

# 基于 LS MPP 的图像分块模板匹配并行算法<sup>1</sup>

李俊山\*\*\* 沈绪榜\*

\*(西安微电子技术研究所 西安 710054)

\*\* (第二炮兵工程学院计算机系 西安 710025)

**摘 要** 该文提出了一种新颖的在基于  $K$  元 2-立方体网络的 SIMD 计算机上运行的图像分块模板匹配并行算法。该算法对于  $N \times N$  的图像、 $M \times M$  的模板和  $K \times K$  的处理元阵列 ( $M < K, N = BK, B > 1$ )，可通过将图像分成  $B \times B$  个图像块的分块模板匹配来实现。与已知的各算法相比，该算法具有可处理比处理元阵列尺寸大的图像的优越性。

**关键词** 图像分块，模板匹配，SIMD 计算机，并行算法

**中图分类号** TN911.73

## 1 引 言

模板匹配是图像处理和计算机视觉中的基本操作，是进行滤波、边缘检测、图像匹配和目标识别的有效方法。由于图像匹配和计算机视觉问题一样都是计算密集型的，所以各种串行算法在应用中都遇到了难于实时计算的困难。随着并行图像处理技术的发展，人们围绕模板匹配这一图像处理中的基本操作已提出了一些适用于不同互连网络结构的 SIMD(及 MIMD) 计算机的模板匹配并行算法<sup>[1-4]</sup>。然而，迄今为止所报道的图像模板匹配并行算法均假设图像与 SIMD 计算机的处理元 (PE) 阵列 (或 MIMD 计算机的处理机数目) 具有相同的大小。但事实上由于集成电路工艺水平等众多因素的影响，图像的尺寸可能比 PE 阵列的大小大若干倍。本文通过模板匹配算法在大规模并行处理计算机 (Large Scale Massively Parallel Processors—LS MPP) 中的应用，研究在图像分块的情况下实现模板匹配算法的并行处理方法。

LS MPP  $K$  元 2-立方体网络 SIMD 计算机由 PE 阵列、阵列存储器和带有程序存储器的控制器组成，是一台称为协处理器的计算机<sup>[5]</sup>。PE 阵列由  $32 \times 32$  个 PE 组成，每个 PE 包含有算术逻辑部件 ALU，寄存器组 (用作局部存储器，包括有  $r_0, r_1, \dots, r_7$ )，缓冲器和数据通信用的路由器。各个 PE 利用一个 32 元 2-立方体网络相互连接<sup>[6]</sup>。PE 之间有两种通信机制：一是邻近双向通信，即对于任何一个 PE 可进行  $E \rightarrow W, W \rightarrow E, S \rightarrow N, N \rightarrow S$  的数据传送；二是播送通信，即允许将数据同时按行或按列方式播送到所有处理元中。

## 2 图像分块模板匹配的原理

设  $F = \{F_{u,v}\}$  表示  $N \times N$  的图像， $u, v \in [0, N-1]$ ； $W = \{W_{st}\}$  表示  $M \times M$  的模板矩阵， $s, t \in [0, M-1]$ ； $\{PE_{i,j}\}$  表示  $K \times K$  的 PE 阵列， $i, j \in [0, K-1]$ ，且  $N = BK, M < K, 1 < B$ 。显然，图像的大小是 PE 阵列大小的  $B \times B$  倍，因而必须先将图像分成  $B \times B$  个与 PE 阵列大小相同的图像块  $F_b(q,r)(q, r \in [0, B-1])$ ，然后在与图像块大小相同的 PE 阵列上实现模板匹配。在 LS MPP 中可处理的块的大小为  $32 \times 32$  的像素阵列。图像分块模板匹配的结

<sup>1</sup> 2000-06-20 收到，2000-12-07 定稿  
国家部级项目资助

果  $C = \{C_{u,v}\}$  可描述为

$$C_{uv} = \sum_{s=0}^{M-1} \sum_{t=0}^{M-1} F_{(qK+i+s) \bmod N, (rK+j+t) \bmod N} \times W_{st} \quad (1)$$

