Hierarchical Additive Hough Transform for Lane Detection

Ravi Kumar Satzoda, Suchitra Sathyanarayana, Thambipillai Srikanthan, *Senior Member, IEEE*, and Supriya Sathyanarayana

***Abstract*—Detection of lanes is an important problem in the upcoming field of vehicle safety and navigation, for which linear Hough transform (HT) is widely used. In order to meet real-time requirements, various attempts to accelerate the HT have been proposed in the past, including hierarchical HT. Existing hierarchical approaches involve the overhead of recomputing HT values at every level in the hierarchy. In this letter, we propose a novel, computationally efficient hierarchical HT by extending and applying the additive property of HT to multiple levels of hierarchies. This proposed approach, called hierarchical additive Hough transform (HAHT) is shown to lead to significant computational savings of up to 98–99% in the Hough voting process. The HAHT has been validated on a wide range of straight lane images and it is shown to successfully detect lanes.**

***Index Terms*—Hierarchical Hough transform, lane detection.**

I. Introduction

VISION-BASED automotive safety is becoming increasingly popular with applications such as lane departure warning [1], road-sign marking identification [2], etc. Lane detection is a vital operation in most of these applications as lanes provide important information like region-of-interest, for further processing. In most cases, lanes appear as well-defined, straight-line features on the image (especially in highways), or as curves that can be approximated by smaller straight lines.

The linear Hough transform (HT), a popular line detection algorithm, is widely used for lane detection [3]. The HT [4] is a parametric representation of points in the edge map. It consists of two steps—"voting" and "peak detection." In the process of voting, every edge pixel $P(x,y)$ is transformed to a sinusoidal curve by applying the following:

$$\rho = x\cos\theta + y\sin\theta \qquad (1)$$

where ρ is the length of the perpendicular from the origin to a line passing through (x,y), and θ is the angle made by the perpendicular with the x-axis. The resulting $\rho-\theta$ values are accumulated using a 2-D array, with the peaks in the array indicating straight lines in the image. Peak detection involves analysis of the $\rho-\theta$ array to detect straight lines. More details on HT can be found in [5], [6].

Manuscript received April 04, 2010; accepted May 04, 2010. Date of publication May 27, 2010; date of current version June 18, 2010. This manuscript was recommended for publication by Z. Haibo.

The authors are with the Center for High Performance Embedded Systems, Nanyang Technological University, Singapore, 637553, Singapore (e-mail: ravi.satzoda@gmail.com; suchi.sathya@gmail.com; astsrikan@ntu.edu.sg; supriya.sathya@gmail.com).

Color versions of one or more of the figures in this letter are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LES.2010.2051412

The high computational time incurred by conventional Hough voting, attributed to the trigonometric operations and multiplications in (1) applied to every pixel in the edge map, makes it unsuitable for direct use in lane detection, which demands real-time processing. Hierarchical pyramidal approaches have been proposed in [7]–[9] to speed up the HT computation process through parallelism. These hierarchical approaches in [7]–[9] filter candidates to be promoted to the higher levels of hierarchy by thresholding the accumulation spaces. For each candidate that qualifies, they perform a complete HT computation again using (1). Hence, although the hierarchical approaches speed up the HT by parallelizing the process, additional costs are incurred for recomputing HT at every level. These increased computational costs are not desirable in embedded applications like lane detection in vehicles, where computational resources are limited.

In this letter, we propose a novel hierarchical approach to accelerate the HT process in a computationally efficient manner, thereby making it suitable for real-time lane detection. This is based on the additive HT property that was previously described by us in [6] and [10]. The proposed method, called hierarchical additive Hough transform (HAHT), is applied for straight lane detection and it is shown to give good results, with significant savings in computation cost. The rest of the letter is organized as follows. In Section II-A, we propose an extended additive Hough transform property that will be used to devise the proposed hierarchical additive HAHT in Section II-B. The detection of straight lanes using the proposed HAHT is presented in Section III. In Section IV, a detailed discussion on computational complexity of the proposed HAHT and its implication on parallelism and timing is presented before concluding the letter in Section V.

