Frame Registration of In-vehicle Normal Camera with Omni-directional Camera for Self-position Estimation

Hiroyuki Uchiyama*, Tomokazu Takahashi*,†, Ichiro Ide*, Hiroshi Murase*
*Graduate School of Information Science, Nagoya University,
Furo-cho, Chikusa-ku, Nagoya, Aichi, 464–8601, Japan
{uchiyama, ttakahashi, ide, murase}@murase.m.is.nagoya-u.ac.jp
†Japan Society for the Promotion of Science

Abstract

We propose a method for frame registration between in-vehicle omni-directional and normal cameras aimed at self-position estimation of a vehicle. We assume that the position of a vehicle is estimated by frame registration between the inputted normal camera images and the omni-directional video database with accurate position information. A DTW-based algorithm is used for nonlinear time-series matching because we consider that time-series information contributes to robustness. According to an evaluation experiment, the proposed method showed precise self-position estimation ability.

1. Introduction

High accuracy self-position estimation is needed for improvement of car navigation systems. Current systems generally estimate the self-position by GPS (Global Positioning System) combined with an odometer and gyroscope. However, in places such as a valley or under a high-level road where GPS cannot be used, estimation error gradually accumulates even if an odometer and gyroscope are used. As a solution, self-position estimation methods using in-vehicle cameras have been proposed [1]. On the other hand, there are attempts to include various types of information on maps. For example, omni-directional camera images are embedded in practical GIS services such as Google Maps [2].

Sato et al. proposed a method that improves the precision of self-position estimation by applying DTW to pairs of omni-directional camera videos aligned with the GPS coordinates [3]. Many self-position estimation methods using pairs of normal camera image sequences have been proposed. Lee et al. estimated the locations of a mobile robot without camera calibration [4]. Recently, not omni-directional cameras but normal cameras have begun to be installed in many cars. However, it is still difficult to register between two cameras without accurately knowing the posture of the cameras. Ikeda et al. proposed a method to construct a 3D feature landmark database using an omni-directional camera [5]. They estimated the position and posture of a normal camera. In order to register between two normal cameras, Ono et al. proposed to represent a space-time image as a sphere [6], and Kawasaki et al. proposed a method to use EPI (Epipolar Plane Image) [7]. These two methods are applied to video streams that are recorded during the same trip.

In this paper, we propose a method that registers a frame image from a normal camera with that from an omni-directional camera by modified DTW (Dynamic Time Warping), for self-position estimation of a car equipped with a normal camera referring to an omni-directional video database with accurate position information. In most practical cases, the in-vehicle camera is different from that for constructing the database. So, we assume a system that uses different kinds of cameras. In this paper, we assume that the direction of the normal camera is horizontal and not known to the system. We consider that time-series information reduces the computing cost and is more robust compared with the frame-to-frame matching method.

2. Frame Registration of Normal Camera with Omni-directional Camera

2.1. Overview of the Method

Figure 1 shows an overview of the proposed frame registration method. First, each image is projected onto a cylinder. Next, color histograms of omni-directional and normal camera images are normalized in order to reduce the difference of tones depending on the input devices. Then, multiple windows for matching are placed on the cylindrical sur-

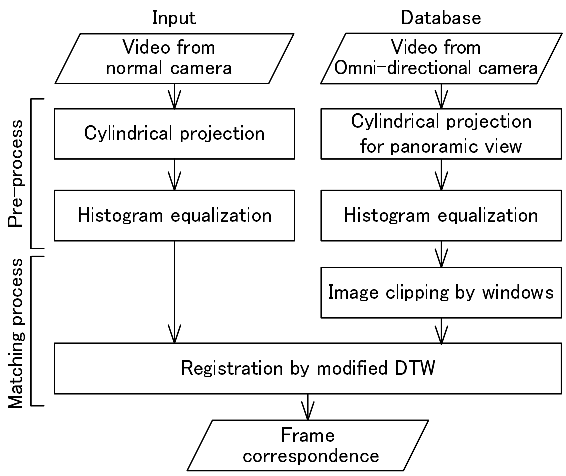


Figure 1. Flow of the proposed method.

