

# 近似欧氏距离变换的一种并行算法

崔峰 汪雪林 彭思龙

(中国科学院自动化研究所 国家专用集成电路设计工程研究中心, 北京 100080)

**摘要** 提出了一种基于超大规模集成电路(VLSI)硬件结构的新型距离变换并行处理算法。距离变换是一种基于二值图像的全局操作,在骨架抽取、形状匹配、目标重建、机器人避障等图像分析与模式识别算法中有着广泛的应用。欧氏距离是精确的 $L_2$ 范数距离,但是由于欧氏距离的非线性,不利于各种并行算法和加速算法的设计与实现,因此在应用中各种变形的加权距离作为欧氏距离的近似得到了实际推广。本文算法是有别于传统近似欧氏距离的并行计算方法,可应用于传统IC硬件或数字信号处理芯片(DSP)。理论分析和实验结果表明,该方法具有算法简单、快速、误差小等特点,可以更好地近似欧氏距离,并同时得到图像的Voronoi图,是一种实际可行的升级算法。

**关键词** 距离变换 Voronoi图 欧氏距离 并行算法 实时算法

中图分类号: TP391.41 文献标识码: A 文章编号: 1006-8961(2004)06-0693-06

## A Parallel Algorithm for Quasi Euclidean Distance Transform

CUI Feng, WANG Xue-Lin, PENG Si-Long

(NADEC, Institute of Automation, Chinese Academy of Science, Beijing 100080)

**Abstract** A novel parallel algorithm for distance transformation based on VLSI hardware structure is proposed in this paper. Distance transformation is a global operation on binary image which is widely used in skeletonization, shape matching, object reconstructing, obstacles elusion, etc. Because of the nonlinearity of euclidean distance, it is not convenient to design parallel and fast algorithm for it. Many deformed weighed distance transformation is adopted in practice. In this paper, we propose a parallel algorithm deferent from the traditional Quasi-Euclidean distance transformation. It can be used in traditional IC hardware and DSP. Theoretical analysis and experiments show that the algorithm is simple, fast and the results can approximate the Euclidean distance more exactly. The Voronoi diagram can be got synchronously. Simulation results on PC are also listed in this paper.

**Keywords** distance transformation, Voronoi diagram, Euclidean distance, parallel algorithm, real time algorithm

## 1 引言

距离变换(distance transform, DT)<sup>[1,2]</sup>是一种基于二值图像的全局操作,在骨架抽取、形状匹配、目标重建、机器人避障等图像分析与模式识别算法中有着广泛的应用。距离变换的结果不是另一幅二值图像,而是一幅灰度图像,每个像素的灰度级代表了该像素与距离最近的特征像素间距离大小的测度。

一般说来,需要进行距离变换的原始图像是灰度图像,在对原图像进行二值化后,再进行距离变换。距离变换后得到的灰度图像包含着细化了的几

何信息。距离变换的距离函数有很多,这是因为不同的变换函数对于不同的应用有不同的功效。欧氏距离变换是精确的 $L_2$ 范数距离变换,可以适用于各种不同的场合,但是欧氏距离变换是一种非线性距离变换,耗时很大,不利于各种并行算法和加速算法的设计与实现。Breu, Gil等人提出了一种计算欧氏距离的线性时间算法<sup>[3]</sup>,按照图像中每行欧氏距离的递推延续性质对全局操作进行细化,分割成局部线性时间操作。Maurer, Raghavan等人对该方法进行了改进<sup>[4]</sup>,使其可以扩展到任意维。Borgefors<sup>[5]</sup>、Danielsson<sup>[6]</sup>等人提出了应用各像素附近小邻域的距离变换值进行更新迭代的欧氏距离变换算法,

Yamada 将该方法应用于并行计算<sup>[7]</sup>,采用固定的邻域大小迭代到收敛从而得到距离图像。Pan, Li 等人应用光学总线(optical bus)搭建的 LARPBS 系统<sup>[8]</sup>也有效地将距离变换应用于并行计算。管伟光、马颂德提出了一种欧氏距离变换的线性快速算法<sup>[9,10]</sup>。这些方法的计算复杂度都比较大,需要较多的存储空间、计算时间与控制代价。