II. Proposed HAHT

A. Extending the Additive Property to Multiple Levels

According to the additive property of HT stated in [10], the HT of point A with respect to (w.r.t.) the image origin "O" is equal to the sum of the HT of A w.r.t. any intermediate point B, and HT of B w.r.t. O, i.e.

$$HT(A,O) = HT(A,B) + HT(B,O) \qquad (2)$$

where $HT(X,Y)$ represents the HT of point X w.r.t. point Y. In other words, the HT of a point w.r.t. to a global origin can be broken into two parts: the HT of that point w.r.t. a local origin and the HT of the local origin w.r.t. the global origin.

We extend this property further by breaking the HT into multiple parts instead of two, as stated in the following corollary.

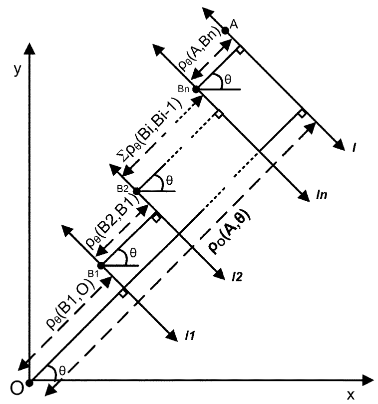

Fig. 1. Extending the additive property of HT across multiple levels.

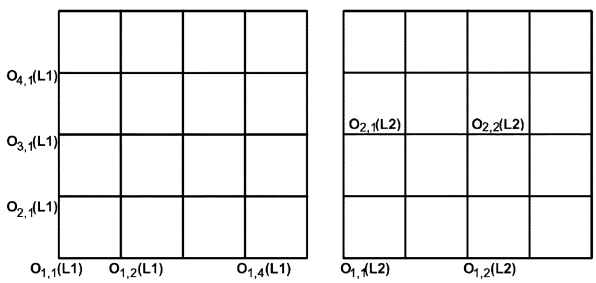

Fig. 2. Division of edge map into blocks and macroblocks and their origins.

Corollary of (2): Given a point A and a set of n points $\{B_1, B_2 \ldots B_n\}$ defined w.r.t. origin O, HT of A w.r.t. O is given by

$$HT(A, O) = HT(A, B_n) + \sum_{i=n}^{2} HT(B_i, B_{i-1}) + HT(B_1, O). \tag{3}$$

Proof: Consider a line l passing through point A as shown in Fig. 1. Let $l_1, l_2 \ldots l_n$ be lines passing through a set of points $\{B_1, B_2 \ldots B_n\}$, respectively, such that l is parallel to $l_1, l_2 \ldots l_n$, i.e. $l_1 || l_2 || \ldots || l_n || l$. Let $\rho_O(A, \theta)$ be length of the perpendicular from O to the line l passing through point A subtending angle θ with the x-axis. Similarly, we can define $\rho_\theta(B_1, O)$, $\rho_\theta(B_2, B_1)$, $\rho_\theta(B_i, B_{i-1})$, and $\rho_\theta(A, B_n)$ as indicated in Fig. 1. From Fig. 1, it can be seen that

$$\rho_\theta(A, O) = \rho_\theta(A, B_n) + \sum_{i=n}^{2} \rho_\theta(B_i, B_{i-1}) + \rho_\theta(B_1, O). \tag{4}$$

Each term in (4) can be mapped directly, on a one-to-one basis, to the corresponding HT terms in (3) resulting in the HT of A w.r.t. O for angle θ.

This extended additive Hough property for multiple levels is now applied to compute HT of edge pixels using the proposed HAHT.

