論 文 の 要 旨

氏　　名　　　関　俊剛

論 文 題 目　Research on Hardware Implementation of Straight-Line Detection Based on Hough Transform for Real-Time Applications
（リアルタイムアプリケーション用ハフ変換に基づく直線検出のハードウェア実装に関する研究）

An efficient hardware-implementation architecture for straight-line detection is developed in this thesis. Straight-line detection is an important part of image-analysis techniques in computer vision and image processing. Its aim is to recognize the straight-line coordinates and thus provide the information (e.g. location) of straight-line-characterized objects (e.g. road lanes) in an image.

The Hough Transform (HT) has good stability for the purpose of straight-line detection, as it is robust against many problems like line gaps or noise in real-world applications. The HT can be viewed as an evidence-gathering approach in an accumulator array followed by a final voting process on this evidence. It defines a mapping process from the Cartesian coordinate (x, y) space to the polar coordinate (θ, ρ) Hough space by the function $\rho = x \cdot cos\theta + y \cdot sin\theta$, where ρ is the shortest distance from the origin to the straight line, while θ is the angle between the x-axis and the vector orthogonal to the line. For each straight line in an image, there is a distinct coordinate in Hough space corresponding to it. Due to its algorithm characteristics, the HT has two main practically application-limiting characteristics:

1) High computational complexity.
2) Large memory usage.

Hardware implementation for highly complex algorithms can improve the computational performance by exploiting parallelism. In this thesis, for $sin\theta$ and $cos\theta$ determination, a look-up-table (LUT) solution without any accuracy loss is proposed. These LUTs for $sin\theta$ and $cos\theta$ replace the time-consuming runtime computation. The stored $sin\theta$ and $cos\theta$ fractional values are scaled by a certain factor, and two's complement notation is used. After computing the HT equation, the least significant result bits are truncated to obtain the same number of bits as applied for the ρ discretization. The computing unit for (ρ, θ) is divided into n parallel parts with additional pipeline architecture based on the chosen number of discrete θ-values of the Hough space. The complete LUTs for $sin\theta$ and $cos\theta$ are distributed across the local LUTs of the parallel parts. The parallelisms n is kept flexible in the developed architecture through the usage of counters. Corresponding to n computing units, the Hough space is also divided into n parallel modules. The LUT solution for computing $sin\theta$ and $cos\theta$ leads to regular increments of θ $(\Delta\theta)$, so that the 2-dimensional Hough space can be mapped onto a 1-dimensional array. As a result, the determination of ρ and the voting on a potential straight-line location (ρ, θ) in Hough space can be implemented as a pipeline without address conflicts.

The voting procedure of the HT is implemented by increasing the vote values at the corresponding locations (ρ, θ) in the Hough space and by comparing these vote values with a pre-defined threshold

to identify straight lines. In this thesis, a parallel peak-searching unit is designed, which is associated with the voting procedure. The (ρ, θ) pair for an identified straight line is outputted when the vote value at a Hough-space location becomes larger than the pre-defined threshold. At the same time, the stored vote value at this Hough-space location is reformatted to zero, in preparation of the straight-line detection process for the next image frame. Since the Hough space cannot be initialized by one control signal, the initialization of the Hough space becomes one challenge for the hardware implementation of HT in real-time applications. In this work, a parallel initialization method with double clock operation is implemented, which initializes the Hough space during the voting procedure. The initialization of the Hough space is hidden by execution in the background of the actual HT processing and has normally no effect on the speed performance of the HT in video-sequence processing.

The threshold value method (TVM) is the most popular way to find the polar coordinates in the Hough space for determining straight lines. All values above the threshold are interpreted to represent straight lines. Owing to the quantization error caused by resolution parameters $\Delta\rho$ and $\Delta\theta$, the votes are usually distributed over a small area around the desired peak point. Unfortunately, a static threshold cannot perfectly reproduce the original straight lines, because of the many interfering straight lines which are often mistakenly generated around the real straight line. Therefore, the TVM is combined with a local-maximum approach in this work. The local maximum represents a maximum vote value within a certain neighborhood of the Hough space. In this work, the neighborhood is defined by a window of size $A \times A$, the Hough space is divided into many overlapped sub-windows of the same size, and then all of the sub-windows are traversed to find the local maximum values individually. The process for finding the polar coordinates (peak points) of the straight lines must be carried out after the voting procedure for a given image frame, which normally leads to additional time losses in the straight-line-detection process. In this thesis, a local maximum searching module is implemented to follow the voting procedure in parallel. Instead of traversing all of the sub-windows of the Hough space, the neighborhood comparison during the local maximum searching is only carried out for the sub-windows containing peak-points which indicate potential straight-lines. The centers of these sub-windows are then updated with the determined local peak-points. After the voting procedure for one image frame is finished, the peak-points corresponding to the real straight-lines are thus already extracted without additional time loss.

A prototype system of the developed straight-line detection architecture without the local maximum searching module has been implemented on the high-performance FPGA platform DE4-230-C2, which realize the achievement of video-based straight-line detection with a speed of 185 frames/s for XGA (1024×768 pixels) videos. This demonstrates the practical usability in time-critical real-time applications like lane detection. A second prototype system, which includes a local maximum searching module, has been implemented on the low-cost FPGA platform DE1-SoC and can process a VGA size (640x480 pixels) video at 60 frames/second with an operation speed of 135 frames/s. During their real-time verifications, both prototypes have been successfully demonstrated to satisfy the requirements of accuracy and operation speed for real-time applications such as lane detection.