Static Environment Recognition Using Omni-camera from a Moving Vehicle

Teruko Yata, Chuck Thorpe

The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213 USA

Frank Dellaert

College of Computing
Georgia Institute of Technology
Atlanta, Georgia 30332 USA

Abstract

This research aims to develop a sensor system which will be able to recognize city area environments from a moving vehicle. A city area which is a rather cluttered outdoor man-made environment requires different recognition methods from indoor or highway environments. Toward this goal, we employed an omni-directional camera and vehicle odometry. Also, for better recognition in the total sensing system, estimated geometrical information is fed-back to image processing. In this paper, as our first progress report, we explain our approach of using omni-camera on a vehicle and setup of a platform, and show an experimental result of two dimensional static environment recognition from a moving vehicle.

Keywords : City environment, Vehicle, Environment recognition, Optical flow, Omni-camera.

1 Introduction

This research aims to develop a sensor system which will be able to recognize city area environments from a moving vehicle.

There have been several research projects which have achieved autonomous driving using multiple sensors and communications in a limited environment like highways [1]. However, we want to recognize much more complicated environments like city area which is rather cluttered out-door man-made environment. It requires much more ability to detect complicated environments including buildings, bicycles, cars and humans. In city area, a vehicle moves slower, but there are cars, bicycles, humans which move unexpectedly. And it needs to detect various objects rather nearer in the environment. Consequently, it requires different recognition methods from indoor or highway environments. Achieving environments recognition of those city area environment from a vehicle is important for future drivers assistance system or autonomous driving systems.

For general mobile robots sensing, it is important to extract essential information of a wide area from fewer sensors especially for practical reasons [2]. Recently, omni-camera which can get 360 field of view in an image has been developed and commercialized [3]. It is a good device to cover wider area even though image resolution is lower than for a regular camera. And, there are several researches using it on small size mobile robots [5] [6] [7] and showing its potential.

Toward environment recognition in a city area, as a first step, we have implemented a method to recognize a static environment from a vehicle using an omni-directional camera and vehicle odometry.

In this approach, we have employed template based optical flow, selecting explicit landmarks in the image processing. Because the environment contains various shaped objects and we want to detect locations of those objects, edge detection by accumulating vertical edge information in the horizontal direction as in [5] can not be applied. Optical flow detection of entire images as in [6] has a different purpose than our work. Once we could achieve successful tracking of points in the environment, we can apply the structure from motion method to take corresponding points between frames [7] to enforce accuracy of vehicle motion. Based on those previous researches, we decided to use an omni-camera on a vehicle and template based optical flow.

One of the original points of our method is applying feedback of estimated landmark location for template based optical flow processing. From the omni-camera image, basically we can get only angle information without depth information. Once we estimate the location of the landmarks, we can predict where the template will appear in the next image by knowing vehicle location using odometry data. That prediction improves the reliability of template matching search. We have applied this feedback for omni-camera images and successfully improved the environment recognition result.

In this paper, we explain our setup of a platform and sensors in section 3, and strategies of environment recognition in section 4, and then show a preliminary result of

an experiment in section 5. In section 6, we will discuss our current results and future work, and then conclude this paper in section 7.

2 Hardware setup

As a sensor, we employ an omni-directional camera which can get a 360 degrees view in a single frame, and we use odometry data of a platform. Those odometry data and omni-directional image are synchronized by using time stamps.

2.1 Platform – Vehicle

As a platform, we are using a car – 2000 Jeep Wrangler Sport named "NavLab11." One modification which used in this research was encoder data which is obtained by counting the four 51 slots/wheel rotation gears used with the anti-lock braking system. Odometry of the vehicle, X and Y location and heading direction of the vehicle in the world coordinate, are calculated from the encoder data.

