

数字图像处理中的几何形状 变换及图像数据组织

杨世仁

(中国科学院空间科学技术中心遥感应用研究所)

摘 要

本文提出一种实现图像几何形状变换及图像数据分解成子矩阵的算法和装置。由于采用软件和专用解算装置相结合的方法,因此能有效地节省计算机处理时间,并适用于在小型计算机图像处理系统中,实现影像的几何畸变纠正及影像和线划图的投影变换。

一、引 言

在遥感影像处理和自动制图等领域中,常常需要进行图像的几何纠正或变换。

在陆地卫星影像预处理系统和高性能的计算机图像处理系统中,通常用专门设计的阵列处理机来完成影像的几何纠正。在小型图像处理系统中,进行几何纠正需要的机器时间很长,在软件设计上也存在不少麻烦。这里提出的方法是在图像数据用行扫描方式输入和输出计算机过程中,用简单的专用解算装置来实现图像的几何变换。用这种方法进行几何变换几乎不占计算机时间,并能用在小型计算机图像处理系统中。

本文中提出改进图像输入输出设备和计算机之间的直接数据通道接口,使图像数据以行扫描方式输入计算机时,能自动把图像数据转换成子矩阵形式的一种方法。子矩阵形式的图像数据在输出记录时同样能自动改变成记录行数据。这种影像数据表达方式的自动变换,可以节省计算机处理时间,简化计算机的程序设计。

上述方法同样适用于以计算机兼容磁带作为图像数据输入输出的计算机图像处理系统中。

二、图像几何变换的算法

图像的几何变换指的是将图像 $F(x, y)$ 变换成 $G_c(\xi, \eta)$, 而

$$\xi = f_1(x, y), \quad \eta = f_2(x, y). \quad (2-1)$$

其中 x, y 为像元在原稿 F 上的坐标, ξ, η 为该像元在变换后的图像 G_c 上的对应坐标。函数 $f_1(x, y)$ 和 $f_2(x, y)$ 有时是解析地推导出来的,但为了计算的方便,常用多项式

本文 1978 年 11 月收到。

$$\xi = f_1(x, y) = \sum_{j=0}^n \sum_{k=0}^{n-j} a_{jk} x^j y^k, \quad (2-2a)$$

$$\eta = f_2(x, y) = \sum_{j=0}^n \sum_{k=0}^{n-j} b_{jk} x^j y^k \quad (2-2b)$$

来逼近。如果无法用分析方法求出函数 $f_1(x, y)$ 和 $f_2(x, y)$, 而已知图像 F 上 i 个控制点 $p_i (i = 1, \dots, j)$ 在几何变换前后的位置, 则可用最小二乘法来求得多项式 (2-2a) 和 (2-2b)。控制点在 F 上的座标可以在图像显示器上放大比例尺后测定, 控制点在 G_c 上的座标是已知的。

在几何变换计算中, 一般使用 (2-2a) 和 (2-2b) 式的反函数, 或根据控制点在图像 F 和 G_c 上的座标对应关系, 求出多项式

$$x = f_3(\xi, \eta) = \sum_{j=0}^n \sum_{k=0}^{n-j} a'_{jk} \xi^j \eta^k, \quad (2-3a)$$

$$y = f_4(\xi, \eta) = \sum_{j=0}^n \sum_{k=0}^{n-j} b'_{jk} \xi^j \eta^k. \quad (2-3b)$$

对于图像 $G_c(\xi, \eta)$ 上座标为 $\xi, \eta (\xi = 1, 2, \dots, N; \eta = 1, 2, \dots, M)$ 的所有像元, 可以用 (2-3a) 和 (2-3b) 式求出在 $F(x, y)$ 上的对应座标 x, y 。 x, y 不一定在 F 的像元格点中心。通常用最近邻点法, 线性插值法或三次褶积法插值来求出原稿 F 在 x, y 点上的灰度值 $f_{x,y}$, 并作置换

$$g_{\xi, \eta} = f_{x,y}. \quad (2-4)$$

当图像用行扫描方式输入计算机时, y 和 x 分别表示顺扫描线和垂直于扫描线方向的位置, 因此进行几何变换计算时, 计算机内存中应当保持足够多个扫描行的数据。当影像需要旋转较大的角度时, 在内存中应当保存的扫描行数目更多, 而一般小型计算机难以完成。