B. Deriving the HAHT

Consider an edge map divided into k^2 blocks using a $k \times k$ grid. The steps of the HAHT can be explained using in the following manner, using the example of 4×4 grid as shown in Fig. 2. The lowest level of hierarchy is indicated by $L(1)$ wherein Hough space is computed for each of the k^2 edge map blocks. The ρ values are computed for edge pixels in every block w.r.t. to the origin of each block. The Hough space thus, generated is called local Hough space and the origin is called local origin. In Fig. 2(a), the local origins of the blocks at level $L(1)$ are given by $O_{1,1}(L1), O_{1,2}(L1) \ldots O_{4,4}(L1)$. These Hough spaces are given by $H(L1)$as

$$H(L1) = \left[\begin{array}{cc|cc} H_{4,1}(L1) & H_{4,2}(L1) & H_{4,3}(L1) & H_{4,4}(L1) \\ H_{3,1}(L1) & H_{3,2}(L1) & H_{3,3}(L1) & H_{3,4}(L1) \\ \hline H_{2,1}(L1) & H_{2,2}(L1) & H_{2,3}(L1) & H_{2,4}(L1) \\ H_{1,1}(L1) & H_{1,2}(L1) & H_{1,3}(L1) & H_{1,4}(L1) \end{array}\right]. \tag{5}$$

At each subsequent hierarchy level $L(i)$, where $i > 1$, the local Hough spaces of previous level $L(i-1)$ are grouped into groups of four. In each group, the constituent Hough spaces are used to generate a combined Hough space, wherein the ρ values are computed w.r.t. the local origin of the group of four image blocks. For example, in Fig. 2(b), four image blocks are now combined into one macroblock and the local origins for $L(2)$ are given by $O_{1,1}(L2) \ldots O_{2,2}(L2)$. This process of combining blocks is similarly extended to the next hierarchy level $L(3)$. The Hough spaces in $L(2)$ then become the local Hough spaces which will be combined in $L(3)$ to generate a combined Hough space w.r.t. a local origins in $L(3)$.

In order to compute the Hough space of a macroblock at $L(2)$, we apply the extended additive property of Hough transform. The HTs of four local origins of the constituent blocks are computed w.r.t. the origin of the macroblock. The same values can be used for all the macroblocks because the four local origins are placed at the same relative positions w.r.t. to the origins of all the macroblocks. Referring to one macroblock, say top right corner, the local Hough spaces, $H_{4,3}(L1)$, $H_{4,4}(L1)$, $H_{3,3}(L1)$, and $H_{3,4}(L1)$, of the constituent blocks from $L(1)$ are added with the HTs of their respective local origins w.r.t. to the macroblock origin as

$$\begin{aligned} H_{4,3}(L2) &= H_{4,3}(L1) + HT\left(O_{4,3}(L1), O_{2,2}(L2)\right) \\ &\vdots \\ H_{3,4}(L2) &= H_{3,4}(L1) + HT\left(O_{3,4}(L1), O_{2,2}(L2)\right). \end{aligned} \tag{6}$$

$H_{4,3}(L2)$, $H_{4,4}(L2)$, $H_{3,3}(L2)$, and $H_{3,4}(L2)$ are the Hough spaces of the four blocks w.r.t. the macroblock origin at level $L(2)$. The Hough space of the macroblock, i.e., $H_{2,2}(L2)$, w.r.t. its origin $O_{2,2}(L2)$ is the sum of these Hough spaces

$$H_{2,2}(L2) = H_{4,3}(L2) + H_{4,4}(L2) + H_{3,3}(L2) + H_{3,4}(L2). \tag{7}$$

Similarly, the Hough spaces of the rest of the three macroblocks are computed to give four Hough spaces in $L(2)$

$$H(L2) = \left[\begin{array}{c|c} H_{2,1}(L1) & H_{2,2}(L1) \\ \hline H_{1,1}(L1) & H_{1,2}(L1) \end{array}\right]. \tag{8}$$

Following the extended additive property of Hough transform, the four Hough spaces generated in $L(2)$ can be further combined to generate $L(3)$ Hough space. This Hough space would correspond to the macroblock generated by combining the four blocks from $L(2)$. The Hough transforms of the local origins in $L(2)$ are computed w.r.t. the origin of the macroblock in $L(3)$. Applying the extended additive Hough property, the

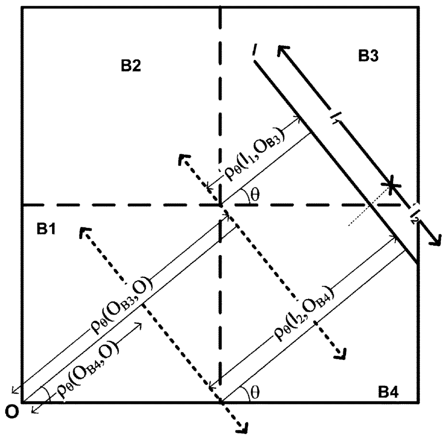

Fig. 3. Detection of long lines using HAHT.

