A VLSI-IMPLEMENTATION-FRIENDLY EGO-MOTION DETECTION ALGORITHM BASED ON EDGE-HISTOGRAM MATCHING

Jia Hao and Tadashi Shibata

School of Frontier Science, The University of Tokyo
Department of Frontier Informatics
7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8656, Japan
haojia@if.k.u-tokyo.ac.jp, shibata@ee.t.u-tokyo.ac.jp

ABSTRACT

An ego-motion detection algorithm compatible to hardware implementation has been developed. The algorithm utilizes local motion detection scheme based on edge-histogram matching, which enables us to detect local motions robustly in segmented blocks in a visual field. An 18-dimension motion field vector is generated by summarizing local motions. Then the vector quantization is carried out to recognize the ego-motion. In order to achieve further robustness, two thresholding techniques, block thresholding and median processing, are employed in the procedure. In computer simulation, over 93% of detecting accuracy has been experimentally demonstrated by template matching using 30 template vectors generated from each of four ego-motion types.

1. INTRODUCTION

Ego-motion detection is one of the applications of motion estimation algorithms, which is to recognize the motion of observer itself. It plays an important role in various fields, such as automotive guidance, robotics control, etc. Different from the ordinary motion detection, which is to detect and extract the motion of special objects for tracking, ego-motion needs to determine the global motion of the visual field in order to recognize the motion of the observer. It is computationally expensive, thus a number of algorithms are proposed aiming at realizing a real-time ego-motion [1] - [3]. However, most of these algorithms in literature involve operations in frequency and spacial domain, i.e., estimating the motion models by solving some complex equations with floating-point calculations. Since the circuitry needed for the calculation of floating-point numbers is very complicated, it is difficult to implement these algorithms as a very large scale integration (VLSI) system.

The purpose of this work is to present a hardware friendly algorithm for ego-motion detection, which employs the projection histograms of extracted edge information. Directional-Edge-Based-Matching scheme is proposed in order to detect local motions in visual field. Two types of thresholding techniques, block thresholding and median processing, are introduced. As a result, robust ego-motion detection is achieved. All operations are executed with integer or 1-bit calculation, namely fixed-point operation, thus it is easy to implement the algorithm with compact hardware circuitry.

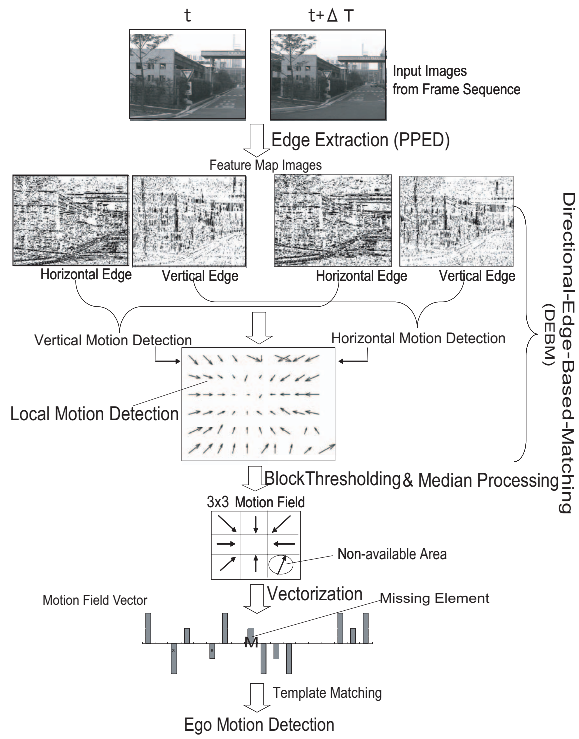

Fig. 1. Flowchart of Proposed Algorithm

To evaluate the performance of the proposed algorithm, four types of ego-motion detection, i.e., vertical and horizontal moving, zooming and rotation, are simulated, and over 93% of detection accuracy is successfully demonstrated.