由于图像是用行扫描方式数字化的, 对于第 x_i 扫描行, $x = x_i$ 为常数, (2-2b) 式可改写成

$$\eta = F_2(y)_{x=x_i} = \sum_{k=0}^n \left(\sum_{j=0}^{n-k} b_{jk} x_i^j \right) y^k = \sum_{k=0}^n B_k(x_i) y^k. \quad (2-5)$$

对于该扫描行中的所有 y 值, 可以用上式算出 η 值, 再用插补法求出格点中心像元的灰度值。用 (2-5) 式, 只须在每一扫描行开始前计算一次 $B_k(x_i)$, 然后用一元多项式来计算 η 值, 因此只需在计算机内存中保存一扫描行图像数据, 并减少计算时间。

当整张图像各扫描行的数据全部用 (2-5) 式计算完毕后, 图像矩阵 $F(x, y)$ 已转换成 $G_1(x, \eta)$, 如图 1(b) 所示。现在, 应当先把矩阵 $G_1(x, \eta)$ 旋转 90° 而得到图像矩阵 $G_2(\eta, x)$ 。在顺时针向旋转时

$$G_2(\eta, x) = G_1^T(x, \eta) \cdot J. \quad (2-6)$$

其中 G_1^T 为 G_1 的转置矩阵, 而 J 为排列矩阵

$$J = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 1 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 1 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \end{bmatrix}. \quad (2-7)$$

根据图像矩阵 $G_1(\eta, x)$ 和 $G_c(\eta, \xi)$ 上 j 个已知控制点 $p_i (i = 1, \cdots, j)$ 的对应关系, 可以用最小二乘法求出多项式

$$x = f_2(\eta, \xi) = \sum_{k=0}^n \sum_{j=0}^{n-k} a''_{jk} \eta^j \xi^k = \sum_{k=0}^n A''_k(\eta) \xi^k. \quad (2-8)$$

对于 $G_2(\eta, x)$, 由于它是 $G_1(x, \eta)$ 的 90° 旋转, 因此, x 可看成是第 η 列上的变量. 对于 $\eta = \eta_i$, (2-8) 式中的系数 $A''_k(\eta)$ 变成常数而可改写为

$$x = F_1(\xi)_{\eta=\eta_i} = \sum_{k=0}^n A''_k(\eta_i) \xi^k. \quad (2-9)$$

对于图像 $G_c(\eta_i, \xi)$ 上座标为 $\xi = 1, 2, \cdots, M$ 的所有像元, 可以用 (2-9) 式求出在 $G_2(\eta_i, x)$ 上的对应座标值 x , 并用插值法求出 x 点的灰度值 f_{x, η_i} 而作置换

$$g_{\xi, \eta_i} = f_{x, \eta_i}.$$

用 (2-2a) 和 (2-2b) 式进行变换而用三次褶积法插值时, 格点 k, l 上的灰度值可用下式计算

$$g_{kl} = S(y_{ij}) D_l(f_{ij}) S'(x_{ij}). \quad (2-10)$$

其中

$$S(u_{ij}) = [f(u_{11}), f(u_{12}), \cdots, f(u_{1n}), f(u_{21}), \cdots, f(u_{2n}), \cdots, f(u_{m1}), \cdots, f(u_{mn})],$$

$$D(f_{ij}) = \text{diag}[f_{11}f_{12} \cdots f_{1n} f_{21} \cdots f_{2n} \cdots f_{m1} \cdots f_{mn}].$$

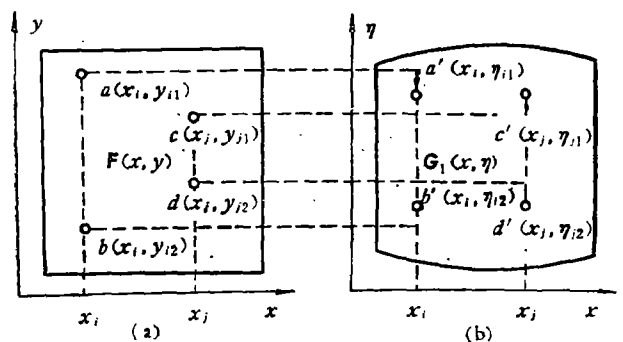
这里 $f(u_{ij})$ 为 $\sin c(u_{ij})$ 的三次逼近式, f_{ij} 为第 ij 点像元灰度值, u_{ij} 为第 ij 点到插值点 k, l 的距离 (见图 2). 将 (2-10) 式展开得

$$g_{kl} = \sum_{i=1}^4 \left(\sum_{j=1}^4 f_{ij} S(x_{ij}) \right) S(y_{ij}) = \sum_{i=1}^4 \left(\sum_{j=1}^4 f_{ij} S(y_{ij}) \right) S(x_{ij}). \quad (2-11)$$

在两个方向分别用 (2-5) 和 (2-9) 式进行变换, 相当于先进行内括号中的总加, 再进行括号外的总加.