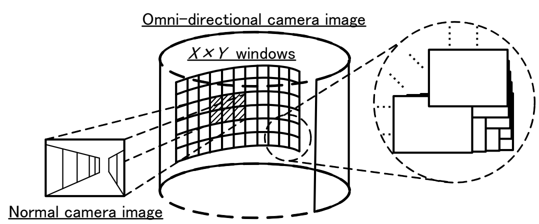
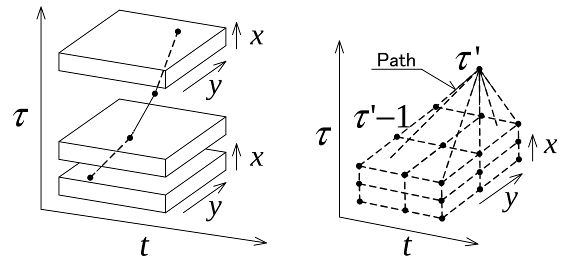


Figure 2. Images are projected onto a cylindrical surface. Multiple matching windows are placed on the surface.

face. Frames are registered by extracting correspondences between the matching windows in two videos. A modified DTW method is applied for the frame registration. Following are details of the proposed method.

2.2. Image Clipping on a Cylindrical Surface using Matching Windows

Before histogram equalization, frames from both the normal camera video and the omni-directional camera video streams are projected onto a cylindrical surface. We used Hyper Omni Vision [8] as the omni-directional camera. Since this camera consists of a hyperbolical mirror, which provides a perspective view, it can be treated the same as a normal camera. Assuming the track of focus of the hyperbolical mirror is located on one of the optical centers of the normal camera, matching can be accomplished only by parallel translations on the cylindrical surface. Next, matching windows of the same size as the normal camera image are placed on the omni-directional image. They are overlapped with slightly different positions (Figure 2). The number of windows in the horizontal and vertical positions are X



(a) Four-dimensional calculation space used in the proposed method. **(b) DP path: Optimal path is selected from 27 paths.**

Figure 3. Modified RSM algorithm.

and Y, respectively. A normal camera image and an image clipped by a window from an omni-directional image are compared by the SAD (Sum of Absolute Difference) of the brightness of each pixel. Here, we assume that the intrinsic parameters of the two cameras are known.

2.3. Frame Registration by Modified DTW

A normal camera video $V_{normal}(\tau)$ and omni-directional camera video clipped by the windows $V_{omni}(x, y, t)$ are represented as one- and three- dimensional data series, respectively. Sakoe et al. proposed a method for nonlinear pattern matching between one- and two- dimensional series based on DP [9] called RSM (Rubber String Matching). In this approach, a lower cost path is selected locally in three-dimensional calculation space and then the lowest distance (or highest similarity) is calculated globally. We extend this method in order to apply it to the matching between one- and three- dimensional data series for the purpose of corresponding a normal camera video with an omni-directional camera video (Figure 3). In this case, the calculation space is four-dimensional. In this paper, DP paths shown in Figure 3 (b) are used, where the path that has the lowest cost is selected from the 27 paths. Following is the algorithm to estimate the position of the optimal window $w(x_{est}, y_{est})$ and the t_{est}-th frame, which correspond to the τ-th frame of the normal camera video from T frames of the omni-directional camera video database.

Local distance

$$d(x, y, t, \tau) = \mathrm{SAD}\{V_{omni}(x, y, t), V_{normal}(\tau)\} \qquad (1)$$

Initial condition

$$\begin{cases} g(-1, y, t, \tau) = g(x, -1, t, \tau) = \infty \\ g(X, y, t, \tau) = g(x, Y, t, \tau) = \infty \\ g(x, y, -1, \tau) = g(x, y, -2, \tau) = \infty \end{cases} \qquad (2)$$

Iteration ($\tau = 0, 1, \dots$)
for $\tau = 0$

$$g(x, y, t, 0) = d(x, y, t, 0) \tag{3}$$

for $\tau \geq 1$

$$g(x, y, t, \tau) = \min_{\substack{-1 \leq \Delta x \leq 1 \\ -1 \leq \Delta y \leq 1 \\ 0 \leq \Delta t \leq 2}} \{g(x - \Delta x, y - \Delta y, t - \Delta t, \tau - 1) + d(x, y, t, \tau)\} \tag{4}$$

Output ($\tau = 0, 1, \dots$)
Estimated parameters x_{est}, y_{est} and t_{est}, where,

$$g(x_{est}, y_{est}, t_{est}, \tau) = \min_{\substack{0 \leq x \leq X-1 \\ 0 \leq y \leq Y-1 \\ 0 \leq t \leq T-1}} g(x, y, t, \tau) \tag{5}$$

Local distance is defined as Eq. (1). In the initial condition, the outside of the calculation space is set to ∞. For each input image, Eq. (3) or (4) is calculated iteratively, and then the parameters are estimated.