2. EGO-MOTION DETECTION METHOD

Fig. 1 shows the flowchart of the proposed ego-motion detection algorithm. In this algorithm, the input data are images in a sequence. First, to detect local motions in a visual field, Directional-Edge-Based-Matching (DEBM) scheme is conducted. DEBM includes two steps, edge extraction and local motion detection. Edge extraction step is carried out for each frame in the sequence according to the algorithm described in [4]. As a result, horizontal and vertical edge maps

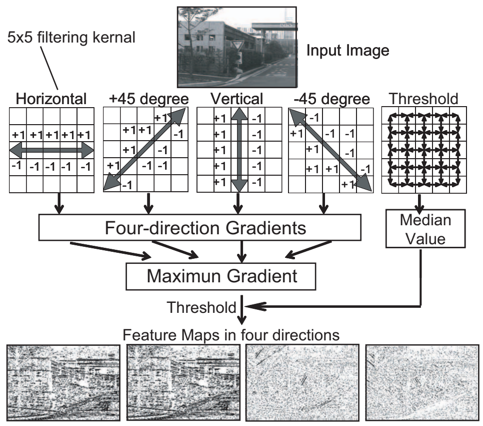

Fig. 2. Edge Extraction Process and Feature Maps Generation

are generated. Then, two edge-map images corresponding to continuous frames are utilized to detect *local motion*. Local motion is the motion inside a small segmented block in the visual field. Block thresholding is used to filter the local motion of the block including few edges, which is regarded as invalid. After that, valid local motions are summarized into nine areas in a 3x3 form. Median processing of local motions is operated to each area in order to generate area motion. Area having few local motions is regarded as a non-available area. Motion field is composed of 3x3 area motions. In the vectorization of motion field, x and y components of area motions are represented as separate elements. The elements corresponding to the non-available area are called *missing elements*. Finally, by matching the current motion field vector with the template motion field vectors prepared in advance, ego-motion in the image sequence is detected.

2.1. DIRECTIONAL-EDGE-BASED-MATCHING

2.1.1. EDGE EXTRACTION

Edge extraction process is executed to improve the accuracy of motion detection. In this process, essential edge features are retained very well, while unexpected noise is subtracted to a considerable extent. Fig. 2 illustrates the operation. Firstly the input image is filtered spatially pixel-by-pixel by using four kinds of filtering kernels of 5x5 pixel size. As a result, edge gradients in four directions are obtained, i.e., horizontal, vertical, +45 degrees, and -45 degrees. Then the edge direction of each pixel is decided by the maximum gradient. In addition, the local variance of intensity data in image is taken into account in threshold processing. Namely, the median of the values of neighboring pixel intensity differences in the 5x5-pixel filtering kernel is defined as threshold. It is utilized to decide whether the edge flag gradient of the current pixel is large enough to represent essential feature in the image or not. Only the gradients exceeding the threshold value are retained. Then four feature maps including the spacial distribution of edge flags are produced, as shown in Fig. 2.

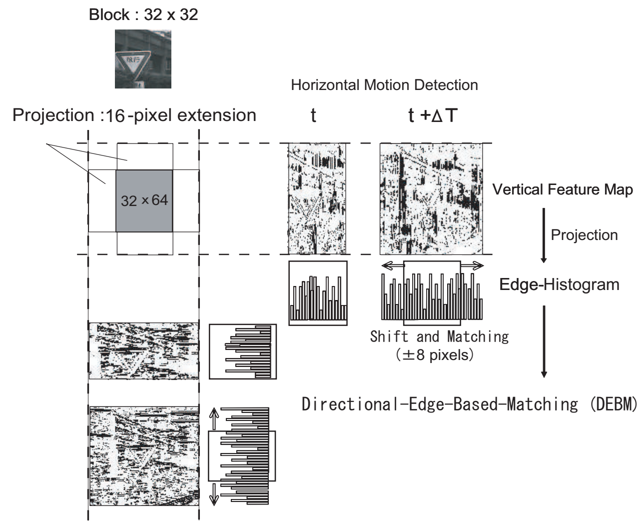

Fig. 3. Local Motion Detection

2.1.2. LOCAL MOTION DETECTION

To detect the motions in different parts of one image, firstly feature maps are segmented into several blocks. Local motion is defined to be the motion of each block. Fig. 3 depicts the principle of local motion detection. The block size in simulation is 32x32 pixel. The local motion detection is carried out by shift and matching. Content of current block in the first image is searched in the second image.