其中  $u = qK + i, v = rK + j; 0 \leq q, r \leq B - 1; 0 \leq i, j \leq K - 1$ 。

当  $M = 3$  时将 (1) 式展开 (为简单起见, 展开式省去了  $F_{i,j}$  下标中的求模表示 “mod  $N$ ”):

$$\begin{aligned} C_{qK+i, rK+j} &= \sum_{s=0}^{M-1} \sum_{t=0}^{M-1} F_{(qK+i+s) \bmod N, (rK+j+t) \bmod N} \times W_{st} \\ &= F_{qK+i, rK+j} W_{00} + F_{qK+i, rK+j+1} W_{01} + F_{qK+i, rK+j+2} W_{02} \\ &\quad + F_{qK+i+1, rK+j} W_{10} + F_{qK+i+1, rK+j+1} W_{11} + F_{qK+i+1, rK+j+2} W_{12} \\ &\quad + F_{qK+i+2, rK+j} W_{20} + F_{qK+i+2, rK+j+1} W_{21} + F_{qK+i+2, rK+j+2} W_{22} \end{aligned} \quad (2)$$

由 (2) 式可知: 对于任意的  $M < K$  和  $PE(i, j) (i, j \in [0, K - 1])$ , 在  $PE(i, j)$  上完成模板匹配运算需要将图像像素数据向左平行移动  $M - 1$  次, 将图像像素数据向上平行移动  $M - 1$  次。但图像分块后按块分别进行模板匹配操作的事实, 导致了部分向上和向左平移的图像像素数据不在其下邻和右邻的  $PE(i, j)$  中。例如, 若  $N = 160, K = 32, M = 3$ , 并假设正在处理的图像块为  $Fb(0, 0)$ , 则当  $j = 31$  时, 展开 (2) 式应为

$$\begin{aligned} C_{i, 31} &= \sum_{s=0}^2 \sum_{t=0}^2 F_{(i+s) \bmod 32, (31+t) \bmod 32} \times W_{st} \\ &= F_{i, 31} W_{00} + F_{i, 32} W_{01} + F_{i, 33} W_{02} + F_{i+1, 31} W_{10} + F_{i+1, 32} W_{11} + F_{i+1, 33} W_{12} \\ &\quad + F_{i+2, 31} W_{20} + F_{i+2, 32} W_{21} + F_{i+2, 33} W_{22} \end{aligned} \quad (3)$$

由 (3) 式可知, 图像像素数据  $F_{i, 32}, F_{i, 33}, F_{i+1, 32}, F_{i+1, 33}, F_{i+2, 32}, F_{i+2, 33}$  不是当前处理的图像块  $Fb(0, 0)$  中的像素数据, 而是 (通过向左平移) 来自右邻块  $Fb(0, 1)$  和右下邻块  $Fb(1, 1)$ , 在模板匹配的并行操作运算中是来源于阵列存储器。同理可推知当  $i$  为 30 和 31 时, 有部分图像像素数据是来自下邻块  $Fb(1, 0)$ 。因此图像分块模板匹配并行算法中的向上平行移动和向左平行移动操作的定义同时涉及到, 要按各图像块匹配的先后次序, 分别为各图像块组织该块在与模板进行模板匹配时, 向上平移移入该块最下面的一行和向左平移移入该块最右面的一列的图像像素阵列的问题。

### 3 数据平行移动操作及其上移和左移图像像素数据阵列的组织

#### 3.1 向上平行移动操作及其图像像素数据的组织

**定理 1** 图像块  $Fb(q, r)$  的上移像素数据阵列的大小为  $(M - 1) \times K$ , 上移像素数据阵列的数据元素是图像块  $Fb(q + 1, r)$  的第 0 行至第  $M - 2$  行的像素数据。且其一般表示形式如 (4) 式所示 (为简单起见, 式中省去了  $F_{i,j}$  下标中的求模表示 “mod  $N$ ”)。

$$\begin{bmatrix} F_{(q+1)K, rK} & F_{(q+1)K, (rK+1)} & \cdots & F_{(q+1), (rK+K-1)} \\ F_{((q+1)K+1), rK} & F_{((q+1)K+1), (rK+1)} & \cdots & F_{((q+1)K+1), (rK+K-1)} \\ \vdots & \vdots & \vdots & \vdots \\ F_{((q+1)K+M-1), rK} & F_{((q+1)K+M-1), (rK+1)} & \cdots & F_{((q+1)K+M-1), (rK+K-1)} \end{bmatrix} \quad (4)$$

对定理 1 的证明, 可一般地在  $N \times N$  的图像,  $M \times M$  的模板,  $K \times K$  的 PE 阵列, 且  $M < K, N = BK, 2 < B, 0 \leq q \leq B - 2$  的假设下, 通过对  $M$  的归纳得到. 由于篇幅所限, 证明过程从略. 实际上, 向上平移像素数据阵列中有用的数据只有前  $M - 1$  行 (如定理 1 所述), 第  $M$  行数据只是为了算法的规整性而设置的 (参见算法 3 的语句段 (2.3)).