2.2 Omni-camera

An omni-directional camera – Remote Reality ParaCamera S-360, is mounted in front of the vehicle as shown in Fig. 1. This is a black and white camera with parabolic reflection mirror designed to get 360 degrees view in an image. The camera output is NTSC signal and it was connected to a digitizer on a PC. In these experiments, only half of the image was used because camera was fixed in front of the vehicle, so half of the frame showed the vehicle itself.

Figure 1: The omni-camera is mounded in front of the vehicle – NavLab11.

3 Strategy

For the first step for environment recognition, we use template based optical flow in images and odometry data to estimate landmark positions in a two-dimensional environment assuming the environment is static.

Our global strategy is repeating the following steps on every new frame:

1. Transform the omni-directional image into a cylindrical image.
2. With respect to landmark which has been tracked from the previous image, predict the initial search point of the template on the image using the previously calculated location of the landmark and the vehicle motion data.
3. Search for the best matching location of the template on the image starting from the initial point calculated in step 2.
4. If nice matching was not found, stop tracking the template.
5. Search for candidate new landmarks in the image.
6. If there are areas with no tracked templates, add new templates using the new landmarks found in step 5.
7. From detected location of the template on the image, and the vehicle location obtained from odometry data, calculate location of the landmarks in world coordinate.

Detail of each process is explained in following subsection.

Calculating optical flow of the entire images is also an idea for extracting environment shape [6], however we would like to aim to detect various objects which is impossible to have accurate shape such as humans or bicycles in future moving objects detection, so that template tracking methods were employed in the vision process. Moreover, the template tracking is a simpler and faster method.

Also, basically the vehicle can move only on a two dimensional surface, and it is useful to know the location of vertical objects for environment recognition. However, compared to an indoor environment, in outdoor environment, vertical edges are not explicit, and a landmark selection process and template tracking method is necessary.

3.1 Pre-process of Omni-image

We transform the part of the omni-image corresponding to the forward-looking 180 degrees into a cylindrical image.

We also trim off the bottom (nearer to the vehicle) and top (sky) portions of the image. When there is no corresponding pixel of cylindrical image on the omni-directional image, the pixel value is interpolated by values around the pixel. All following image process are done on this extracted cylindrical image.

3.2 Landmark Selection

Vertical continuous edges are selected as landmarks to calculate optical flow. Many vertical lines like edges of building, humans, edges of vehicle are present in city environments. Also because vehicles can move mainly in two-dimensional surfaces, those vertical lines are useful landmarks to know its own motion.

Vertical Sobel filter [4] was applied on the image to detect vertical lines. We define a peak single pixel which exceed a threshold level as a vertical line. Then landmarks are generated by the following simple rules:
– as long as the detected vertical edges continue, those pixels are regarded as a group to make a landmark.
– the group must contain more than a certain number of pixels.
The rectangular area which contains edges which satisfied the rules is used as the templates of the landmark.

3.3 Template Tracking

Optical Flow Calculation

Optical flow is calculated by using the templates in following frames. Actual templates are extracted from the original image based on the rectangular area of the landmark. Optical flow is calculated by finding the best matching area of the entire template in the next following frame. The contents of the template are updated each frame when tracking is successful.

Initial Point Feedback

The initial template search location on the image is calculated from the landmark's estimated position. (The landmark location estimation is explained at the section 3.5) Using the odometry data which correspond to each image, it is possible to predict where the template of the landmark will be on the image. The template is searched for starting from the predicted initial point.

Deleting or Adding Landmark

Once a landmark was selected, it is tracked updating the template contents at each frame. When the average difference of the template is going over a threshold value, tracking stops. On the other hand, at each frame, new landmarks selecting process is done, and when there are no existing landmarks near the new landmark, the new landmark is added to landmarks to be tracked.

3.4 Angle Measurement

For constructing a two dimensional map, bearing angle information of each tracked template is used. Bearing angles are defined in the vertical direction by taking the median of the template. Because template size depends on the size of landmark and constant among frames, the template median gives reasonable measurement value. The bearing angle information is used for calculating locations of the landmarks.