Because Eq. (4) requires an enormous calculation cost, we introduced some restrictions. First, because the position of a vehicle gradually changes, we can define that the search space in the omni-directional database is around the estimated frame number. So, $g(x, y, t, \tau)$ is calculated in the range of $t_{est} - N_{search} \leq t \leq t_{est} + N_{search}$, where, N_{search} is a threshold. Second, we prepare flags for each window, and $g(x, y, t, \tau)$ that corresponds to the window is calculated when the flag is true. Windows with small $g(x, y, t, \tau)$ tend to become the optimal window in the next iteration. So, $g(x, y, t, \tau)$ is sorted by ascending order and the flags of windows smaller than the N_{sort}-th window are set to false. In addition, assuming the position of the optimal window shifts gradually, a window near the optimal window is considered as a candidate of the optimal window in the next iteration. Then, the flag of the window near the optimal window is set to true.

3. Experiment

3.1. Condition

We conducted an experiment using in-vehicle camera videos recorded along a street to demonstrate the effectiveness of the proposed method. An omni-directional camera (Vstone VS-C14N) and a normal camera (Panasonic NV-GS150-S) were attached to the roof and the front window, respectively. The frame rate was 30 [frames/sec]. For evaluation, accurate position information was embedded into the omni-directional video by comparing the features (corners of buildings and utility poles) in the omni-directional video and aerial photography obtained from Google Maps [2], then interpolating between them linearly. The normal camera video was posed to the directions of 1) the front of the vehicle, 2) about 15 [deg] left of the front and 3) about 35 [deg] right of the front, and the length along the street was approximately 400 [m]. The omni-directional video database and the normal camera videos were recorded in different trips. Only the top half of the normal camera image was used because there were regions of roads and hoods that would not contribute to matching in the bottom. In addition, only the front 180 [deg] of the omni-directional video image was used. The initial condition t_{est} was given manually and the thresholds were determined by pilot experiments.

Table 1. Estimation error.

Direction	Estimation error
Front	5.1 frames (1.8 m)
Front left	1.3 frames (0.5 m)
Front right	3.1 frames (1.1 m)

3.2. Results and Discussion

For the inputted normal camera frames, the average error between the estimated frame and the ground truth are shown in Table 1. The highest accuracy was obtained from the front left camera, followed by the right front and front. The proposed method outperformed the conventional GPS, which has 5–30 [m] error. Figure 4 shows examples of frames registered correctly for each direction.

Errors for the front and the front right angles were higher than those of the front left. In the case of the front angle, poor variation of appearance of distant scenery caused the registration error. In the case of the front right, occlusions from oncoming traffic caused some outliers (Figure 5 (d)). However, just a few frames after the registration error occurred, correct registration was restored.

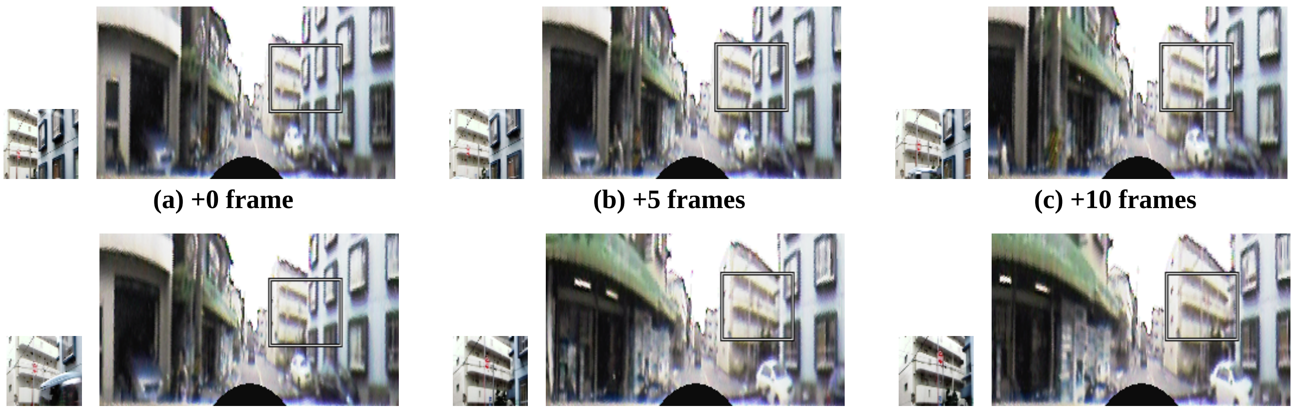
The processing speed was 10.6 [frames/sec] with a PC (Core2Quad Q6600 2.4GHz, 4.0GB). We expect that we can achieve real-time processing by improving the implementation and adjusting the parameters.