设有图像块  $Fb = \{Fb_{qr}\}, q, r \in [0, B - 1]$ ; 模板  $W = \{W_{st}\}, s, t \in [0, M - 1]$ ; 且  $rs$  和  $rd$  是各处理元 PE 中的寄存器编号,  $R(i, j, rx)$  表示 PE( $i, j$ ) 中的寄存器  $rx$ . 则向上平行移动操作的实现如算法 1 所示. 在本文的各个算法描述中, 符号 “ $\leftarrow$ ” 指变量赋值, 同一 PE 内的各寄存器之间的数据传送, 以及各 PE 间的寄存器之间的数据传送. 符号 “ $\Leftarrow$ ” 指将数据存储器中的数据加载到 PE 单元的寄存器中, 或将 PE 单元的寄存器中的数据传送到数据存储器中. “pardo on all PE( $i, j$ ) (约束条件)  $\cdots$  do” 和 “end pardo” 是一对并行处理标识括号, 意指对于处理元阵列中所有满足约束条件的 PE 同时 (并行) 执行该括号内的操作.

**算法 1**

```

procedure ROTATE-UP( $q, r, s, rs, rd$ )
begin
  pardo on all PE( $i, j$ ) ( $0 \leq i \leq K - 2, 0 \leq j \leq K - 1$ ) do
     $R(i, j, rd) \leftarrow R(i + 1, j, rs)$ ;
  end pardo;
  pardo on all PE( $i, j$ ) ( $i = K - 1, 0 \leq j \leq K - 1$ ) do
     $R(i, j, rd) \Leftarrow Fb(q, r)$  的向上平移图像像素数组的第  $s$  行的像素  $F_{s, j}$ ;
  end pardo;
end.
    
```

**3.2 向左平行移动操作及其图像像素数据的组织**

向左平行移动操作是和向上平行移动操作相配合来支持模板匹配的并行操作的. 由 (3) 式及其分析可知, 模板匹配并行算法的向左平移阵列的像素数据是来自当前处理的图像块  $Fb(q, r)$  的右邻块  $Fb(q, r + 1)$  和右下邻块  $Fb(q + 1, r + 1)$ , 这些像素数据组成了一个  $(K + M - 1) \times (M - 1)$  的像素数据阵列  $FA$ , 如 (5) 式所示 (为简单起见, 阵列中省去了  $F_{i, j}$  下标中的求模表示 “mod $N$ ”).

$$FA = \begin{bmatrix}
 F_{qK, (r+1)K} & F_{qK, ((r+1)K+1)} & \cdots & F_{qK, ((r+1)K+M-2)} \\
 F_{(qK+1), (r+1)K} & F_{(qK+1), ((r+1)K+1)} & \cdots & F_{(qK+1), ((r+1)K+M-2)} \\
 \vdots & \vdots & & \vdots \\
 F_{(qK+K-1), (r+1)K} & F_{(qK+K-1), ((r+1)K+1)} & \cdots & F_{(qK+K-1), ((r+1)K+M-2)} \\
 F_{(qK+K), (r+1)K} & F_{(qK+K), ((r+1)K+1)} & \cdots & F_{(qK+K), ((r+1)K+M-2)} \\
 \vdots & \vdots & & \vdots \\
 F_{(qK+K+M-2), (r+1)K} & F_{(qK+K+M-2), ((r+1)K+1)} & \cdots & F_{(qK+K+M-2), ((r+1)K+M-2)}
 \end{bmatrix} \tag{5}$$

为了便于描述向左平移操作的图像像素数据的组织, 首先给出有关的矩阵运算的定义.

**定义 1** 设  $A$  是  $M1 \times N1$  的矩阵,  $i$  和  $j$  是正整数, 且  $j - i + 1 < M1$ , 则由  $A$  的第  $i$  行向量至第  $j$  行的行向量组成的  $(j - i + 1) \times N1$  矩阵称为  $A$  的真子矩阵, 并记为  $A^{(i, j)}$ .