3.5 Landmark Location Calculation on 2D Map

Based on the odometry data, the position of the omni-camera can be calculated. The omni-camera provides only bearing information of the landmarks by the previously explained process. From the bearing angle information, a straight line of sight to tracked landmarks is drawn at each position, and the crossing points of those lines belonging to the same landmark are resultant landmark location in two dimensions. Theoretically, if there are measurement from two frames in different locations, it is possible to determine a landmark location. However, because of insufficient accuracy or error in tracking, the points are not given accurately from the line of sight crossing. For estimating those locations in two dimensional map, we applied extend Kalman filter using measured bearing angle and odometry data.

Bearing Measurements

A landmark has global coordinates $X = (x, y)$, and is seen from a vehicle at $V = (a, b, c)$, where c is the heading of the vehicle in radians. We measure the bearing z to the landmark, obeying the following measurement equation

$$z = h(x, y) + n$$

where n is Gaussian noise with standard deviation σ, and the measurement function h is simply the arc-tangents in the vehicle coordinate frame:

$$h(X) = arctan\frac{(y-b)}{(x-a)} - c$$

The Jacobian H is

$$H = [\frac{(b-y)}{r^2}, \frac{(x-a)}{r^2}]$$

where $r = \sqrt{(x-a)^2 + (y-b)^2}$ is the range to the landmark.

Extended Kalman Filter

Here is a very simple derivation of the extended Kalman filter in terms of maximum a posteriori estimation. In general, if Z is a vector valued measurement with noise covariance matrix R, X is the state and (X_0, P) is the mean-covariance of the predicted state, we need to minimize the log-posterior:

$$\hat{X} = argmin(Z - h(X))^T R^{-1}(Z - h(X)) + (X - X_0)^T P^{-1}(X - X_0) \tag{1}$$

Let's say $X = X_0 + \Delta X$ and replace the non-linear measurement function by its linear approximation:

$$h(X) = h(X_0) + H\Delta X$$

and take derivative with respect to ΔX and set to 0:

$$-H^T R^{-1}(Z - h(X_0) - H\Delta X) + P^{-1}\Delta X = 0$$

The solution of this system is simply:

$$\Delta X = (H^T R^{-1} H + P^{-1})^{-1} H^T R^{-1}(Z - h(X_0))$$

or:

$$\hat{X} = X_0 + (H^T R^{-1} H + P^{-1})^{-1} H^T R^{-1}(Z - h(X_0))$$

and the covariance of the new estimate is simply

$$\hat{P} = [H^T R^{-1} H + P^{-1}]^{-1} \tag{2}$$

Hence, we also have:

$$\hat{X} = X_0 + \hat{P} H^T R^{-1}(Z - h(X_0)) \tag{3}$$

where

$$h(X_0) = \arctan \frac{(y_0 - b)}{(x_0 - a)} - c \tag{4}$$

Integrating Bearing Measurements

In the case of bearing measurements, we have $R = \sigma^2$, and

$$H^T R^{-1} = \frac{1}{r^2 \sigma^2}[(b-y), (x-a)]^T \tag{5}$$

and

$$H^T R^{-1} H = \frac{1}{\sigma^2 r^4} \begin{bmatrix} (b-y)^2 & (b-y)(x-a) \\ (b-y)(x-a) & (x-a)^2 \end{bmatrix} \tag{6}$$

These can be plugged in straightforwardly in (2) and (3) to do the bearing update.

Note that $H^T R^{-1} H$ is singular: it does not yield any information on the range, hence the Gaussian is 'infinitely' long along the axis going from the vehicle to the landmark.

We define initial estimated location as line of sight crossing points using the first and the second bearing information of tracked template. We then repeat this estimation at each time update our appearance, template, and feedback the estimated location for the next template search initial point in the next image.

4 experiment

We performed an experiment to evaluate our strategies. The vehicle was driven for about 15 meters in a parking lot, and odometry data and 85 frames image from omni-camera were obtained during driving.