4. Conclusion

In this paper, we proposed a method for frame registration between an in-vehicle omni-directional camera and a normal camera to achieve self-position estimation using images. The registration process was conducted by projecting omni-directional video to a cylindrical surface. A DTW-based algorithm was used for the alignment process. The experimental results showed high enough accuracy to be used for self-position estimation of a vehicle. In the future, we will improve robustness against occlusion and change in streetscapes due to construction and reconstruction of buildings.

(a) Front (b) Front left (c) Front right

Figure 4. Result of the registration. The image to the left is a normal camera frame and to the right is the estimated omni-directional camera frame. The estimated position is marked by a rectangle.



(a) +0 frame (b) +5 frames (c) +10 frames

(d) +15 frames (e) +20 frames (f) +25 frames

Figure 5. Registration error from oncoming traffic. (a)(b)(c)(e)(f): Correctly registered. (d): Incorrectly registered due to a vehicle in the bottom half region.

Acknowledgment

The authors would like to thank their colleagues for useful discussions. Parts of this research were supported by the Grant-In-Aid for Scientific Research from the Japan Society for Promotion of Science. The works are developed based on the MIST library, available at *http://mist.suenaga.m.is.nagoya-u.ac.jp/*.

References

[1] T. K. Xia, M. Yang, R. Q. Yang, "Vision Based Global Localization for Intelligent Vehicles," Proc. IEEE Intelligent Vehicles Symposium (IV2006), pp. 571–576, June 2006.

[2] Google Maps, http://maps.google.com/

[3] J. Sato, T. Takahashi, I. Ide, H. Murase, "Change Detection in Streetscapes from GPS Coordinated Omni-Directional Image Sequences," Proc. 18th Int. Conf. on Pattern Recognition (ICPR'06), Vol. 1, pp. 975–979, Aug. 2006.

[4] W. Lee, K. Roh, I. Kweon, "Self-localization of a Mobile Robot without Camera Calibration using Projective Invariants," Patten Recognition Letters, Vol. 21, Issue 1, pp. 45–60, Jan. 2000.

[5] S. Ikeda, T. Sato, K. Yamaguchi, N. Yokoya, "Construction of Feature Landmark Database using Omnidirectional Videos and GPS Positions," Proc. Int. Conf. on 3-D Digital Imaging and Modeling (3DIM), pp. 249–256, Aug. 2007.

[6] S. Ono, T. Mikami, H. Kawasaki, K. Ikeuchi, "Space-time Analysis of Spherical Projection Image," Proc. 18th Int. Conf. on Pattern Recognition (ICPR'06), Vol. 1, pp. 975–979, Aug. 2006.

[7] H. Kawasaki, A. Miyamoto, Y. Ohsawa, S. Ono, K. Ikeuchi, "Multiple Video Camera Calibration using EPI for City Modeling," Proc. Asian Conf. on Computer Vision, Vol. 1, pp. 569–574, Jan. 2004.

[8] K. Yamazaki, Y. Yagi, M. Yachida, "Visual Navigation with Omnidirectional Image Sensor HyperOmni Vision (in Japanese)," Trans. IEICE D-II, Vol. J79-D-II, No. 5, pp. 698–707, May 1996.

[9] H. Sakoe, Y. Katayama, "One Dimensional-Two Dimensional Dynamic Programming Matching Algorithm for Character Recognition," IEICE Trans. Information & Systems, Vol. E77-D, No. 9, pp. 1047–1054 Sep. 1994.