**定义 2** 设矩阵  $B_i (i = 1, 2, \cdots, n)$  是  $M2 \times N2$  的矩阵, 则由  $n$  个矩阵  $B_i$  组成的块组合矩阵  $B$  是一个  $M2 \times (n \times N2)$  的矩阵, 并表示为

$$B = (B_1 \ B_2 \ \cdots \ B_n) \tag{6}$$

**定理 2** 设有如 (5) 式所示的  $(K+M-1) \times (M-1)$  的像素数据阵列  $\mathbf{FA}$ , 图像块  $F_b(q, r)$  的左移像素数据阵列是一个由  $\mathbf{FA}$  中的行向量组成的大小为  $(K+M-1) \times M(M-1)$  的矩阵, 其一般表示形式为

$$\begin{aligned} & (\mathbf{FA}^{qK \bmod N, (qK+K-1) \bmod N} \quad \mathbf{FA}^{(qK+1) \bmod N, (qK+K) \bmod N} \quad \dots \\ & \mathbf{FA}^{(qK+M-2) \bmod N, (qK+K+M-3) \bmod N} \quad \mathbf{FA}^{(qK+M-1) \bmod N, (qK+K+M-2) \bmod N}) \end{aligned} \quad (7)$$

设有和算法 1 相同的条件和符号约定, 则向左平行移动操作的实现如算法 2 所示。

#### 算法 2

```

procedure ROTATE-LEFT( $q, r, t, rs, rd$ )
begin
  pardo on all PE( $i, j$ )( $0 \leq i \leq K-1, 0 \leq j \leq K-2$ ) do
     $R(i, j, rd) \leftarrow R(i, j+1, rs)$ ;
  end pardo;
  pardo on all PE( $i, j$ )( $0 \leq i \leq K-1, j = K-1$ ) do
     $R(i, j, rd) \leftarrow F_b(q, r)$  的向左平移图像像素数组的第  $r$  列的像素  $F_{i,r}$ ;
  end pardo;
end.

```

### 4 基于 $K$ 元 2-立方体网络 SIMD 计算机的图像分块模板匹配并行算法

设对于所有的  $PE(i, j)$ ,  $0 \leq i, j \leq K-1$ , 用  $PIXEL0$  表示图像像素  $F_{ij}$ , 并用  $FIJ$  暂存用于平行上移时用的图像像素  $F_{ij}$ ; 用  $WCOL0$  表示模板元素  $W_{st}$ ; 用  $PMUL$  表示乘积  $F_{ij} \times W_{st}$ ; 用  $CSUM$  表示累加和并最终结果。且图像、模板及 PE 阵列之间的尺寸关系如前所述, 并设阵列存储器中的模板矩阵  $\mathbf{W}$  的元素按以行为主序号排列, 则基于 LS MPP 的  $K$  元 2-立方体网络的 SIMD 计算机的图像分块模板匹配并行算法如算法 3 所示。

#### 算法 3

```

procedure TEMPLATE-DivBlock
begin
  for  $q \leftarrow 0$  to  $B-1$  do
    for  $r \leftarrow 0$  to  $B-1$  do
      (1) pardo on all PE( $i, j$ )( $u = qK + i, v = rK + j, 0 \leq i, j \leq K-1$ ) do
         $PIXEL0 \leftarrow$  阵列存储器中图像  $F$  的图像块  $F_b(q, r)$  的像素  $F_{uv}$ ; /* 从存储器
          加载图象块 */
         $FIJ \leftarrow PIXEL0$ ; /* 暂存  $PIXEL0$  */
         $CSUM \leftarrow 0$ ; /* 给累加和变量  $CSUM$  赋初值 */
      end pardo;
      (2) pardo on all PE( $i, j$ )( $u = qK + i, v = rK + j, 0 \leq i, j \leq K-1$ ) do /* 计算部分和
        与匹配结果  $CSUM$  */
        (2.0) for  $s \leftarrow 0$  to  $M-1$  do
          (2.1)  $WCOL0 \leftarrow W(s, 0)$ ; /* 模板的  $s$  行 0 列元素加载到 PE( $i, j$ ) 的  $WCOL0$ ,
             $0 \leq s \leq M-1$  */
             $PMUL \leftarrow PIXEL0 \times WCOL0$ ;
             $CSUM \leftarrow CSUM + PMUL$ ;
          end do
        end do
      end do
    end do
  end do
end.