Fig. 2 shows an image from the omni-camera. The right side of the image is the grille of the vehicle. From the omni-camera image, only the part used for processing is extracted and transformed into cylindrical image as shown in Fig. 3. This image was transformed as 1 degree in the view corresponds to 2 pixels in the image.

Fig. 4 shows edges of the entire image detected by a vertical Sobel filter and Fig. 5 shows detected landmarks on the image (long connected edges).

Fig. 6 and Fig. 7 show tracked landmarks on the two continuous images. Two landmarks on the right side are added as new tracking landmarks as the result of landmark detection shown in Fig.5.

Fig. 8 shows bearing angles of tracked landmarks in frames (Locus map). Horizontal axis shows each frame number (time axis) and vertical axis shows measured bearing angle of each tracked template. Each dot connected lines is showing each track of the observed landmark.

Fig. 9 shows lines of sight to a single tracked landmark from the locations at which we observed that landmark. It can be observed most of lines cross at a single location in this figure. The motion of the vehicle obtained from odometry data is shown with lines. In this case, the vehicle made a loose curve at the left bottom and proceeded to the right top.

Fig. 10 shows all crossing points of line of sight for all landmarks in all sets of two sequential frames. It shows that just taking crossing points of pair of line of sight is in-

sufficient. The motion of the vehicle obtained from odometry data is also shown with lines.

Fig. 11 shows Kalman filter based detected landmark location in the 2D map. In this case, the initial template search location was the one previous template location. The motion of the vehicle obtained from odometry data is also shown with lines.

Fig. 12 shows Kalman filter based detected landmark locations in the 2D map using feedback for initial template search location using predicted location from estimated location. The only difference from 11 is whether initial template search location is predicted from estimated location or not. The motion of the vehicle obtained from odometry data is also shown with lines.

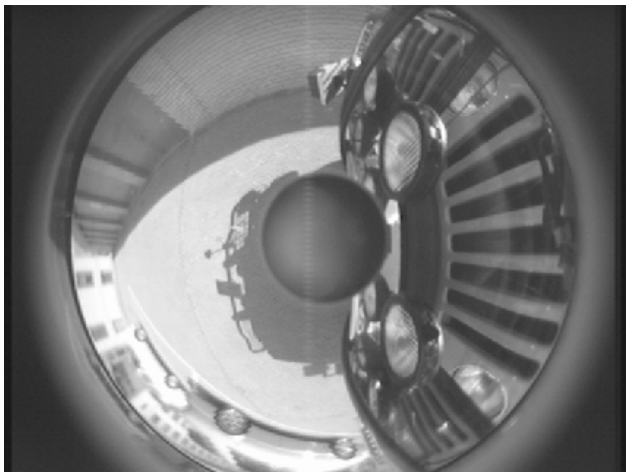

Figure 2: Image from the omni-camera.

5 Discussion

- Applying Kalman filter to estimate landmark location works. Fig. 10 shows points widely spread, however the filtering used in later figures gives better estimation for environment shape recognition.
- Kalman filter based template initial point feedback works. Comparing Fig.11 and Fig.12, feedback gives better environmental shape.
- Shadows or marks on the ground are sometimes detected. Because we do not calculate point elevation, marks on the ground can be mistaken for obstacles. Adding information from other simple distance sensor will be useful to prune those points.
- Landmark selection method should be considered

Figure 3: Extracted and transformed cylindrical image.

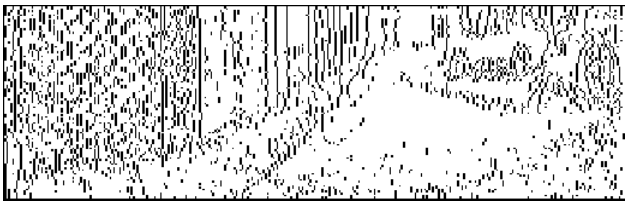

Figure 4: Detected vertical edges.