由于欧氏距离的局限性,在实际应用中,各种变形的加权距离作为欧氏距离的近似得到了推广。在实际应用中,计算某一个像素到本图像所有特征点的最短距离是一种全局操作,耗费的计算资源和硬件资源非常大,通常都是将全局操作划分为局部操作来实现。街区距离、棋盘距离、倒角距离<sup>[5,11~13]</sup>,近似欧氏距离<sup>[5,6,12,14]</sup>等都是对欧氏距离的近似,Fujiwara, Inoue<sup>[15,16]</sup>, Lee 和 Horng<sup>[17,18]</sup>等人也提出了一系列加权距离的快速扫描和并行算法。加权距离虽然可以满足大部分应用的需要,但是在精度方面与欧氏距离相比还是存在较大的差距。

本文提出了一种新颖的距离变换并行算法,它有别于传统的欧氏距离和加权距离,其计算过程是线性传递的,而相对于欧氏距离的误差则是非线性分布的,它既具有加权距离所具有的简单、快速,占用资源少的优点,又使计算误差更小,是一种介于欧氏距离和加权距离之间的距离变换并行算法。该方法在计算距离变换的同时,可以得出图像的最近邻点变换和 Voronoi 图。

## 2 距离变换的相关概念

### 2.1 距离变换与最近邻特征点变换的定义

设  $I$  为一幅  $n \times n$  的二值图像,  $I \in \{0, 1\}$ ,  $I(i, j)$  表示图像中行坐标为  $i$ , 列坐标为  $j$  像素点的值 ( $1 \leq i \leq n, 1 \leq j \leq n$ ), 设  $I(1, 1)$  代表图像中最左上角的像素。  $I(i, j) = 1$  表示点  $(i, j)$  是特征像素(白色),  $I(i, j) = 0$  表示点  $(i, j)$  是背景像素(黑色)。对于  $I$  中的任意 2 点  $p_1(i_1, j_1), p_2(i_2, j_2)$ , 可以定义它们之间的距离测度:

欧氏距离

$$d_E(p_1, p_2) = \sqrt{(i_1 - i_2)^2 + (j_1 - j_2)^2} \quad (1)$$

加权距离

$$d_w(p_1, p_2) = \begin{cases} w_0|i_1 - i_2| + w_1|j_1 - j_2| & |i_1 - i_2| > |j_1 - j_2| \\ w_1|i_1 - i_2| + w_0|j_1 - j_2| & |i_1 - i_2| \leq |j_1 - j_2| \end{cases} \quad (2)$$

其中,  $w_0, w_1$  为非负常数。通过定义不同的  $w_0, w_1$  值, 可以得到各种不同的加权距离。例如:

$$d_w \begin{cases} \text{街区距离} & (w_0, w_1) = (1, 1) \\ \text{棋盘距离} & (w_0, w_1) = (1, 0) \\ \text{2-3 倒角距离} & (w_0, w_1) = (2, 3) \\ \text{近似欧式距离} & (w_0, w_1) = (\sqrt{2} - 1, 1) \end{cases} \quad (3)$$

可以看出, 欧氏距离是一种两点间坐标值之差的非线性距离度量, 而各种加权距离则是线性的。基于二值图像的距离变换是求出每个像素点的最近邻特征点, 将这两点之间的距离值作为该点在距离图上对应的灰度值, 同时距离图上的各点还包含了原二值图像上对应各点的最近邻特征点的坐标信息, 从而得到距离图像。显然每个特征像素的最近邻特征点就是其本身, 特征点在距离图上的值为 0。因此可以这样定义距离变换:

$$DT[i, j] = \min\{d(p, p_w) | p_w \in \text{white}\} \quad (4)$$

这里假设  $d(p, p_w)$  表示像素  $p(i, j)$  与特征点像素  $p_w$  的距离,  $\text{white}$  代表全体特征像素(白点)的集合。最近邻特征点变换(nearest feature transform, NFT)就是求出与该像素距离最近的特征点的坐标向量:

$$NFT[i, j] = \{(x, y) | d(p(i, j), p_w(x, y)) = DT[i, j]\} \quad (5)$$

由以上定义可知, 对应于某种距离定义, 每幅图像的距离变换是唯一的, 而最近邻特征点变换是不唯一的, 因为像素的最近邻特征点并不一定唯一, 可能有数个点与该像素的距离都等于该点的距离变换值。可以看出, 只要得到了最近特征点变换, 就可以很容易地求出距离变换。

