IAPR Workshop on CV - Special Hardware and Industrial Applications OCT.12-14, 1988, Tokyo

REAL-TIME TRACKING SYSTEM OF MULTIPLE LINE SEGMENTS FOR MOTION STEREO

Shigeru Kimura, Shinji Ozawa

Faculty of Science and Technology, Keio University
3-14-1, Hiyoshi, Kohoku-ku, Yokohama 223, Japan

ABSTRACT

This paper describes a new computer vision architecture specialized for motion stereo applications, which has the capability of tracking multiple line segments concurrently in real time. We have developed new algorithms optimized for tracking line objects, and implemented them in hardware. The first-step prototype system operates on 6-MHz video data (256[H]*240[V]-pixel images) and performs tracking of 8 line segments successfully at the video rate (30 frames/s.). The number of line segments the system tracks can be increased by connecting additional tracking filters. The decomposition method is the key feature in this real-time architecture. The Cut & Shrink algorithm and subpixel address generation are introduced in order to apply the decomposition method effectively.

INTRODUCTION

In computer vision, a wide variety of methods of computing 3-D structure from 2-D images have been extensively studied [1]. However, there are still many difficulties remained in practical applications. Especially, in stereo vision systems, matching tokens (points, lines , etc.) between images is the main and largest problem [2]. We try to remove this problem by tracking line segments at the real-time speed. If we could consistently keep tracks of the identical line segments in images, the problem of matching tokens can be considerably reduced or eventually omitted [3].

Recently, there are a number of general-purpose image processors available. But, they are essentially not suitable for the specific operations like real-time target tracking. For real-time target tracking, a special image processing hardware is inevitably required. Many target tracking algorithms, including the centroid tracking and the correlation tracking, have been proposed and implemented [4][5]. However, our tracking system to deal with line objects is quite new and unique.

SYSTEM ARCHITECTURE

The system is optimized especially for tracking multiple line segments concurrently in real time. Here, the term real-time means the speed of processing 30 image frames per second. In real-time systems, the values of the processing speed and delay are the very important factors. Throughout the system, the pipelined architecture is adopted in order to achieve real-time tracking. Moreover, if possible, some unpredictable looping or feedback operations are removed not to cause disorder of a pipelined data stream.

The main advantage of our system is derived from the sophisticated technique of real-time data conversion from raster image data to non-raster data (data stream of feature vectors). Thus this makes it possible to execute tracking operations quite flexibly and efficiently in the feature space. The key feature in this real-time architecture is the decomposition method described below.

Decomposition method: The Hough Transform is a well-known method for detecting colinear points in images, e.g., straight lines. But this method is extremely time consuming for line tracking. Except for the Hough Transform, the conventional way of extracting line segments from image is usually the linkage based approach. In this approach, the edge fragments which locate near and have the similar angles are gradually connected together to construct a line segment. If we follow this approach, the mechanism of data accesses can be too complex to implement in hardware and makes it difficult to perform real-time processing.

On the other hand, the approach we take here is completely different. In our approach,

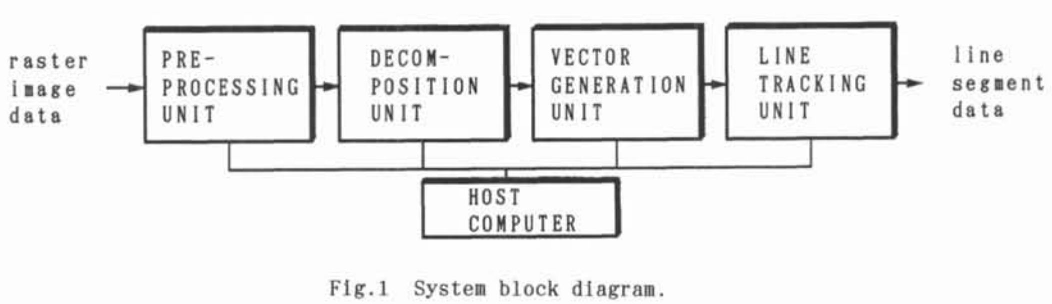


Fig.1 System block diagram.