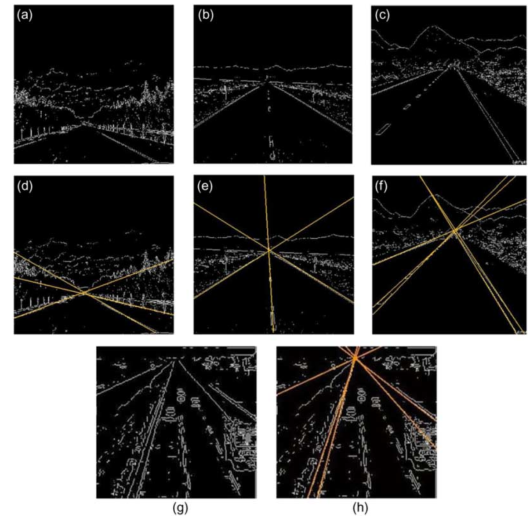

Fig. 4. Edge maps (a) Lane1, (b) Lane2, (c) Lane3, and (g) Lane4, and corresponding detected lanes in (d), (e), (f), and (h) using the proposed HAHT.

HTs of these origins are added to the Hough spaces from $L(2)$ hierarchy level and then combined to give $L(3)$ Hough space.

This hierarchical addition of Hough spaces using the extended additive property of HT can thus, be used to generate the Hough spaces at different levels of hierarchy.

III. HAHT for Detection of Straight Lanes

In this section, we demonstrate the use of the proposed HAHT on detection straight lanes. It is to be noted that the proposed HAHT can be used to detect curved lanes also using straight line approximation of curves. This is, however, not in the scope of this letter.

Lanes are usually characterized by straight lines. Depending on the block size and length of these lines, a line can be distributed across different blocks. The HAHT approach aids in accumulating the vote counts of the straight lines across different blocks. To illustrate this, we refer to Fig. 3. As shown in Fig. 3, consider a macroblock having four blocks, say $B1$, $B2$, $B3$, and $B4$. Let line l be divided as l_1 and l_2 in $B3$ and $B4$, respectively. At level $L(1)$, the local Hough spaces of $B3$ and $B4$ are given by $H_{B3}(L1)$ and $H_{B4}(L1)$, with peaks at two different ρ values, $\rho_\theta(l_1, O_{B3})$ and $\rho_\theta(l_2, O_{B4})$ corresponding to the edge pixels lying on l_1 and l_2. However, what we need is to locate the lanes w.r.t. to global origin, i.e., the origin of the entire edge map, O. At level $L(2)$, the blocks are grouped into one macroblock. The local Hough spaces of $B3$ and $B4$ are combined to compute the accumulation count of the entire line l that is made up of l_1 and l_2. The proposed HAHT method can now be used to generate the combined Hough space of line l w.r.t. the global origin O. This is done by adding the HT of the origins of $B3$ and $B4$ to the local Hough spaces of $B3$ and $B4$ that were generated in $L(1)$, and then adding the resulting Hough spaces into one Hough space for the macroblock. This follows from the extended Hough property, i.e.

$$\begin{aligned}\rho_\theta(l_1, O) &= \rho_\theta(l_1, O_{B3}) + \rho_\theta(O_{B3}, O)\\ \rho_\theta(l_2, O) &= \rho_\theta(l_1, O_{B4}) + \rho_\theta(O_{B4}, O)\end{aligned} \tag{9}$$

and the overall accumulation count for the line l is given by

$$n(\rho_l, \theta_l) = n\left(\rho_{l_1}, \theta_{l_1}\right) + n\left(\rho_{l_2}, \theta_{l_2}\right) \tag{10}$$

where $n(\rho_l, \theta_l)$ is the accumulation count for the line l. Depending on the block size, the additive property of HT can be applied repetitively to different hierarchy levels to generate the Hough space at the desired levels.