### 2.2 Voronoi 图的定义

在定义了最近邻特征点变换后, 就可以得出 Voronoi 图的定义。将最近邻特征点相同的像素各自归为一类, 这样就完成了对原图像空间的一个完备划分, 得到的分割图像称为原图像的 Voronoi 图。

### 2.3 计算模型

采用一种并行计算普遍应用的 PRAM 模型解释算法结构。PRAM 是一种 Systolic 网格逻辑阵列, 其  $n \times n$  的网孔结构可以使得 4 邻域连接的各个处理单元(PE)在共享的记忆空间中同步访问任何存储单元, 每个处理单元可以独立完成乘、加等基本操作, 是单指令流多数数据流(SIMD)结构。实际应用中每个 PE 和存储单元可以分别对应一个像素位置。

### 3 近似欧氏距离变换的并行算法

#### 3.1 基本思想

为了计算每个像素点的距离变换和最近邻特征点变换,先以每个像素  $p(i, j)$  为基点将输入图像(尺寸为  $n \times n$ )以过该点的水平和竖直线为基准,划分为 4 个区域  $P_{RU}, P_{RD}, P_{LU}, P_{LD}$ , 它们的数学定义如下:

$$P_{RU} = \{p(i-g, j+h) | 0 \leq g \leq i-1, 0 \leq h \leq n-j\} \quad (6)$$

$$P_{RD} = \{p(i+g, j+h) | 0 \leq g \leq n-i, 0 \leq h \leq n-j\} \quad (7)$$

$$P_{LU} = \{p(i-g, j-h) | 0 \leq g \leq i-1, 0 \leq h \leq j-1\} \quad (8)$$

$$P_{LD} = \{p(i+g, j-h) | 0 \leq g \leq n-i, 0 \leq h \leq j-1\} \quad (9)$$

图 1 为示意图。

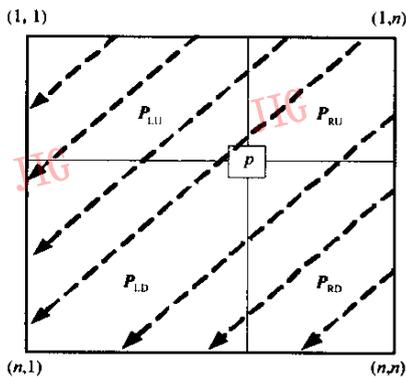


图 1 算法示意图

应用这 4 个分区,按以下步骤计算距离变换:

- (1) 在  $P_{RU}$  区域内寻找距离每个像素  $p(i, j)$  的最近特征点;
- (2) 在  $P_{RD}$  区域内寻找距离每个像素  $p(i, j)$  的最近特征点;
- (3) 在  $P_{LU}$  区域内寻找距离每个像素  $p(i, j)$  的最近特征点;
- (4) 在  $P_{LD}$  区域内寻找距离每个像素  $p(i, j)$  的最近特征点;
- (5) 将以上求得的 4 个点进行比较,得到每个像素  $p(i, j)$  的最近邻特征点  $NFT[i, j]$ ;
- (6)  $p(i, j)$  与  $NFT[i, j]$  的距离即为每个像素的距离变换  $DT[i, j]$ 。

显而易见,步骤 1~步骤 4 可以通过相似的方法并行地计算,步骤 5、6 的计算对各个像素来说也是独立的。所以以如何在  $P_{RU}$  区域内寻找  $p(i, j)$  的最近邻特征点为例说明算法。