```

```

(2.2)   for  $t \leftarrow 1$  to  $M - 1$  do
          /* PE( $i, j + 1$ ) 中的 PIXEL0 向左平移到 PE( $i, j$ ) 的 PIXEL0*/
          ROTATE-LEFT( $q, r, t, \text{PIXEL0}, \text{PIXEL0}$ );
          WCOL0  $\leftarrow W(s, t)$ ; /* 模板  $s$  行  $t$  列元素加载到 PE( $i, j$ ) 的 WCOL0,
          1  $\leq t \leq M - 1$ */
          PMUL  $\leftarrow \text{PIXEL0} \times \text{WCOL0}$ ;
          CSUM  $\leftarrow \text{CSUM} + \text{PSUM}$ ;
        end for- $t$ ;
(2.3)   ROTATE-UP( $q, r, s, \text{FIJ}, \text{PIXEL0}$ ); /* FIJ 暂存的像素向上平移后再暂存 */
          FIJ  $\leftarrow \text{PIXEL0}$ ;
        end for- $s$ ;
      end pardo;
(3)   pardo on all PE( $i, j$ )( $u = qK + i, v = rK + j, 0 \leq i, j \leq K - 1$ ) do
           $C_{uv} \leftarrow \text{PE}(i, j)$  中的 CSUM; /*PE( $i, j$ ) 的匹配结果 CSUM 送到阵列存储器的
           $C_{uv}$ */
        end pardo;
      end for- $r$ ;
    end for- $q$ ;
  end.

```

在图像尺寸与处理元阵列大小相同的条件下, 文献 [3] 给出的通信复杂性和计算复杂性均为  $O(M^2)$  的模板匹配并行算法应该是最优的。当图像分成  $B \times B$  个块进行匹配时, 理想情况的通信复杂性和计算复杂性应均为  $O(B^2M^2)$ 。作者已经证明, 本文设计的图像分块模板匹配算法的计算复杂性为  $O(B^2M^2)$ , 通信复杂性为  $O(4B^2M^2)$ 。其通信复杂性增加的原因是由于分块处理时重复(叠)加载图像数据块造成的。所以本文设计的算法应该是最优的。图像分块模板匹配算法作者尚未见有其它报道, 所以不便于再进一步比较。

## 5 模板匹配并行算法在图像锐化中的应用

图像锐化是一种补偿图像轮廓的处理方法, 它能加强图像的边缘信息, 使图像看起来比较清晰。按锐化模板(及滤波)的意义: 对于一个  $N \times N$  的图像  $F = \{F_{i,j}\}$ , 且  $i, j \in [0, N - 1]$  和  $M \times M$  的锐化模板  $W = \{W_{st}\}$ , 且  $s, t \in [0, M - 1]$ ,  $M < N$ ,  $N$  是 2 的幂。输入图像  $F$  与高通模板  $W$  卷积所得到的输出结果  $C = \{C_{i,j}\}$  的关系可描述为

$$C(u, v) = \begin{cases} F(u, v), & (u = 0 \text{ 或 } u = N - 1) \text{ 或} \\ & (v = 0 \text{ 或 } v = N - 1) \\ \Sigma \Sigma F_{(i+s) \bmod N, (j+t) \bmod N} \times W_{st}, & (0 \leq i, j \leq N - M) \text{ 和} \\ & (u = i + 1, v = j + 1) \end{cases} \quad (8)$$

常用的锐化高通模板是拉普拉斯(Laplace)模板, 主要有<sup>[8]</sup>

$$W1 = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}, \quad W2 = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}, \quad W3 = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{bmatrix}.$$