A. Further Optimization of HAHT for Lane Detection and Prevent False Line Detection

The proposed HAHT method of adding Hough spaces at different hierarchy levels can be further optimized to reduce the number of computations. This is done by thresholding the Hough spaces at every hierarchy level. If a threshold for Hough space at every hierarchy level is considered, the (ρ, θ) candidates that cross a threshold at a given hierarchy level $L(i)$ represent the line segments that cross the minimum length criteria in the macroblocks at $L(i)$. In the next hierarchy level $L(i+1)$, only the selected candidates from $L(i)$ are added to the HTs of their respective block origins. This computation process is continued across hierarchy levels till the accumulation counts cross an overall minimum length condition. This process also leads to significant savings because the redundancy of processing noisy edge pixels (that contribute to false lines) is controlled.

The proposed HAHT was applied on various lane images and the lanes were picked up successfully as shown in three of test cases in Fig. 4.

IV. Discussion on Hardware Related Issues

A. Computational Complexity

In all the hierarchical approaches considered in this letter [7]–[9], and because a filtering process to allow only prominent peaks through thresholding is done at each level, peak detection is also performed along with voting. This is unlike the conventional HT computation approaches where peak detection is performed after the entire voting is completed. The proposed HAHT leads to a drastic reduction in computational complexity of the voting step as compared to the existing hierarchical approaches. This is achieved because of two main reasons. First,

TABLE I
COMPARISON OF HT COMPUTATIONS

Image	Edge content (no. of pixels)	Proposed HAHT	Hierarchical HT in [9]	
		Adds	Adds	Multiplies
Lane1	8498	948	1530588	3061176
Lane2	4902	371	882731	1765462
Lane3	6137	488	1105148	2210296
Lane4	4063	11588	742928	1485856

as explained in Section I, existing hierarchical HT approaches [7]–[9] perform a recomputation of HT at every level. That is, for all the qualifying points upon thresholding at a lower level, the HT is recomputed w.r.t. the new origin of the hierarchy level. This operation is performed at every level of the hierarchy. The cost incurred for performing a HT in terms of computations is two multiplications and one addition, assuming the trigonometric values are stored in an LUT. However, in HAHT, because of the additive property, an incremental update on the vote is achieved at every level through just one addition (of the HT of block origins to selected candidates from previous hierarchy level), rather than a recomputation of the HT value at the current level.

Second, at the lowest level $L(1)$, even the additions need not be performed as the precomputed local HT's and the HT's of the origins of the blocks need to be accessed only from LUTs as shown in [6] and [10]. After accumulation, only those that exceed the threshold are subjected to addition in the next level. Table I shows the number of computations involved to compute the Hough spaces till level-2 for the different lane images shown in Fig. 3. It can be seen that the number of addition operations are drastically reduced in the proposed method as compared to [9]. Also, the proposed HAHT does not involve any multiplications as it uses the LUT-based method in [6] and [10]. This shows the number computations is lower by as high as 98%–99%. Similar savings were found in other test cases also. As far as the peak detection step is concerned, the complexity is governed by the comparisons that are done in the thresholding process. If we assume that the same thresholding scheme is employed for both the HAHT and the hierarchical method described in [9], then the peak detection processes would be similar in terms of complexity.