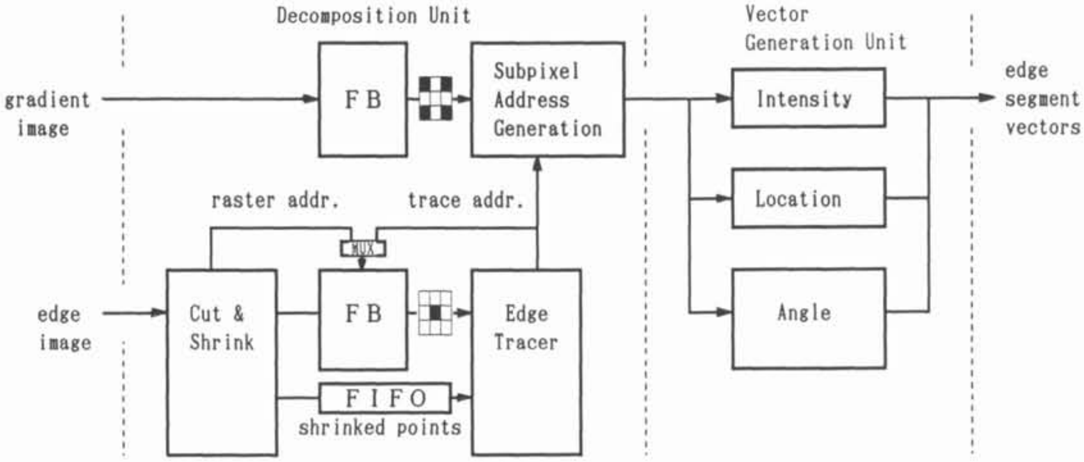


Fig.2 Decomposition Unit and Vector Generation Unit.

referred to as Decomposition Method, the connected edges are decomposed into the small edge segments of the unit length which represent the local edge structures. After converting these edge segments to the feature vectors, the line segments are built up by the filters in the feature space. This method offers to the architecture much of flexibility. And it is applicable not only for line tracking, but for detecting and handling structural objects effectively. For example, the Hough Transform can be easily implemented, if a small modification is made in this architecture.

The system consists of the four processing units (Pre-processing, Decomposition, Vector generation, and Line tracking). The system block diagram is shown in Fig.1. The first-step prototype system operates on 6-MHz video data (256[H]*240[V]-pixel images). A data stream of one image frame runs through all the processing units within one frame period (33 ms).

Pre-processing unit: This unit performs the low-level raster operations mainly for edge detection. The gradient and edge images are outputs of this unit. The gradient images are calculated by the Sobel operator (3 * 3). The edge images are obtained through the 4 stage operations of noise reduction and thinning after the binarization of the gradient images.

Decomposition unit: The decomposition unit decomposes the connected edges of the binary edge image into the edge segments which represent the local edge structures. The edge segments hold the information about the subpixel addresses and gradient values of the pixels contained in the edge segments. The number of pixels contained in the edge segment is a constant depended on the characteristics of images (usually 2 to 4).

Vector generation unit: This unit generates the edge segment vector that is a set of the features (intensity, location, and angle). The edge segment vector represents the edge segment in the feature space and is generated from the gradient values and subpixel addresses of the pixels in the edge segment.

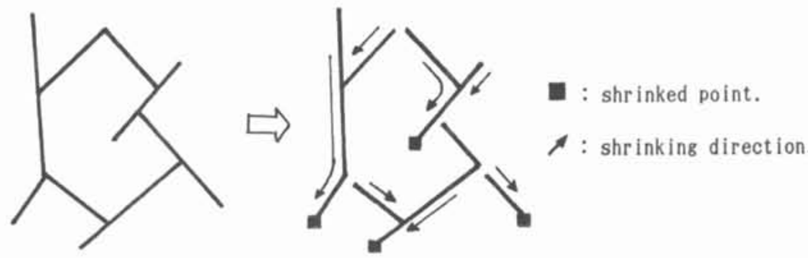
Line tracking unit: This unit performs tracking operations for the multiple line segments. The line tracking unit is composed of a number of line tracking filters. Tracking for a single line segment is executed by one of the tracking filters. The line tracking filter synthesizes the identical line segment from a stream of edge segment vectors.

DECOMPOSITION

The decomposition unit extracts the direct addresses of the edge segments from the edge image and converts them to the subpixel addresses by using the adjacent gradient values of the edge pixels. Fig.2 shows the block diagram of the decomposition and vector generation unit. The major tasks of the decomposition unit are the Cut & Shrink operation, edge tracing, and subpixel address generation.

First, the starting addresses for edge tracing are detected and pushed into FIFO(First-In-First-Out buffer) by the Cut & Shrink operation, and the processed binary edge image is stored in the frame buffer. These operations are executed in the manner of raster scan. On the other side of the frame buffer, the edge tracer tracks the edge pixels by checking the 8 adjacent pixels and divides the traced edge pixels into the edge segments. For all edge segments, the subpixel address generations are executed.