Projection is an effective technique for local motion detection. It reduces the amount of data dimensionally and also improves the accuracy. For instance, in the case of detecting local motion in x direction, feature map is projected onto horizontal axis. Even if there is also motion in y direction, the rough shape of the projection histogram will not change rashly, therefore the accuracy is well maintained.

In order to improve accuracy further, during projection and matching, block is extended ± 16 pixels in the same direction of projection to include more feature information into edge histogram. Then the block size becomes 32x64 pixel, as shown in Fig. 3, overlapping with neighboring blocks.

If original intensity images are utilized, the essential features of the objects are not represented well by the projection histogram, since it is difficult to extract features based on intensity data. On the contrary, the projection histogram of edge feature maps represents the original image features in the proper level. Moreover, the horizontal and vertical feature maps can be utilized separately to efficiently exploit the directional edge information, as shown in Fig. 2. Then, based on these edge-histograms, Manhattan distance between the consequent frames is calculated. The minimum distance represents the best match. As illustrated in the figure, ± 8 pixels is shifted horizontally from the initial location of the current block in order to detect maximum ± 8 pixels local motion.

2.2. BLOCK THRESHOLDING

Block thresholding is the postprocessing scheme of local motion detection results for noise reduction. Fig. 4 describes

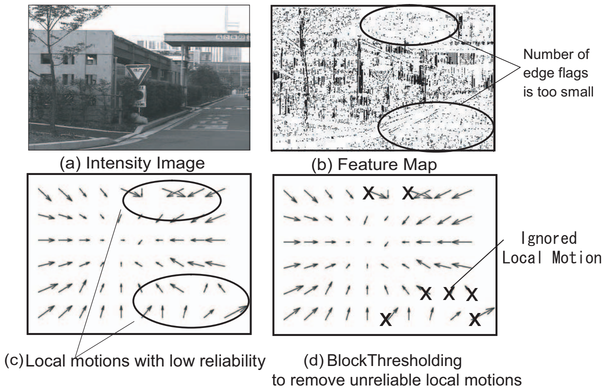

Fig. 4. Block Thresholding process

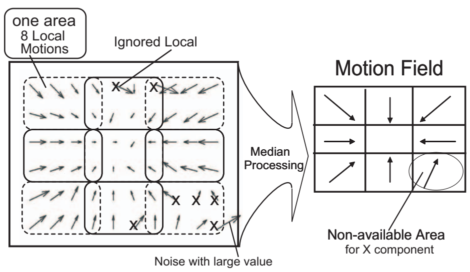

Fig. 5. Motion Field Generation

the principle of this process. Let's think the process on vertical feature map for instance. In the blocks having few edge flags, local motion may not be detected correctly, as shown in Fig. 4(b) and (c). If the edge flags in a block are less than a limit percent of the total pixel number, the local motion of this block is regarded as having low reliability and ignored. These local motions are called invalid local motions. According to the simulation results, 5 % is determined to be the limit percent. The block thresholding result is illustrated in Fig. 4 (d). X marks mean that the local motions with low reliability are ignored. Apparently, this operation removes the effects of negative factors in the background, i.e., sky, wall surface and wide road, which have plain textures and thus, give poor edges, and ineffective DEBM results.

2.3. MOTION FIELD GENERATION

Motion field represents more global motion detected in one frame. Fig. 5 depicts the generation process. In the proposed method, firstly motion field is summarized into a 3x3 form. Namely, one frame is partitioned into nine areas which may overlap with each other. Each area includes eight local motions. Then median processing is executed for local motions in this area in order to generate the motion representative of this area. Finally, the motion field is built up by combining these nine area motions.