B. Parallelism and Timing

The main motivation for most hierarchical HT approaches has been to allow for parallelism, thereby accelerating the HT process. Hence, in terms of timing, these methods are usually far superior compared to the serial approaches in [11] and [12]. Further, due to the reduced computational complexity of the voting step in HAHT compared to other existing hierarchical HT methods, the HAHT leads to savings in time costs as well. This is because the timing cost at every hierarchy level involves only *one* addition per (ρ, θ) tuple in the proposed HAHT whereas a recomputation of HT in [7]–[9] involves a multiplication and addition. If a 16-bit multiplication and addition are employed, each HT computation for every (ρ, θ) tuple in [7]–[9] would require $5T_{16-\mathrm{bitAdd}}$, assuming $T_{16-\mathrm{bitMult}} = 4T_{16-\mathrm{bitAdd}}$ [13]. In the proposed HAHT, each HT computation would require only $T_{16-\mathrm{bitAdd}}$ time units.

V. CONCLUSION

In this letter, we have proposed the use of additive Hough property of HT across multiple levels of hierarchy to yield a novel computationally efficient HT architecture. We have explained how the use of the extended additive Hough property aids in not only replacing the complex operations with simpler additions, but also reduces the number of points processed at the lowest level of hierarchy. Hence, we have shown that the proposed HAHT leads to significant savings of over 99% in computations in the voting step, without any extra cost in peak detection. Further, we have also illustrated how the HAHT approach aids in accumulating the vote counts of the straight lines across different blocks. Finally, we have applied the HAHT on many test images to illustrate that the method successfully detects straight lanes in images. Future work includes detecting curved lanes using HAHT by approximating the curve as a set of line segments and studying the angle variations of these line segments.

REFERENCES

[1] P. Y. Hsiao, C. W. Yeh, S. S. Huang, and L. C. Fu, "A portable vision-based real-time lane departure warning system: Day and night," *IEEE Trans. Veh. Technol.*, vol. 58, no. 4, pp. 2089–2094, May 2009.

[2] J. P. Carrasco, A. de la Escalera, and J. M. Armingol, "Driving supervision through traffic sign analysis," in *Proc. IEEE Int. Conf. Veh. Elect. Safety*, Columbus, OH, 2008, pp. 42–247.

[3] V. Hardzeyeu and F. Klefenz, "On using the hough transform for driving assistance applications," in *Proc. 4th Int. Conf. Intell. Comput. Commun. Process.*, Shanghai, China, 2008, pp. 91–98.

[4] R. O. Duda and P. E. Hart, "Use of the Hough transform to detect lines and curves in pictures," *ACM Commun.*, vol. 15, no. 1, pp. 11–15, 1972.

[5] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2nd ed. New York: Prentice Hall, 2002.

[6] S. Suchitra, R. K. Satzoda, and T. Srikanthan, "Exploiting inherent parallelisms for accelerating linear Hough transform computation," *IEEE Trans. Image Process.*, vol. 18, no. 10, pp. 2255–2264, Oct. 2009.

[7] C. Espinosa and M. A. Perkowski, "Hierarchical Hough transform based on pyramidal architecture," in *Proc. 11th Annu. Int. Phoenix Conf. Comput. Comm.*, Scottsdale, AZ, 1992, pp. 743–750.

[8] J. Princen, J. Illingworth, and J. Kittler, "A hierarchical approach to line extraction," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog.*, San Diego, CA, 1989, pp. 92–97.

[9] S. B. Yacoub and J. M. Jolion, "Hierarchical line extraction," *IEE Proc. Vis. Image Signal Process.*, vol. 142, no. 1, pp. 7–14, 1995.

[10] R. K. Satzoda, S. Suchitra, and T. Srikanthan, "Parallelizing Hough transform computation," *IEEE Signal Process. Lett.*, vol. 15, pp. 297–300, 2008.

[11] M.-Y. Chern and Y.-H. Lu, "Design and integration of parallel Hough-transform chips for high-speed line detection," in *Proc. 11th Int. Conf. Para. Dist. Syst.*, Fukuoka, Japan, Jul. 2005, vol. 2, pp. 42–46.

[12] A. K. Majumdar, "Design of an ASIC for straight line detection in an image," in *Proc. 13th Int. Conf. VLSI Design*, Calcutta, India, Jan. 2000, pp. 128–133.

[13] K. K. Parhi, *VLSI Digital Processing Systems: Design and Implementation*. New York: Wiley, 1999.