图 1(a) 到 1(d) 说明上述变换过程. 图 1(a) 中的图像矩阵 $F(x, y)$ 经 (2-5) 式变换成图 1(b) 中的图像矩阵 $G_1(x, \eta)$. 图 1(a) 中的像元 $a(x_i, y_{11})$, $b(x_i, y_{12})$, $c(x_i, y_{11})$ 和 $d(x_i, y_{12})$ 在图 1(b) 中分别移到了 $a'(x_i, \eta_{11})$, $b'(x_i, \eta_{12})$, $c'(x_i, \eta_{11})$ 和 $d'(x_i, \eta_{12})$. 在此变换中, 像元只沿 y 方向移到对应的 η 而 x 保持不变. 然后将 $G_1(x, \eta)$ 旋转 90° , 变成 $G_2(\eta, x)$, 而改变了行和列的位置. 为了说明方便, 设像元 b' 和 d' , a' 和 c' 正好在 $G_1(x, \eta)$ 的同一行中, 旋转 90° 后它们分别在 $G_2(\eta, x)$ 的同一列中. 最后, 用 (2-9) 式的反函数把 $G_2(\eta, x)$ 变换成 $G_c(\eta, \xi)$. 这时, 图 1(c) 中的像元 $b'(\eta_1, x_i)$, $d'(\eta_1, x_j)$, $a'(\eta_2, x_i)$ 和 $c'(\eta_2, x_j)$ 在图 1(d) 中分别移到 $B(\eta_1, \xi_{11})$, $D(\eta_1, \xi_{11})$, $A(\eta_2, \xi_{12})$ 和 $C(\eta_2, \xi_{12})$. 在这一变换中像元只在 x 方向移动到 ξ 而 η 方向不变.

用控制点进行图像的几何变换时, 图像上应当有一定数目分布合适的控制点, 并且多



(a) 原稿图象

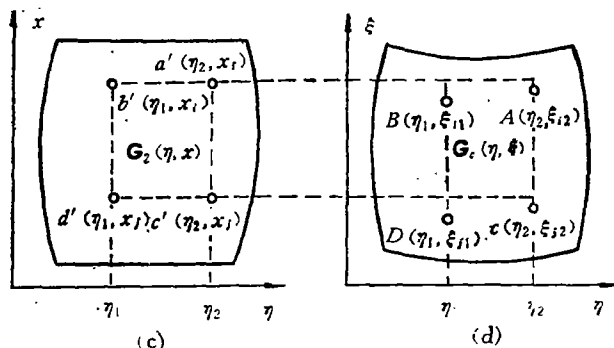
(b) 经过 $\eta = F_2(y)_{x=x_i}$ 变换(c) 顺时针旋转 90° (d) 经过 $\xi = F(x)_{\eta=\eta_i}$ 变换

图1 图像在行和列方向分开进行几何变换

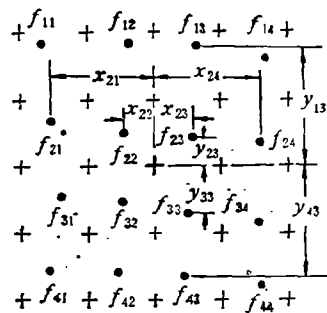


图2 三次褶积法插值

项式的阶数应该取得恰当,才能保证几何变换的精度。阶数过高不但增加计算量,而且计算的舍入误差有时反会降低精度。

得到几何变换式后,可以把图像分成若干个较小的子矩阵,或在行扫描方式下变换时把每一扫描行分成若干切段。每一子图像矩阵或切段可用较低阶的多项式来逼近。例如,把一个扫描行分成若干切段,对于每一切段,用 n' 阶的 Lagrange 多项式 $\eta = F'_2(y)$ 和 $\xi = F'_1(x)$ 来逼近函数(2-5)和(2-9)而 $n' < n$, 使在 $n' + 1$ 点上,两组函数的值相等,在 $n' + 1$ 点外,误差的计算式为

$$R(y) = \left[F_2^{(n'+1)}(k) \prod_{i=j}^{j+n'} (y - y_i) \right] / (n' + 1)!.$$

