

# 一种快速高效的Hough变换并行算法

陈<sup>1,2</sup>, 陈宏建<sup>1</sup>, 徐晓华<sup>1</sup>, 秦玲<sup>1</sup>

(1 扬州大学信息工程学院计算机系, 江苏扬州 225009;

2 南京大学软件新技术国家重点实验室, 江苏南京 210093)

**摘要:** 本文提出了一种基于流水光总线可重构线性阵列(LARPBS)模型快速高效Hough变换算法. 该算法利用LARPBS有力的通讯功能和计算能力, 对直线上的黑色像素进行抽取和统计. 对于大小为 $n \times n$ 的二值数字图像、 $m$ 个H参数值, 本算法可以使用 $mn^2$ 个处理机在 $O(1)$ 时间完成, 从而使速度和效率都达到最优. 文中还给出当LARPBS中处理机个数少于 $mn^2$ 时的算法分解执行方法, 说明算法具有很好的可扩展性.

**关键词:** LARPBS模型; Hough变换; 并行算法

**中图分类号:** TP37, O141.1.3 **文献标识码:** A **文章编号:** 0372-2112(2004)05-0752-04

## A Fast Efficient Parallel Algorithm for Hough Transform

CHEN Ling<sup>1,2</sup>, CHEN Hongjian<sup>1</sup>, XU Xiaohua<sup>1</sup>, QIN Ling<sup>1</sup>

(11 Department of Computer Science, Yangzhou University, Yangzhou, Jiangsu 225009, China;

21 National Key Lab of Novel Software Tech, Nanjing Univ. Nanjing, Jiangsu 210093, China)

**Abstract:** A parallel Hough transform algorithm on linear array with reconfigurable pipeline bus system (LARPBS) is presented. Utilizing the strong communication and computation ability of LARPBS, the algorithm can extract and count the foreground pixels on each line. Suppose the number of H values considered is  $m$ , the algorithm can complete Hough transform for an image with  $n \times n$  pixels in  $O(1)$  time using  $mn^2$  processors and get optimal speed and efficiency. We also illustrate how to partition and perform the algorithm on a LARPBS with less than  $mn^2$  processors, thus show the algorithm is highly scalable.

**Key words:** LARPBS model; Hough transform; parallel algorithm

### 1 引言

Hough变换<sup>[1-3]</sup>已被广泛地应用于计算机视觉、图像处理、模式识别等领域. 人们为了加速Hough变换计算, 提出了各种计算模型中的Hough变换的并行算法<sup>[4-15]</sup>. 在这些算法中, 文献[8, 13, 14]的效率最高, 对一幅大小为 $n \times n$ 的二值数字图像、 $m$ 个H参数值, 文献[8]使用 $m \times n^3$ 个处理机, 文献[13]使用 $m \times n^2$ 个处理机, 文献[14]使用 $m \times (\log n)^2 \times n^2$ 个处理机. 本文在LARPBS模型的基础上, 提出一种使用 $m \times n^2$ 个处理机 $O(1)$ 时间的Hough变换并行算法. 该算法与其他常数时间Hough变换并行算法相比, 具有更高的效率, 且速度和效率上都达到最优. 我们还给出了当LARPBS中处理机个数少于 $mn^2$ 时, 算法的分解执行方法, 从而说明它具有很好的可扩展性.

载体. 图1示出了LARPBS模型中的光总线与 $n$ 个处理机的连接结构. 在此结构中, 每个处理器与两个方向耦合器相连, 上方的耦合器用于发送信息, 下方的耦合器用于接受信息. 在LARPBS模型中有三条结构完全相同的光总线: 一条用于传送信息, 称为信息总线; 其余两条分别是用于传送地址信息的参考总线和选择总线, 如图1所示. 为简单起见, 图中只画出了参考总线和选择总线, 信息总线同参考总线类似, 故略去.

在选择总线发送端上的每两个处理机之间加上了一个调

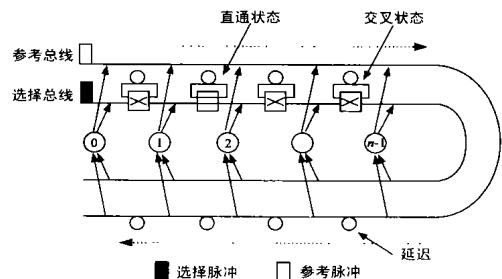


图1 LARPBS并行计算模型的结构

### 2 LARPBS模型的结构和基本数据操作

在基于流水光总线的可重构线性阵列系统LARPBS模型中, 以光波导代替原先的电子信号作为处理器间信息传递的

收稿日期: 2003-05-06; 修改日期: 2003-11-10

基金项目: 国家自然科学基金(No. 60074013); 国家高性能计算基金(No. 00219); 江苏省教育厅自然科学基金(No. 02KJB520009); 南京大学软件新技术国家重点实验室开放基金(No. 200110)