Cut & Shrink algorithm: The Cut & Shrink operation is a simple but powerful primitive to avoid backtracking and secure the uniformity of edge tracing. It allows us to perform the fast data conversion by tracing the edge pixels from raster to non-raster data. An overview of the Cut & Shrink algorithm is illustrated in Fig.3. The right-hand branches on the junctions forking downward are enforced to be disconnected, in parallel with the shrinking operation. The arrows in Fig.3(b) denote the shrinking directions. The lowest pixels of the connected edges become the shrinked points after the shrinking operation based on raster scan. The shrinked points are utilized as the starting points for edge tracing.



(a) Original edge image (b) Processed image

Fig.3 Overview of the Cut & Shrink algorithm.

Subpixel address generation: The direct addresses of the edge pixels are not adequate to represent the accurate the edge positions in order to calculate a set of feature vectors. Especially, the angle precision of the edge segment vectors does not satisfy the requirement for line segment tracking. To solve this problem, we introduce the subpixel addresses which represent the local edge positions with higher precision than the direct pixel addresses. The subpixel addresses are generated from the gradient values of the traced edge pixels including the 4 adjacent pixels.

TRACKING OF LINE SEGMENTS

The large problem about token correspondence can be reduced to much simpler tracking operations on the frame-by-frame basis. In this architecture, the tracking operations for line segments are executed by special filters efficiently in the feature space. This filter, called Line Tracking Filter, synthesizes the targeted line segment by filtering a stream of edge segment vectors.

For tracking multiple line segments concurrently, the line tracking unit has the distributed processing architecture constructed by a number of line tracking filters. All of the line tracking filters are identical with respect to hardware and software.

Line tracking filter: The line tracking filter is a full microprogrammable processor equipped with the ALU, MPY/ACC, LUTs (SIN,COS,1/N), Dual-Port RAM, FIFO, etc. The clock speed of the sequencer is 12-MHz. The microcode is developed on the host computer and downloaded to the line tracking filters. The line tracking filter has the separate input and output ports. Literally, it functions as a filter. It inputs and outputs a data stream of edge segment vectors, and does not change any data except the special flags related to the gate operation of the filter described below.

Line segment gate: Each line tracking filter has a Line Segment Gate implemented in firmware. This gate is utilized for the exclusive data acquisition of the edge segment vectors constructing the targeted line segment. As illustrated in Fig.4, the line segment gate is the elongated window (angle θ, width W, length L) located at the point (XG,YG) in the image frame. Data acquisition of the line tracking filter is automatically controlled by this gate. The gate accepts only a specific group of edge segment vectors which locate within the gate window and have the similar angles. From this exclusive data acquisition, the filter is able to select the target data and synthesize the targeted line segment from a large amount of diversified edge segment vectors in the feature space.

Tracking of the line segment in images is performed by sequentially updating the angle θ and position (XG,YG) of the line segment gate frame-by-frame to the best fit location with the actual line segment in images. In the present system, the length L and width W of the tracking gate are constant. The values of these parameters are determined at the initialization stage.

Tracking operations: The basic tracking operations performed for the single image frame are as follows:

1) collects the edge segment vectors accepted by the line segment gate.
2) calculates the position and angle of the targeted line segment by the least-squares estimation about the locations of the accepted edge segment vectors.
3) predicts the next position and angle of the line segment.
4) updates the line segment gate for the next frame, according to the prediction above.

Configurations: The present system is just a first-step implementation towards the practical motion stereo applications. However, The system performs tracking of 8 line segments successfully in real-time. The line tracking unit contains only 8 line tracking filters in the present configuration. They are connected together in serial, as shown in Fig.5(a). This serial connection is a basic configuration for the systems with the small number of filters. It is obviously easy to detect the mutual interferences between the line tracking filters connected in serial, by only checking the flags in the data stream which are affected by the gate operations of the preceding filters. By using this mechanism, the basic arbitration can be implemented in order not to track the same line segment between the tracking filters. In this connection, the problem concerning the delay of the data stream occurs inevitably. The delay will be increased in proportion to the number of the filters being passed. In the present system, the delay after passing through all 8 tracking filters is less than 1.0 ms.

The number of the filters implemented in the present system is relatively small compared with the number required for real application systems. For real applications, the number of line segments the system tracks can be increased by connecting additional tracking filters. We suppose that eventually the 200 - 300 tracking filters will be required for practical motion stereo applications. In this case, the serial and parallel connection in Fig.5(b) is recommended. In this connection, the serial connection units of tracking filters are connected in parallel. According to the angles, the edge segment vectors are distributed to the serial connection unit of tracking filters. Each serial connection unit is taking over the line segments whose angles belong to the assigned angle range. This serial and parallel connection allow us to enable the efficient and flexible distributed processing, without increasing the processing delay and having the mutual interferences between the units. This type of parallel processing architecture is more effective in contrast with other architectures which divide the processing load by means of the location in the image space.