其中 k 为该切段上 $n' + 1$ 点以外的任意点。当 $F_2^{(n'+1)}(k)$ 和 $\prod_{i=j}^{j+n'} (y - y_i)$ 同时达最大值时, $|R(y)|$ 最大。给定最大容许误差 ε , 可以从 $|\varepsilon| = |R(y)|$ 求出多项式降低到 n' 阶后,能取的最大切段长度

$$\Delta y_{\max} = \sqrt{\frac{2(n' + 1)! \cdot \varepsilon}{F_2^{(n'+1)}(k)}}.$$

三、实现图像几何变换的解算装置

根据上述算式,可以在计算机控制下用简单的数字解算装置来实现图像的几何变换。当图像以行扫描方式采样数字化并输入计算机时,用专用解算装置解方程式(2-5)以实现一个方向的几何变换。输入计算机中的图像数据经其它处理后旋转 90° 用行扫描方式输出记录时,用同样的解算装置实现另一方向的几何变换。这一方法也适用于对记录在计算机磁带上的图像数据进行几何变换。这时相当于用最近邻点法插值变换。如果增加一个简单的灰度插值装置来加密象元数目,将可按线性或三次褶积法插值。

1. 基本原理

当图像以行扫描方式数字化时,顺扫描线(y)方向的像元尺寸 Δl 和空间采样频率 ν 成反比。若空间采样频率 ν 随 y 的位置变化,则

$$\Delta l_s = l/\nu(y). \quad (3-1)$$

若图像在顺扫描线方向的某些地方要求扩大尺寸或纠正缩小畸变时,应提高空间采样频率 $\nu(y)$,使 Δl_s 变小;当某些地方要求缩小尺寸或纠正扩大畸变时,应降低空间采样频率 $\nu(y)$,使 Δl_s 相应地增大。因为记录时图像旋转了 90° ,像元顺扫描线采样的一边变成垂直于记录线一边。不管 Δl_s 的大小如何,垂直于记录线一边的像元尺寸是固定的大小 Δm ,因此可用下述方法进行 y 方向的几何变换。

图像数字化时有一个 y 方向空间频率固定为 $\nu_0 = Kl/\Delta m$ 的脉冲。有一个产生 y 方向采样脉冲 $\nu_s(y)$ 的数字解算装置。该装置在输入 y 个 ν_0 脉冲的过程中能严格地按(2-5)式产生 η 个顺扫描线方向的空间采样脉冲 ν_s ,则图像 $F(x, y)$ 上顺扫描线方向长 $y\Delta m$ 的图像,记录时将变成垂直于记录线方向长 $\eta\Delta m$ 的图像,因此完成了 $F(x, y)$ 到 $G(x, \eta)$ 或 $y \rightarrow \eta$ 的变换(见图 3(a), 3(b))。

图像另一方向的变换在记录过程中进行。由于图像旋转了 90° 变成了 $G_s(\eta, x)$,顺记录线方向的变量现在是 x 。若记录时的空间频率 ν 随 x 变化,则像元顺记录线方向的

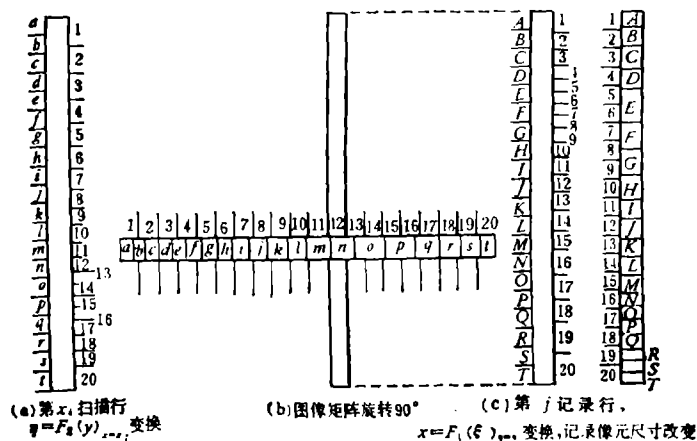


图 3 改变采样和记录空间频率来进行图像的几个变换

大小将为

$$\Delta l_r = l/u(x). \quad (3-2)$$

若图像在顺记录线方向的某些地方要求扩大尺寸或纠正缩小畸变时, 应降低记录空间频率 $u(x)$, 使 Δl_r 增大; 当某些地方要求缩小尺寸或纠正扩大畸变, 则应提高空间频率 $u(x)$, 使 Δl_r 相应地缩小。