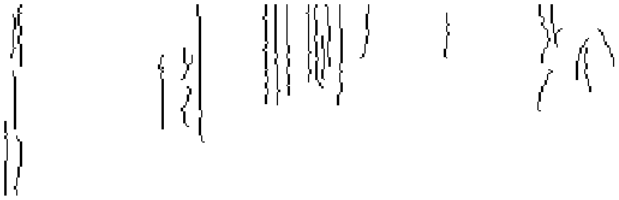

Figure 5: Detected landmarks.

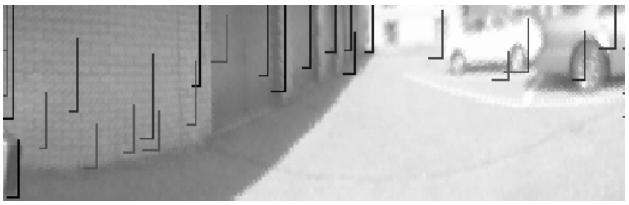

Figure 6: Tracked template (the first frame).

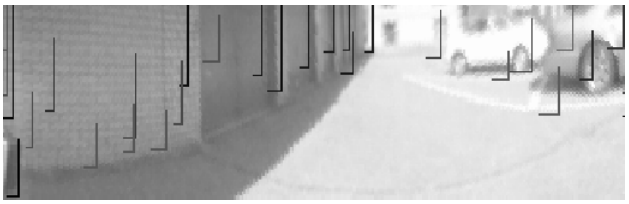

Figure 7: Tracked template (the second frame). Two landmarks on the right side are added as new tracking landmarks.

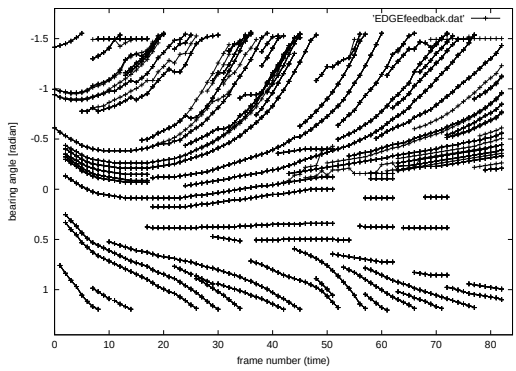

Figure 8: Tracked template in bearing angle. Horizontal axis shows each frame number and vertical axis shows measured bearing angle of each tracked template.

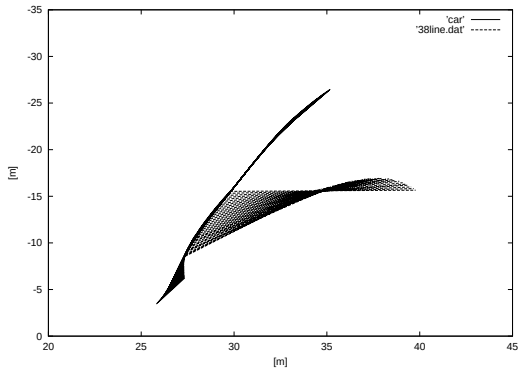

Figure 9: Line of sight lines to a single landmark from multiple locations which observed those landmarks.

more. In this result, edge of building or car were detected well, however, bicycle, human or other object should be considered.

- Omni-camera location has been changed to the corner of the vehicle in the newer setup, so we can get 270 degrees view.

- Odometry error is not considered. It seemed not so much to have a large effect in this result. However, rough surface or more rotation cause larger odometry errors, and it should be added.

- For further evaluation of this method, it will be useful to compare with other sensors, as line laser scanner, which can give environmental shape.