CONCLUSION

We have described a new real-time architecture for tracking multiple line segments for use in motion stereo applications. The decomposition method is the key feature in this architecture. That makes it possible to execute tracking operations quite flexibly and efficiently in the feature space. The decomposition method is also applicable for detecting and handling structural objects. Therefore, the architecture we present here is suitable for various applications where structure-oriented processing is required, especially for real-time computer vision systems such as mobile robots.

REFERENCES

[1] B.K.P.Horn: "Robot Vision", MIT Press, Cambridge, Massachusetts, 1987.

[2] S.Ullman: "The Interpretation of Visual Motion", MIT Press,Cambridge,Massachusetts, 1979.

[3] William M. Wells III:"Visual Estimation of 3-D Line Segments From Motion -- A Mobile Robot Vision System", AAAI'87, pp.772-776, 1987.

[4] T.S.Hung: "Image Sequence Processing and Dynamic Scene Analysis",NATO ASI Series F:Computer and System Science No.2,Springer Verlag,1983.

[5] V.Cappellini:"Time-Varying Image Processing and Moving Object Recognition",Elsevier Science Pub.,1987.

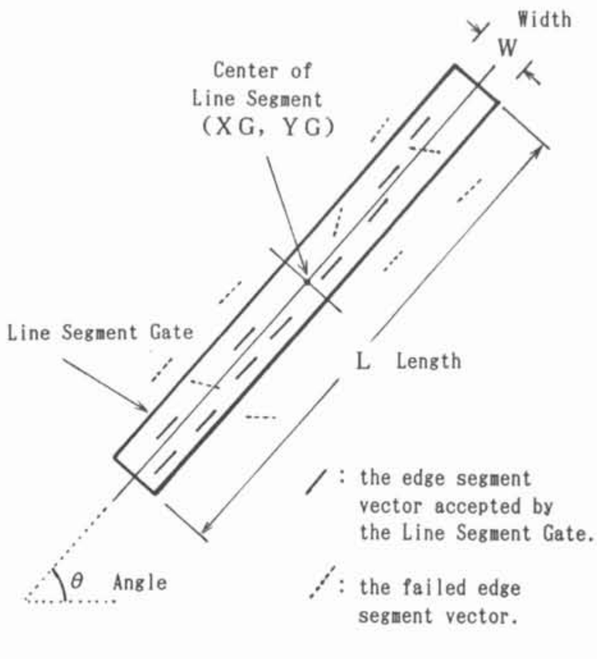


Fig.4 Functions of the Line Segment Gate.

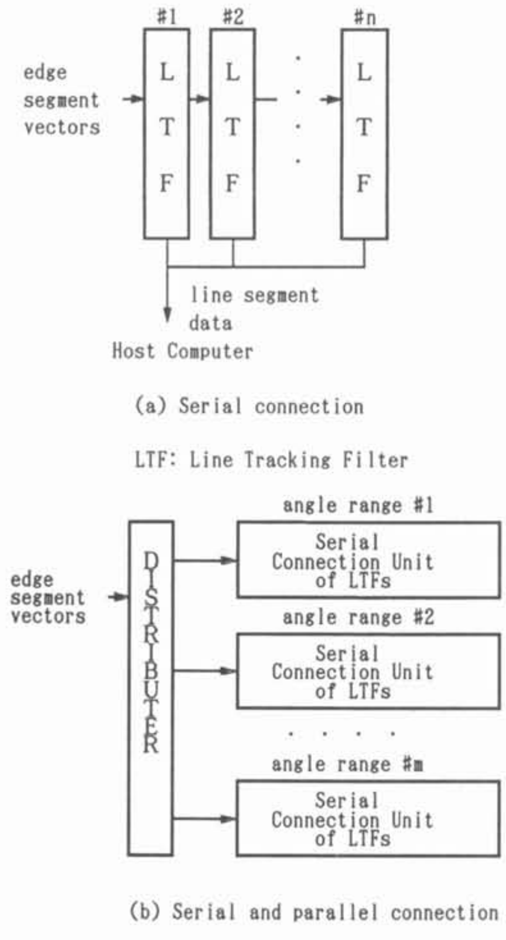


Fig.5 Basic configurations of the Line Tracking Unit.