扫描绘图机中有一个 x 方向空间频率固定为 $u_0 = Kl/\Delta m$ 的脉冲。有一个产生记录脉冲 $u_r(x)$ 的数字解算装置。该装置在输入 ξ 个 u_0 脉冲过程中能按(2-9)式严格地产生出 x 个 u_r 记录脉冲, 则 $G_2(\eta, x)$ 上 x 方向长度为 $x\Delta m$ 的图像将记录成长为 $\xi\Delta m$ 的图像, 因此可完成 $x \rightarrow \xi$ 或 $G_2(\eta, x)$ 到 $G_1(\eta, \xi)$ 的变换(图 3(c))。

2. 图像上下左右边位置的变换

几何变换式(2-5)中常数项 $B_{0i} = \sum_{j=0}^n b_{j0}x_j^i$, ($x_i = 1, 2, \dots, N$) 中的整数部分代表图像变换后上边缘的位置。图像数据通常以一扫描行为一记录段, 通过直接数据通道输入

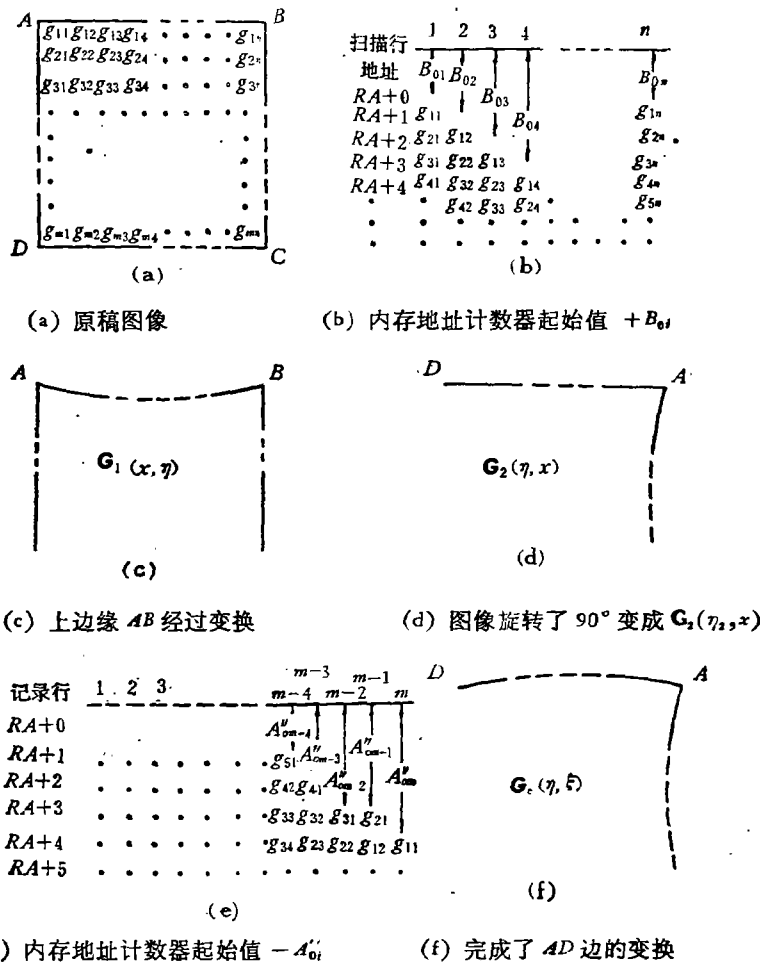


图 4. 图像上下边和左右边的变换

计算机内存中。为了实现上边缘的位置变换,只要在每一扫描行数据输入前,设置直接数据通道接口中的内存地址计数器的内容时,加上相应的 B_{0i} 值,使第 x_i 扫描行的第 1 个像元 g_{1i} 写入存储器第 x_i 记录段的第 B_{0i} 个字节;其余像元 g_{2i}, g_{3i}, \dots 顺序写入该记录段的 $B_{0i} + 1, B_{0i} + 2, \dots$ 地址(见图 4(a), 4(b))。该扫描行所有像元经过正确的变换后,下边缘将自动变换到应有的位置。