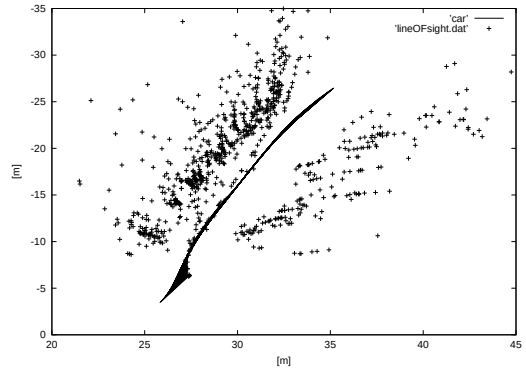

Figure 10: Line of sight crossing points

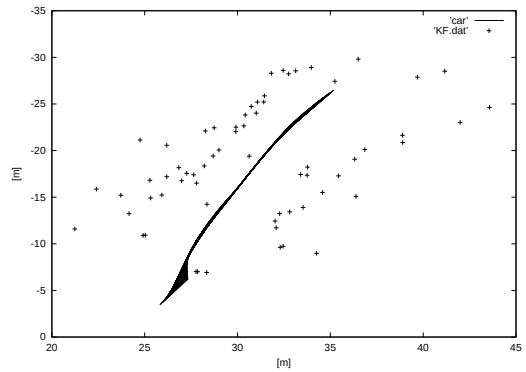

Figure 11: Kalman filter based detected landmark locations in the 2D map without using estimated landmark location feedback for template search. Note how the noisy data from Fig. 10 is turned into much cleaner data by the filtering. The points in a straight line parallel to the vehicle path correspond to the wall visible in Fig. 3.

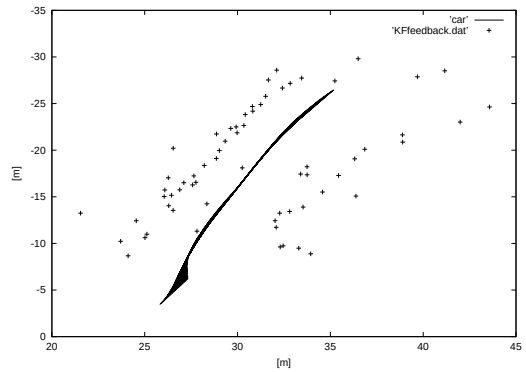

Figure 12: Kalman filter based detected landmark locations in the 2D map using estimated landmark location feedback for template search. The result of Fig. 11 are further improved here.

6 Conclusion

Toward environment recognition in city area, as a first step, we developed a method to recognize a static environment from a vehicle using an omni-directional camera and vehicle odometry. The environment recognition making two-dimensional map of static environment was achieved by using template tracking using feedback of location estimation. The experimental result on a real outdoor vehicle using proposed methods showed environmental shape and it was good as the first result.

References

[1] C. Thorpe, T. Jochem, D. Pomerleau, "Automated Highways and the Free Agent Demonstration", *International Symposium on Robotics Research,* October 1997.

[2] H.R. Everett, 1995. Sensors for Mobile Robots: Theory and Application. A K Peters.

[3] S. Nayar, S. Baker, "Catadioptric image formation", *Proc. of 1997 DARPA Image Understanding Workshop,* pp. 1431 - 1437.

[4] G. Jyo, S. Tsuji, “3D Vision (in Japanese)”, *Kyoritsu-syuppan,* 1998.

[5] Y. Yagi, H. Hamada, N. Benson, M. Yachida, “Generation of Stationary Environmental map under Unknown Robot Motion”, *Proc. of the 2000 IEEE/RSJ Int. Conf. on IROS,* pp. 1487 - 1492.

[6] D. Strelow, J. Mishler, S. Singh, H. Herman, “Omnidirectional Shape-from-Motion for Autonomous Navigation”, *Proc. of the 2001 IEEE/RSJ Int. Conf. on IROS,* to appear.

[7] P. Chang and M. Hebert “Omni-directional structure from motion”, *Proceedings of the 2000 IEEE Workshop on Omnidirectional Vision,* pp. 127 - 133.