Median processing is executed by choosing the median value among the valid local motions to represent the motion of the area. Compared to average processing, median processing has satisfactory performance when noise is strong. Area

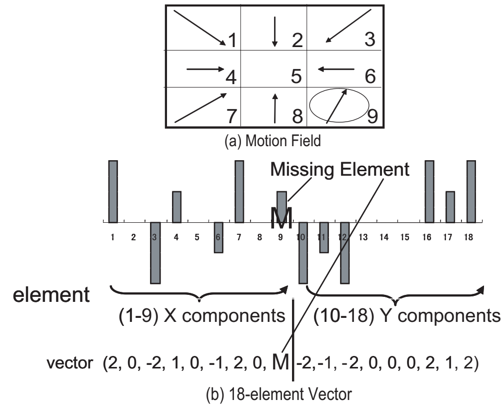

Fig. 6. Motion Field Vectorization

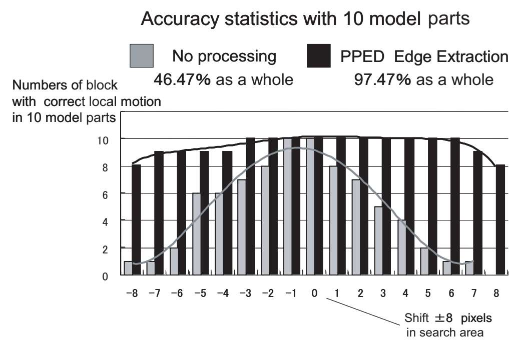

Fig. 7. Accuracy Comparison between DEBM Method and Intensity-based Projection Method

including more than three invalid local motions is regarded as a non-available area, where three is determined by simulation. This availability of area is essential in the next motion field vectorization step.

2.4. VECTORIZATION FOR FRAME MATCHING

Vectorization of motion field is to transform the two-dimension motion field into an 18-dimension vector, which represents the motion information of one frame in a simpler way.

Fig. 6 describes the principle for vectorization. Areas are arranged in the order shown in Fig. 6 (a). Then the x components of the nine area motions are assigned to the nine elements on the left side of the vector also in this order, and y components on the right side, as shown in Fig. 6 (b). In addition, the area, which is regarded as non-available in the previous processing, is marked as $missing\ element$.

2.5. MATCHING AND EGO-MOTION DETECTION

Various motion vectors for panning, zooming and rotation scenes are generated and sorted into groups of different motion patterns, then stored in memory in advance. The matching process is carried out by calculating Manhanttan Distance between the current target motion field vector and the vectors in memory. Here, in the frame level matching computation, if

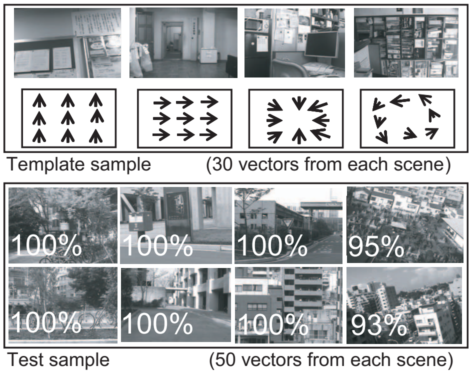

Fig. 8. Matching experiment for ego-motion detection

there is missing element in one pair of comparable elements from two motion vectors, then the difference of this pair is not included in the distance computation for frame matching. The distance is normalized, i.e., divided by the number of elements utilized for distance computation and multiplied by the total element number 18. According to the minimum of distance, the current vector is categorized into a group with which it gets the minimum Manhattan distance. The ego-motion type of this frame is recognized. Also, sequential ego-motion can be recognized based on these matching results.