节开关. 这种调节开关具有直通和交叉两种状态, 如果某两个处理机之间的调节开关置于交叉状态, 则这两个处理机之间的距离比置于直通状态时长出一个单位长度 \$S\$. 这个调节开关可由相应的处理机根据算法的需要来设置状态. 接收端的耦合器使用脉冲吻合技术<sup>[16]</sup>来接收数据, 即只有当参考脉冲和选择脉冲同时到达时, 才接收信息总线上的数据. 为了实现这一技术, 需在参考总线的接受端的每两个处理机间加上一个额外的时间延迟, 即加上一个单位长度 \$S\$, 如图 1 所示. 在 LARPBS 模型的每一条总线的每个处理机的发送端和接收端的耦合器上分别装有选路开关, 它们有直通与交叉两种状态. 若某个处理机的两个开关为交叉状态, 而其余都置为直通, 则阵列在该处理器处分成两个子阵列. 选路开关的状态可由处理机根据算法来设定, 因此, 该模型具有很好的可重构性. 在 LARPBS 模型中, 由于使用了脉冲吻合寻址技术<sup>[16]</sup>, 一对一通讯、广播、多路广播、抽取压缩和求和等基本数据操作都能在常数时间里完成. 各个操作的具体实现原理, 可参见文献 [16].

### 3 快速高效的 Hough 变换并行算法

记二值数字图象的象素为  $a(i, j)$ ,  $0 \leq i, j \leq n-1$ , 其值为 1 表示为黑色象素, 0 表示为白色象素. 我们将 H 角度分为四类, 分别在区间  $[0, P/4)$ ,  $[P/4, P/2)$ ,  $[P/2, 3P/4)$ ,  $[3P/4, P]$  之中. 下面仅讨论  $H \in [0, P/4)$  的情况, 对于其他范围内的 H 值, 可以用类似的方法在相同的时间解决. 设在区间  $[0, P/4)$  内有  $m/4$  个 H 值, 记  $H_k = k/m\#P$ ,  $0 \leq k < m/4$ . 对于在区间  $[0, P/4)$  内固定的 H 角, 设象素  $a(i, j)$  的 Q 值为  $Q_{(i,j)}$ , 则根据图象的几何性质可知下列引理成立.

**引理 1** 对区间  $[0, P/4)$  内固定的 H 值和任意整数  $i, j$ :  $0 \leq i \leq n-1, 0 \leq j \leq n-3$ , 若满足  $Q_{(i,j)} = Q_{(i,j+2)}$ , 则有  $Q_{(i,j)} = Q_{(i,j+1)} = Q_{(i,j+2)}$ .

**引理 2** 对区间  $[0, P/4)$  内固定的 Q、H 值和任意整数  $i, h, j$ :  $0 \leq i, h \leq n-1, 0 \leq j \leq n-2$ , 若象素  $a(i, j)$  和  $a(h, j+1)$  都在直线  $(H, Q)$  上, 则在  $a(i, j)$  和  $a(h, j+1)$  之间没有其他象素位于直线  $(H, Q)$  上.

这些引理说明了: 在同一行上具有相同 Q 值的象素必然是连续相邻的. 同时, 对区间  $[0, P/4)$  内固定的 H 值和 Q 值, 可以通过对图象的逐行探测, 来逐个找出直线  $(H, Q)$  上的象素而不会遗漏. 其他范围内的 H 值也有类似的性质.

如图 2 所示, 对于固定的  $H_k$ , 它所对应的是一组平行的、与水平方向成  $H_k + P/2$  角的直线, 它们的斜率为  $h = \text{tg}(H_k + P/2) = -\text{ctg}H_k$ . 设  $(M_1, 0)$  为过点  $(0, n-1)$ 、斜率为  $h$  的直线与  $x$  轴的交点;  $(M_2, 0)$  为过点  $(n-1, n-1)$ 、斜率为  $h$  的直线与  $x$  轴的交点, 则  $M_1 = 7 \frac{1}{h} (n-1) \delta, M_2 = 7(1 - \frac{1}{h}) (n-1) \delta$ . 称象素  $(0, 0), (1, 0), \dots, (M_1, 0), \dots, (n-1, 0)$  为相应直线上的首元素. 设经过首元素  $(i, 0)$ 、斜率为  $h$  的直线在图象范围内的象素的最低纵坐标为  $l_i$ , 最高纵坐标为  $u_i$ , 则该直线位于图象范围内的所有象素个数  $g_i = u_i - l_i + 1$ .  $l_i, u_i$  的值可由下式计算:

$$\begin{cases} l_i = 0, u_i = \max\{kI - Z | (i + \frac{k}{h}) \setminus 0\}, & \text{当 } 0 \leq i \leq M_1 \text{ 时} \\ l_i = 0, u_i = n - 1, & \text{当 } M_1 < i \leq n - 1 \text{ 时} \\ l_i = \min\{kI - Z | (i + \frac{k}{h}) \setminus [n - 1]\}, u_i = n - 1, & \text{当 } n - 1 < i \leq M_2 \text{ 时} \end{cases} \quad (1)$$

这里,  $Z$  表示整数的集合. 设在 LARPBS 模型中有  $m\#n^2$  个处理机, 记第  $r$  个处理机为  $p_r$ . 设  $r = kn^2 + in + j$ , 则  $p_r$  又可表示为  $p(k, i, j)$ . 设  $H \in [0, P/4)$ , LARPBS 模型中 Hough 变换的并行算法可作如下描述, 对于其他范围内的 H 值, 算法是类似的.

Algorithm PARAHOUGH ( $n, m$ )

```

/* 算法开始时, a(i, j) 存于 P(0, i, j) 的存储器 A(0, i, j) 之中 */
for 0 [ k [ m/4 - 1 pa2do
  { step 1 for 0 [ i, j [ n - 1 pa2do A(k, i, j) z A(0, i, j);
  step 2 /* 在子阵列 p(k, *, *) 中实现如下的步骤 */
  step 2.1 for 0 [ i, j [ n - 1 pa2do H(k, i, j) = k/m#P/4;
  step 2.2 for 0 [ i [ M2 pa2do
    /* 在子阵列 p(k, *, *) 的第 i 个处理机 p_{kn^2+i} 中计算 */
    {对首元素 (i, 0), 根据其所在位置由式 (1) 计算出 l_i, u_i, g_i 和相应的 Q 值 (记为 Q)}
  step 2.3 /* 在处理机 p_{kn^2+1}, p_{kn^2+2}, \dots, p_{kn^2+M_2} 中进行前缀和计算得到首元素移动后所位于的处理机的标号 S_i */
    for 1 [ i [ M2 pa2do S_i z g_{i+1} + g_{i+2} + \dots + g_i
  step 2.4 /* 将 Q、i、l_i、S_i 广播到象素 a(i + 7j/h\delta, j) 所在的处理机中 */
    for 1 [ i [ M2 pa2do
      for l_i [ j [ u_i pa2do
        {Q(k, i + 7j/h\delta, j) z Q; l(k, i + 7j/h\delta, j) z l_i; S(k, i + 7j/h\delta, j) z S_i}
  step 2.5 /* 将所有 A 值及其 Q 值送到目标处理机 */
    for 0 [ i [ n - 1 pa2do
      for 0 [ j [ n - 1 pa2do
        { A(k, i, j) y A[kn^2 + S(k, i + 7j/h\delta, j) + j - 1(k, i + 7j/h\delta, j) + 1];
          Q(k, i, j) y Q[kn^2 + S(k, i + 7j/h\delta, j) + j - 1(k, i + 7j/h\delta, j) + 1]}
  step 2.6 /* 让每个处理机与其左边相邻的处理机比较各自的 Q 值, 找出划分点, 以构成更小的子阵列, 使每个子阵列中的 Q 值相等 */

```

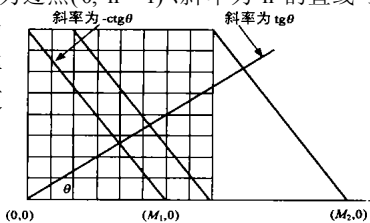


图 2 斜率为  $-\text{ctg}H$  的一组直线

```

for 1[ i[ n2- 1 pa2do
  {Q( kn2+ i) y R( kn2+ i- 1);
   if Q( kn2+ i)= R( kn2+ i) then S( kn2+ i)= 1
   else S( kn2+ i)= 0 }
step 2.7 将子阵列 p(k, *, *) 在 S 值为 0 的处理机
处重构成若干个更小的子阵列. 设第 i 个子阵列
中的 Q 值为 Qi, 对其中所有处理机的 A 值相加, 即
得直线( H, Q) 对应的计数器值 count( H, Q). }
}

```

上述算法的 step1、214 是广播操作, step1 211、212 都是处理机的独立计算, step213 是求前缀和操作, step2 5、216 是点到点传送, step217 为二进制求和, 上述各步均可在 LARPBS 中在 O(1) 时间完成. 因此整个算法可以使用  $m\#n^2$  个处理机在常数时间完成, 使该算法的时间达到了最优. 由于该算法的成本为  $O(mn^2)$ , 其效率也达到了最优. 据我们所知, 这是目前速度最快、效率最高的 Hough 变换并行算法.