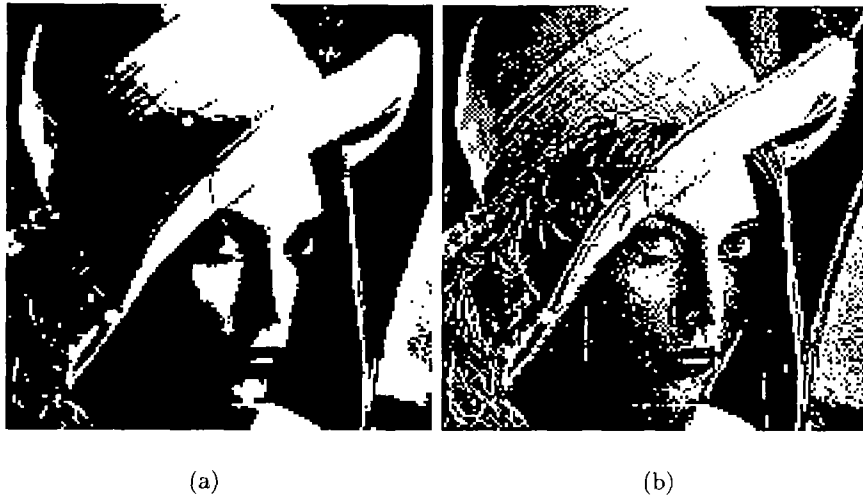


图 1 用图像分块模板匹配并行算法在 LS MPP 上实现的图像锐化效果

图 1 是图像分块模板匹配并行算法在 LS MPP 上实现的图像锐化的结果。其中图 1(a) 为  $160 \times 160$  原图像。图 1(b) 是将原图像分成  $5 \times 5$  个  $32 \times 32$  的图像块后, 用  $W1$  锐化模板锐化后的结果图像。

## 6 结束语

大多数图像处理应用都要求快速甚至实时地完成对图像的处理, 大规模并行处理技术为其提供了有效的实现途径。由于实际应用中要处理的图像大多都比并行计算机的 PE 阵列大得多, 所以研究各类图像处理和图像匹配算法的分块并行处理技术更具有理论和实践意义。本文针对 LS MPP 计算机的应用需求, 设计并实现了一种图像分块模板匹配并行算法, 解决了在小的 PE 阵列上, 实现对较大的图像进行平滑、滤波、锐化、边缘检测等的这一大类问题的并行处理问题。也可为在小的 PE 阵列机上处理较大的图像的有关图像处理任务所借鉴。

## 参 考 文 献

- [1] Z. X. Fang, X. B. Li, L. M. Ni, Parallel algorithms for image template matching on hypercube SIMD computers, IEEE Trans. on PAMI., 1987, PAMI-9(6), 835-841.
- [2] V. K. P. Kumar, V. Krishnan, Efficient image template matching on hypercube SIMD arrays, Proc. Int. Conf. on Parallel Processing, 1987, 765-771.
- [3] S. J. Horng, W. T. Chen, M. Y. Fang, Optimal speed-up algorithms for template matching on SIMD hypercube multiprocessing with restricted local memory, Information Processing Letters, 1991, 38(1), 29-37.
- [4] H. R. Tsai, S. J. Horng, *et al.*, Optimal speed-up parallel image template matching algorithms on processor arrays with a reconfigurable bus system, Computer and Vision Image Understanding, 1998, 71(3), 393-412.
- [5] 沈绪榜, MPP 嵌入式计算机设计, 北京, 清华大学出版社, 1999, 93-110.
- [6] K. Huang, Advanced Computer Architecture Parallelism Scalability Programmability, New York City, McGraw-Hill Book Co., 1993, chapter 2.
- [7] Z. Fang, L. M. Ni, On the communication complexity of generalized 2-D convolution on array processors, IEEE Trans. on Comput., 1989, C-38(2), 184-193.

[8] 张大鹏, 模式识别与图像处理并行计算机设计, 哈尔滨, 工业大学出版社, 1998, 1-8.

## PARALLEL ALGORITHM FOR TEMPLATE MATCHING OF IMAGE DIVISION BLOCK BASED ON LS MPP

Li Junshan\* \*\*      Shen Xubang\*

*\*(Xi'an Microelectronic Technology Institute, Xi'an 710054, China)*

*\*\* (The Second Artillery Engineering College, Xi'an 710025, China)*

**Abstract** This paper proposes a novel parallel algorithm for template matching of image division block based on a  $K$ -ary 2-cube network SIMD computer. For the template matching on an  $N$  by  $N$  image,  $M$  by  $M$  template and  $K$  by  $K$  processing unit array ( $M < K$ ,  $N = BK$ ,  $B > 1$ ), the algorithm is able to realize template matching by dividing an image into blocks. Compared with known algorithms, the algorithm is able to process an image whose size is larger than that of the processing unit.

**Key words** Image division block, Template matching, SIMD computer, Parallel algorithms

李俊山: 男, 1956年生, 博士生, 教授, 硕士生导师。目前主要从事图像的并行处理技术方面的研究。

沈绪榜: 男, 1937年生, 研究员, 博士生导师, 中科院院士。主要研究方向为并行嵌入式计算机体系结构与  
设计。