对于  $P_{RU}$  区域内任意一点  $p_{RU}(i_{RU}, j_{RU}) \in P_{RU}$ ,  $p(i, j)$  与  $p_{RU}(i_{RU}, j_{RU})$  的欧式距离定义为

$$d_E(p, p_{RU}) = \sqrt{(i - i_{RU})^2 + (j - j_{RU})^2}$$

#### 3.2 基本假设

如图 2 所示,假设  $p(i, j)$  的  $P_{RU}$  区域内有任意 2 点  $p_1(i_1, j_1), p_2(i_2, j_2)$ , 并且  $d_E(p, p_1) \leq d_E(p, p_2)$ , 在  $p(i, j)$  的  $-\frac{3}{4}\pi$  方向上有一点  $p'(i+m, j-m)$ , 其中  $m$  是自然数,假设  $d_E(p', p_1) \leq d_E(p', p_2)$  亦成立。

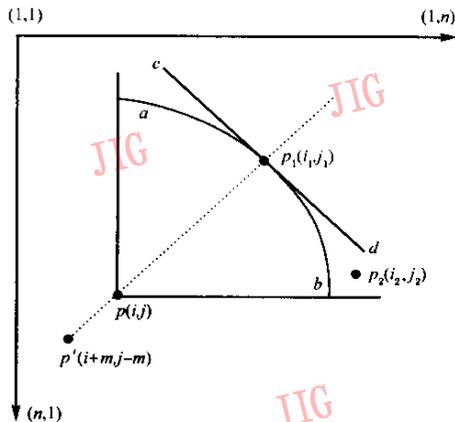


图 2 假设示意图

实际上,这一假设并不是严格成立的,下面做一下推导。由

$$d_E(p, p_1) \leq d_E(p, p_2) \Rightarrow (i_1 - i)^2 + (j_1 - j)^2 \leq (i_2 - i)^2 + (j_2 - j)^2 \quad (10)$$

$$d_E(p', p_1) \leq d_E(p', p_2) \Leftrightarrow (i_1 - i - m)^2 + (j_1 - j + m)^2 \leq (i_2 - i - m)^2 + (j_2 - j + m)^2 \Rightarrow (i_1 - i)^2 - 2m(i_1 - i) + m^2 + (j_1 - j)^2 + 2m(j_1 - j) + m^2 \leq (i_2 - i)^2 - 2m(i_2 - i) + m^2 + (j_2 - j)^2 + 2m(j_2 - j) + m^2 \Leftrightarrow (i_1 - i)^2 + (j_1 - j)^2 + 2m[(i - i_1) - (j - j_1)] \leq (i_2 - i)^2 + (j_2 - j)^2 + 2m[(i - i_2) - (j - j_2)] \quad (11)$$

由式(10)可知式(11)成立的一个充分条件是:

$$(i - i_1) - (j - j_1) \leq (i - i_2) - (j - j_2) \Leftrightarrow i_1 - i_2 \geq j_1 - j_2$$

但是得不到使式(11)成立的充要条件。从图 2 可以看出,以  $p(i, j)$  为圆心,  $pp_1$  为半径作圆弧  $\widehat{ap_1b}$ ,  $p_2(i_2, j_2)$  一定在  $\widehat{ap_1b}$  以外,如果作过  $p_1$  点  $-\frac{1}{4}\pi$  角度的直线  $cd$ , 则当  $p_2$  点位于  $\widehat{ap_1b}$  与  $cd$  之间时,有  $\frac{i_1 - i_2}{j_1 - j_2} > 1$ 。