当处理机的个数  $p$  少于  $mn^2$  时, 对于本算法可通过分解的方法来运行, 且时间复杂度为  $O(mn^2/p)$ , 因此它具有较好的可扩展性.

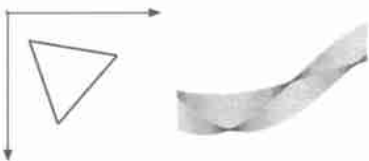
(1) 当  $n^2 \lceil p \lceil mn^2$  时, 设  $p = r\#n^2$ , 则  $1 \lceil r \lceil m$ . 设  $q = 7m/r\hat{o}$ , 我们可以将  $m$  个 H 值分为  $q$  组, 每一组中含有  $r$  个 H 值, 在  $p = r\#n^2$  个处理机构成的 LARPBS 中可以在  $O(1)$  时间里对一组中的  $r$  个 H 值进行处理. 整个算法依次对  $q$  组 H 值进行这样的处理, 完成时间为  $O(q) = O(mn^2/p)$ . 这说明了, 当  $n^2 \lceil p \lceil mn^2$  时算法具有很好的可扩展性.

(2) 当  $1 \lceil p < n^2$  时, 设  $q = \lfloor n^2/p \rfloor$ , 由于处理机数小于图象的像素个数, 我们对 H 值逐个进行处理. 在处理一个 H 值时, 将算法 PARA2HOUGH(  $n, m$ ) 中的 step1 对象素的分布方式作如下改进: 将  $n^2$  个像素分为  $q$  组, 每组中有  $p$  个像素, 它们被分布在各个处理机上. 具体地, 像素  $a(i, j)$  被分配在第  $\lfloor (i-1)n + j/p \rfloor$  个处理机, 作为该处理机的第  $7(i-1)n + j/p\hat{o}$  个像素. 类似地, 对算法中 step 2 的每一步, 我们都可以将其所使用的处理机分解若干组, 使每一组中的处理机的个数为  $p$ , 则 step 2 中的每一步都可以按组依次进行处理. 由于有  $n^2/p$  组, 则时间复杂度为  $O(n^2/p)$ . 对  $m$  个 H 值依次进行处理, 则完成时间为  $O(mn^2/p)$ .

综合 (1)、(2) 可见: 当  $1 \lceil p \lceil mn^2$  时, 算法可以通过分解执行的方法用  $p$  个处理机在  $O(mn^2/p)$  时间内完成, 这说明了本算法具有很好的可扩展性. 由于此时算法的成本为  $O(p \cdot mn^2/p) = O(mn^2)$ , 在成本和效率上仍然为最优.

### 4 实验结果

为了验证算法的正确性, 我们对图 3 和图 5 两幅  $128 \times 128$  的图象在用 16 个 1.6GHz P4 由以太网连成的机群上进行模拟实验. 图 4 和图 6 分别是图 3 和图 5 变换后的 Hough 图象. 图 3 的三条直线 图 3 图象 1 图 4 图象 1 的 Hough 图象



在 Hough 空间中产生如图 5 所示的三个汇聚点. 图 4 中的六条直线在 Hough 空间中产生如图 6 所示的六个汇聚点.

表 1 示出使用不同个数的处理机时所用的计算时间, 由表中可以看出, 计算时间近似地随处理机个数的增加而成比例地减少, 说明了本算法具有很好的可扩展性.

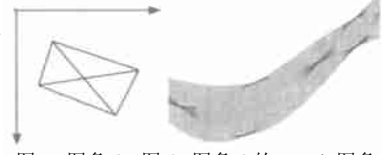


表 1 处理机个数和处理时间的关系

处理机个数	1	2	8	16
图 3 处理时间(s)	0.351750	0.114613	0.041947	0.030361
图 5 处理时间(s)	0.344224	0.179169	0.069856	0.048755

### 5 结论

本文提出了基于流水光总线可重构线性阵列系统 LARPBS 上的一种快速高效的 Hough 变换并行算法, 对于大小为  $n \times n$  的二值数字图象、 $m$  个 H 参数值, 该算法使用  $mn^2$  个处理机在常数时间完成, 从而使算法在速度和效率上都达到最优, 这是以往任何一种并行计算模型所无法比拟的. 我们还给出了当 LARPBS 中处理机个数少于  $mn^2$  时, 算法的分解执行方法, 说明该算法具有很好的可扩展性. 这也说明 LARPBS 是一种高效的新型并行计算模型, 它将在很多领域得到广泛的应用.