3. EXPERIMENTAL RESULTS

To evaluate the performance of the proposed algorithm, the simulation results are presented in this section. Fig. 7 shows the comparison of the local motion detection accuracy between DEBM and conventional intensity method without edge detection. In order to examine the impact of using edge feature maps, one static image is utilized in the simulation, where the second frame is imitated by shifting gradually to ± 8 pixels in y direction. The local motion detection in x direction is computed. Accordingly, the value of zero is regarded as a correct detection, while the background is moving vertically. X-axis of the figure indicates the number of pixel the image should shift to imitate ego-motion, in the range of ± 8 pixels. Y-axis is the number of correct local motion within 10 samples blocks. In the result of non-preprocessing of edge detection, the accuracy evidently decreases when the frame is shifted over 4 pixels. On the other hand, proposed DEBM method achieves the accuracy up to 97.47% as a whole.

Fig. 8 shows the simulation results of ego-motion detection. Scenes of four types of ego-motion, moving vertically and horizontally, zooming, and rotation, are sampled and vectorized into template vectors. 30 vectors are generated from each scenes and stored into the memory. The motion vector sampled from different scene is compared to the templates, then the pattern of the ego-motion is determined. The simulation results successfully demonstrated 100% accuracy for moving vertically, moving horizontally and zooming, where 95% to 93% accuracy for rotation.

Regarding to the implementation of the proposed algorithm, most of the operations consist of simple integer calculations, thus a hardware accelerator can be designed easily with compact digital circuitry. Moreover, although the time-consuming steps, such as pixel-by-pixel edge extraction and median processing, exist in the process, dedicated hardware processors can be used to effectively execute them, which have been already developed by our groups [5], [6]. It is demonstrated that the feature maps generation according to the algorithm in [4] is accelerated by five orders of magnitude as compared to the software processing with Pentium4 (2.2GHz), only at a clock frequency of 100MHz [6].

4. CONCLUSIONS

A hardware-implementation-compatible ego-motion detection algorithm has been developed. Edge-histogram based local motion detection is employed. The robust local motion detection is successfully achieved. Block thresholding and median processing are exploited to noise reduction. The 18-dimension motion field vector is generated to represent the ego-motion. Finally, ego-motion detection is accomplished with template matching based on vector quantization algorithm. Performance of the proposed algorithm is verified by software simulation. Over 93% of detection accuracy for ego-motion detection is successfully achieved.

5. ACKNOWLEDGEMENT

The author would like to thank Kiyoto Ito for his invaluable comments and advices; Tongprasit Benjamas, Ozturk Ovgu and all the members in Shibata lab for their unconditional supports.

6. REFERENCES

[1] S. Kumar, M. Biswas and T. Q Nguyen, "Global Motion Estimation In Freqency And Spatial Domain," *in the Proceeding of 2004 IEEE International Conference on Acoustic, Speech, and Signal Processing (ICASSP 2004)*, pp. III-333-III-336, May, 2004.

[2] W. Burger and B. Bhanu, "Estimating 3-D Egomotion from Perspective Image Sequences," *IEEE Trans. PATTERN AND MACHINE INTELLIGENCE*,Vol. 12, no. 11, pp.1040-1058, Nov. 1990.

[3] J. K. Aggarwal and N. Nandhakumar, "On the Computation of Motion from Sequences of Images-A Review," *in the Proceeding of the IEEE*, Vol. 76, no. 8, pp.917-935, August, 1988.

[4] M. Yagi and T. Shibata, "An Image Representation Algorithm Compatible With Neural-Associative-Processor-Based Hardware Recognition Systems," *IEEE Trans. Neural Networks* ,Vol. 14, no. 5, pp.1144-1161, Sep. 2003.

[5] H. Yamasaki and T. Shibata, "A Real-Time Image-Feature-Extraction and Vector-Generation VLSI Employing Arrayed-Shift-Register Architecture," *in the Proceeding of the 31th European Solid-State Circuit Conference (ESSCIRC)*, Grenoble, France, pp.121-124, Sep. 2005.

[6] H. Yamasaki and T. Shibata, "A High-Speed Median Filter VLSI Using Floating-Gate-MOS-Based Low-Power Major Voting Circuits," *in the Proceeding of the 31th European Solid-State Circuit Conference (ESSCIRC)*, Grenoble, France, pp.125-128, Sep. 2005.