继续由图 3 定性地说一下基本假设的误差分布。

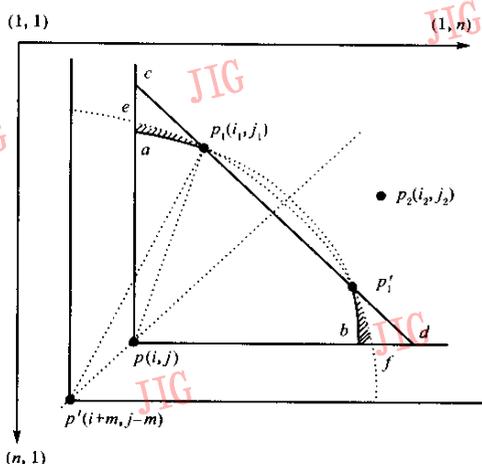


图 3 假设误差示意图

图 3 是对图 2 的扩展,  $\widehat{ep_1f}$  是以  $p'(i+m, j-m)$  为圆心,  $p'p_1$  为半径的圆弧, 如果  $p_1$  不在  $p'p$  的延长线上, 则  $\widehat{ep_1f}$  与  $\widehat{ap_1b}$  必然有关于  $p'p$  延长线对称的另一个交点  $p'_1$ . 过  $p_1, p'_1$  的直线, 交  $P_{RU}$  的边界于  $c, d$ . 因为  $d_E(p, p_1) \leq d_E(p, p_2)$ , 所以  $p_2$  一定在圆弧  $\widehat{ap_1b}$  以外, 而如果  $d_E(p', p_1) \leq d_E(p', p_2)$ , 则  $p_2$  也必须满足在圆弧  $\widehat{ep_1f}$  之外的充要条件, 显然在  $P_{RU}$  中只有  $p_2$  落在  $\widehat{ap_1e}$  和  $\widehat{bp'_1f}$  所形成的 2 个狭小区域内 (图 3 中阴影部分) 才会不满足基本假设, 这就是误差区域。如果  $p, p'$  之间的距离增大时,  $\widehat{ep_1f}$  的曲率会减小, 极限情况是  $\widehat{ep_1f}$  与直线  $cd$  重合, 所以在  $p, p'$  距离很大以及  $p_1$  很接近  $p'p$  延长线时, 误差区域会变大。经实验证明这些误差比传统近似欧氏距离小得多, 因此总是假定上述基本假设成立。

### 3.3 基本算法

以  $P_{RU}$  为例, 定义  $NFT_{RU}[i, j]$  表示  $P_{RU}$  区域内  $p(i, j)$  的最近邻特征点, 对于像素  $p''(i-1, j+1)$ , 其右上区域用  $P''_{RU}$  表示,  $NFT_{RU}[i-1, j+1]$  表示在  $P''_{RU}$  区域内的最近邻特征点。定义一个单像素宽的拐角带为  $DI(i, j) = (P_{RU}(i, j) - P''_{RU}(i-1, j+1))$ , 如图 4 灰色区域所示。  $ruu$  表示该带状区域的上分支部分,  $rur$  代表该带状区域的右分支部分。

用  $NFT_{RUD}[i, j]$  表示  $DI(i, j)$  子带内距离  $p(i, j)$  最近的特征点, 由基本假设, 有

$$NFT_{RU}[i, j] = \begin{cases} NFT_{RU}[i-1, j+1] & \text{如果} \\ d_E(p, NFT_{RU}[i-1, j+1]) \leq d_E(p, NFT_{RUD}[i, j]) & \\ NFT_{RUD}[i, j] & \text{其他} \end{cases} \quad (13)$$

