer-Verlag, 1993, pp 344-358.