### 参考文献:

- [ 1 ] Xu L, OJA E. Randomized Hough transform(RHT): Basic mechanism, algorithms, and computational complexities[ J]. CVGIP: Image Understanding, 1993, 57(2): 131- 154.
- [ 2 ] Jiang T Z, MA S D. Geometric primitive extraction using tabu search [A]. Kropatsch W G. Proceedings of the 13th International Conference on Pattern Recognition [C]. Los Alamitos: IEEE Computer Society Press, 1996. 266- 279.
- [ 3 ] Roth G, Levin M D. Geometric primitive extraction using a genetic algorithm[ J]. IEEE Trans on PAMI, 1994, 16(9): 901- 905.
- [ 4 ] Ibrahim H A H, Kender J R, Shaw D E. The analysis and performance of two middle-level vision tasks on a fine-grained SIMD tree machine [A]. Binford T O. Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition [C]. Los Alamitos: IEEE Computer Society Press, 1985. 387- 393.
- [ 5 ] Fisher A L, Highnam P T. Computing the hough transform on a scan line array processor[ J]. IEEE Transactions on PAMI, 1989, 11(3): 262 - 265.
- [ 6 ] Jolion J, Rosenfeld A. An  $O(\log n)$  pyramid Hough transform[ J]. Pattern Recognition Letters, 1989, 9(5): 343- 349.
- [ 7 ] Pan Y, Chung H Y H. Faster line detection algorithms on enhanced mesh connected arrays[ J]. IEEE Proceeding2E, 1993, 2( 140): 95- 100.
- [ 8 ] Chung K L, Lin H Y. Hough transform on reconfigurable meshes[ J]. Computer Vision and Image Understanding, 1995, 61(2): 278- 284.
- [ 9 ] Kao T W, Hong S J, Wang Y L. An  $O(1)$  time algorithms for compu2

- ing histogram and the Hough transform on a cross2bridge reconfigurable array of processors[ J]. IEEE Transactions on Systems, Man and Cybernetics, 1995, 25(4): 681- 687.
- [ 10] Lin S S. Constant2time Hough transform on the processor array with reconfigurable bus systems[ J]. computing, 1994, 52: 1- 15.
- [ 11] Merry M, Baker J W. Constant time algorithm for computing Hough transform on a reconfigurable mesh[ J]. Image and Vision Computing, 1996, 14: 35- 37.
- [ 12] Pan Y. A more efficient constant time algorithm for computing the hough transform[ J]. Parallel Processing Letters, 1994, 4(1/2): 45- 52.
- [ 13] Pan Y, Li K, Hamdi M. An improved constant time algorithm for computing the Radon and Hough transforms on a reconfigurable mesh[ J]. IEEE Transactions on Systems, Man, and Cybernetics, (Part A), 1999, 29(4): 417- 421.
- [ 14] Pan Y. Constant2time Hough transform on a 3D reconfigurable mesh using fewer processors[ A]. Rolim J, et al. Proceedings of 2000 Workshops on Parallel and Distributed Processing LNCS 1800[ C]. Heidelberg: Springer2Verlag, 2000. 966- 973.
- [ 15] Pavel S, Akl S G. Computing the Hough transformation on arrays with reconfigurable optical buses[ A], Li K, Pan Y, Zheng S Q. Parallel Computing Using Optical Interconnections[ C]. Boston: Kluwer Academic Publishers, 1998. 205- 226.

- [ 16] Pan Y. Basic data movement operations on the LARPBS model[ A]. Li K, Pan Y, Zheng S Q. Parallel Computing Using Optical Interconnections[ C]. Boston: Kluwer Academic Publishers, 1998. 227- 247.

#### 作者简介:



陈 男, 1951 年生于江苏省宝应县, 教授, 主要研究方向为算法设计和并行计算。  
Email: lchen@yzcn.net.



陈宏建 男, 1968 年生于江苏省海安县, 讲师, 主要研究方向为系统优化和并行计算。  
Email: yzchj@yzcn.net.