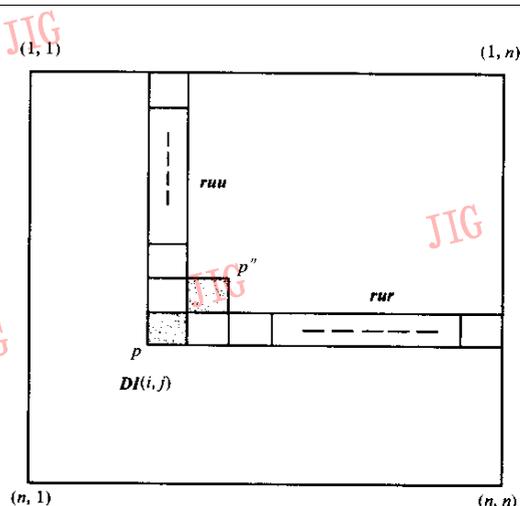


图 4 计算子带示意图

如果在  $DI(i, j)$  或  $P''_{RU}$  区域内没有特征点, 则定义  $NFT_{RUD}[i, j]$  或  $NFT_{RU}[i-1, j+1]$  为  $(\infty, \infty)$ 。由此可见, 如果分别计算出各个  $NFT_{RUD}[i, j]$  的值, 再依次从图像边沿的右上方沿  $-\frac{3}{4}\pi$  方向往左下方进行最小值比较与替换操作, 最终就得到了各个点的  $NFT_{RU}$  值。同理, 用类似的方法从右下方沿  $\frac{3}{4}\pi$  方向扫描比较得到各点  $NFT_{RD}$  值; 从左上方沿  $-\frac{1}{4}\pi$  方向扫描比较得到各点  $NFT_{LU}$  值; 从左下方沿  $\frac{1}{4}\pi$  方向扫描比较得到各点  $NFT_{LD}$  值, 求取  $NFT_{RU}$  扫描行程的几个例子在图 1 中用虚线示例。以上步骤可以并行计算, 最后将各个点的 4 个值进行比较就得到图像的距离变换与最近邻特征点变换的最终结果。

### 3.4 算法流程

仍以计算像素  $p(i, j)$  的  $NFT_{RU}$  为例简要说明算法流程。

(1) 对于每个像素  $p(i, j)$ , 令矩阵  $A, B$  分别存储  $P_{RU}$  区域最近邻特征点变换与距离变换结果。如果  $I(i, j) = 1$ , 则  $A[i, j] = (i, j), B[i, j] = 0$ ; 否则  $A[i, j] = (\infty, \infty), B[i, j] = \infty$ 。

(2) 对于  $A[i, j] = (\infty, \infty)$  的每个像素  $p(i, j)$ , 在  $DI(i, j)$  区域内计算  $NFT_{RUD}[i, j]$ , 将结果和距离值存入矩阵  $A, B$  对应位置。

由于在  $DI(i, j)$  中采用欧式距离测度, 而  $ruu$  和  $rur$  区域内的各点与  $p(i, j)$  仅有一个坐标不相等, 所以大大简化了欧氏距离的求取复杂度。将  $ruu$  和  $rur$  中的白点分别列出, 设  $ruu$  中与  $p$  点横坐标  $i$  值差别最小的特征点为  $p_{ruu}(i_{ruu}, j)$ , 设  $rur$  中  $p$  点与

横坐标  $j$  值差别最小的特征点为  $p_{rur}(i, j_{rur})$ , 则

$$A[i, j] = \begin{cases} (i_{ruu}, j) & \text{如果 } (i_{ruu} - i) \leq (j_{rur} - j) \\ (i, j_{rur}) & \text{其他} \end{cases}$$

$$B[i, j] = \begin{cases} i_{ruu} - i & \text{如果 } (i_{ruu} - i) \leq (j_{rur} - j) \\ j_{rur} - j & \text{其他} \end{cases}$$

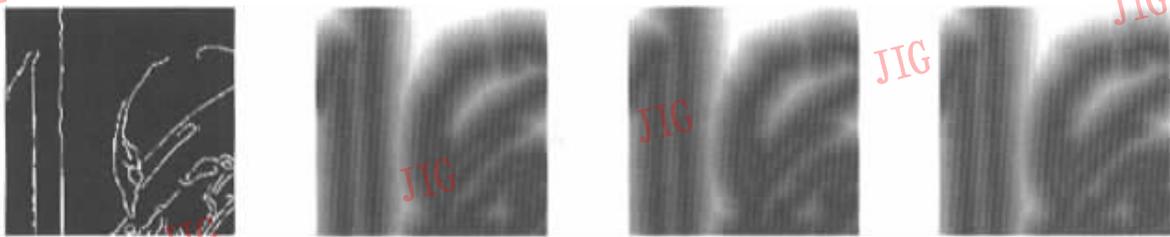
(3) 以图像上边界和右边界的各个像素为起点, 向  $-\frac{3}{4}\pi$  方向扫描比较, 直到各个行程到达图像的左边界或下边界。最终的  $A, B$  就是  $P_{RU}$  区域最近邻特征点变换与距离变换结果。行程中  $A, B$  的替换规则如下:

- ①  $A[i, j] = \begin{cases} A[i, j] & \text{如果 } B[i, j] = 0 \\ A[i, j] & \text{如果 } B[i, j] \leq d_E(p(i, j), A[i-1, j+1]) \\ A[i-1, j+1] & \text{其他} \end{cases}$
- ②  $B[i, j] = d_E(p(i, j), A[i, j])$
- ③ 如果  $i < i_0$  且  $j > 1, i = i + 1, j = j - 1$ , 转至①

用类似的方法分别计算出各像素的  $NFT_{RD}, NFT_{LU}, NFT_{LD}$ , 进行比较后就得到了原图像的距离变换结果和最近邻特征点变换结果。

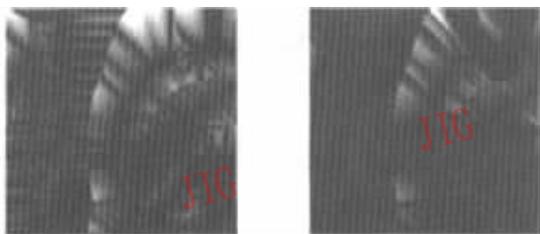
### 4 算法复杂度及实验结果分析

因为本文算法仅在替换比较阶段才会进行较复杂的乘、开方运算, 且 4 个部分可以并行处理, 所以计算复杂度与控制代价都较低。



(a) 输入二值图像 (b) 欧氏距离计算结果 (c) 近似欧式距离计算结果 (d) 本文方法的计算结果

图 5 实验结果示意图



(a) 传统的近似欧氏距离计算结果误差分布图 (b) 本文方法的计算误差分布图

图 6 误差分布比较图

假设图像大小为  $n \times n$ , 含有  $m$  个特征点。

在计算各个子带内的最近邻特征点期间, 每个特征点会有  $4n-1$  次落在 4 个分区的子带内, 也就会有  $4n-1$  次减法运算进行判别比较。由于该阶段是完全并行的, 所以运算时间是一个减法时间和一个比较时间;

在寻找各像素位置在各个分区内的最近邻特征点期间, 假设原图的  $m$  个特征点均不同行不同列, 这样需要计算的特征点数最多, 为  $m$  个。由于该阶段 4 个分区可并行处理, 每个分区的串行扫描也可以并行处理, 需要扫描行程最长的对角线的计算花费最多是  $m-1$  次欧式距离运算和比较, 此即为系统耗时最大值。

在最后的 4 分区合成比较阶段, 各点仍然是并行处理, 系统耗时为一个比较时间。

图 5(a) 是 Lena 图像的局部边缘图, 将其作为测试图像, 对本文算法、标准欧氏距离算法、传统上与欧式距离最接近的近似欧氏距离算法 (Quasi-Euclidean) 进行对比实验。图 5(b) 为标准欧氏距离的计算结果, 将其作为标准信号, 分别计算图 5(c)、图 5(d) 与图 5(b) 的误差分布, 如图 6 所示。近似欧氏距离结果与欧氏距离结果的峰值信噪比为 16.448 dB, 本文方法与欧氏距离结果的峰值信噪比为 23.169 dB, 提高了 40.86%, 而在精确度方面也说明了本文方法的优越性。

### 5 结 论

本文提出了一种有别于传统近似欧氏距离变换的并行算法。经实验验证, 该算法具有简单、复杂度低、控制代价小等特点, 并获得了比较满意的效果。另外, 此方法在计算距离变换的同时完成了最近邻特征点变换, 从而也可以很方便地求得原图像的 Voronoi 图。进一步的工作是对算法中的冗余部分

进行分析,减少运算区域,进一步提高运算效率。

### 参 考 文 献

- 1 Rosenfeld A, Pfaltz J L. Sequential operations in digital picture processing [J]. Journal of the Association for Computing Machinery, 1966, **13**(4):471~494.
- 2 Castleman Kenneth R. Digital Image Processing[M]. Prentice Hall. Inc, New Jersey, USA, 1996.
- 3 Breu Heinz, Gil Joseph, Kirkpatrick David, *et al.* Linear time Euclidean distance transform algorithm[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1995, **17**(5): 529~533.
- 4 Maurer Calvin R, Raghavan Vijay, Qi Rensheng. A linear time algorithm for computing the Euclidean distance transform in arbitrary dimensions[A]. In: 17th International Conference of Information Processing in Medical Imaging [C], Davis, CA, USA, 2001: 358~364.
- 5 Borgefors G. Distance transformations in digital images [J]. Computer Vision, Graphics, and Image Processing, 1986, **34**(3):344~371.
- 6 Danielsson P E. Euclidean distance mapping [J]. Computer Graphics and Image Processor, 1980, **14**(2):227-248.
- 7 Yamada H. Complete Euclidean distance transformation by parallel operation [A]. In: Proc. Seventh Int'l Conf. Pattern Recognition[C], Montreal, Canada, 1984:69~71.
- 8 Pan Yi, Li Yamin, Li Jie, *et al.* Efficient parallel algorithms for distance maps of 2-D binary images using an optical bus [J]. IEEE Transactions on System, Man, And Cybernetics-Part A: Systems And Humans, 2002, **32**(2):228~235.
- 9 Guan Weiguang, Ma Songde. A fast unified distance transformation algorithm [J]. Chinese Journal of Computers, 1995, **18**(8):626~635.
- 10 Guan Weiguang, Ma Songde. A list-processing approach to compute Voronoi diagrams and the Euclidean distance transform [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1998, **20**(7):757~761.
- 11 Das P P, Chakrabarti P P. Distance functions in digital geometry [J]. Information Science, 1987, **42**(2):113~136.
- 12 Yamashita M, Ibaraki T. Distances defined by neighborhood sequences[J]. Pattern Recognition, 1986, **19**(3):237~246.
- 13 Borgefors G. Distance transformations in arbitrary dimensions [J]. Computer Vision, Graphics, and Image Processing, 1984, **27**(3):321~345.

- 14 Borgefors G. A new distance transformation approximation the Euclidean distance [A]. In: Proc. International 1st Joint Conference on Pattern Recognition [C], London, UK, 1986: 336~338.
- 15 Fujiwara Akihiro, Inoue Michiko, Masuzawa T, *et al.* A simple parallel algorithm for the medial axis transform of binary images [A]. In: Proc. IEEE Second International Conference on Algorithms and Architecture for Parallel Processing [C], Singapore, 1996: 1~8.
- 16 Fujiwara Akihiro, Inoue Michiko, Masuzawa T, *et al.* A parallel algorithm for weighted distance transforms [A]. In: Proc. IEEE International Parallel Processing Symposium [C], Geneva, Switzerland, 1997:407~412.
- 17 Lee Y -H, Horng S -J. Fast parallel chessboard distance transform algorithms [A]. In: Proc. 1996 International Conference on Parallel and Distributed Systems [C], Tokyo, Japan, 1996: 488~493.
- 18 Lee Y -H, Horng S -J. The chessboard distance transform and the medial axis transform are interchangeable [A]. In: Proc. 10th International Parallel Processing Symposium [C], Honolulu, HI, USA, 1996: 424~428.



崔峰 1977年生,2000年于北京化工大学获自动化学士学位,现在中国科学院自动化所国家专用集成电路中心攻读模式识别与智能系统方向的博士学位。研究兴趣是图像处理、并行算法、时频分析。



汪雪林 1976年生,1999年于北京理工大学获自动化学士学位,现在中国科学院自动化所国家专用集成电路中心攻读小波分析与图像处理方向的博士学位。研究兴趣是图像压缩、视频压缩、图像视频超分辨率技术。



彭思龙 1971年生,研究员、博士生导师。1998年获中科院数学所基础数学专业理学博士学位。主要研究领域为小波理论及应用、图像处理、模式识别和积分方程数值解。发表论文 30 余篇。