图像矩阵在输出记录前旋转了 90° , 因此输入时图像的左右边变成记录时的上下边。第 η_i 记录行变换式(2-9)中常数项 $A''_{0i} = \sum_{j=0}^n a''_{j0} \eta_i^j$ ($\eta_i = 1, 2, \dots, M$) 的整数部份将为记录时图像的上边缘位置。图像数据通常以一记录行为一记录段,通过直接数据通道输出记录。为了实现这一变换,只要在每一记录行数据输出前,设置的直接数据通道接口中的内存地址计数器的内容减去相应的 A''_{0i} 值就可以了。

3. 专用数字解算装置

当图像以行扫描方式输入输出计算机时,可以在输入、输出端用同样的解算装置来完成多项式(2-5)和(2-9)的解算。不同时输入输出时,可公用一个解算装置。解算装置的作用在于产生严格按多项式(2-5)或(2-9)变化的空间采样或记录频率。一般情况下,(2-5)和(2-9)式取 3 到 4 次多项式就能满足几何精度要求^[3,4]。如把一个扫描或记录行分成两到四个切段来处理,可以把多项式的阶数降低到二次。

图 5(a) 所示的结构是能解多项式的最简单的装置中的一种。在每一扫描行开始前,计算机把已经算好的该扫描行 m 个切段的 y 切段长度 l_{ys} , 对应的 η 切段长度 $l_{\eta s}$ 及每一切段的各次差分初始值 $\eta_s[0], \Delta\eta_s[0], \Delta^2\eta_s[0], \dots$ ($s = 1, \dots, m$) 送入解算装置中容量不大的存储器 RAM 中。其中

$$l_{ys} = y_{s+1} - y_s, l_{\eta s} = \eta_{s+1} - \eta_s. \quad (3-3)$$

在第 s 切段开始前,把 l_{ys} 和 $l_{\eta s}$ 分别送入寄存器 R_y 和 R_η , 并把各次差分初始值

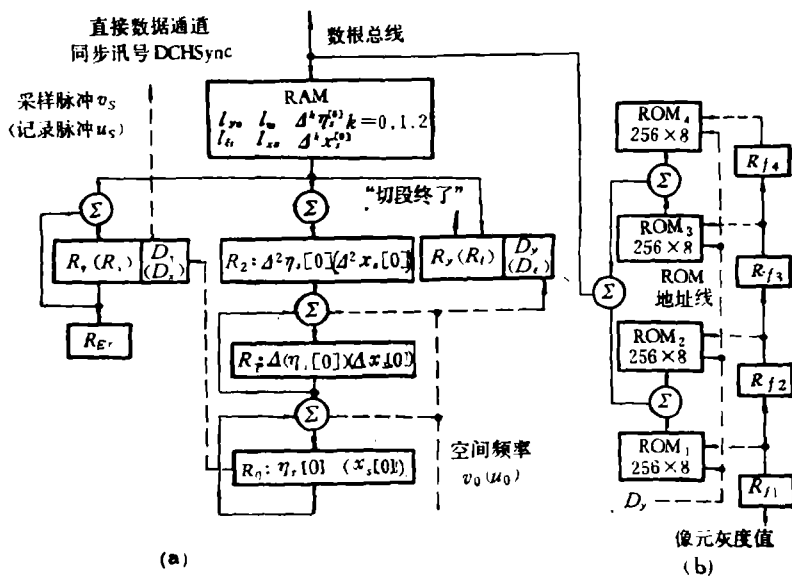


图 5 几何纠正解算装置及像元灰度插值图

$\Delta^k \eta_s[0]$ 分别送入寄存器 $R_k (k = 0, 1, \dots, n)$. 该切段第 i 个 v_0 脉冲将完成下列运算

$$\left. \begin{aligned} \eta_s[i] &= \eta_s[i-1] + \Delta \eta_s[i-1], \\ \Delta \eta_s[i] &= \Delta \eta_s[i-1] + \Delta^2 \eta_s[i-1], \\ &\dots \\ \Delta^{(n-1)} \eta_s[i] &= \Delta^{(n-1)} \eta_s[i-1] + \Delta^n \eta_s[0], \end{aligned} \right\} \quad (3-4)$$

进行 i 次迭代后, 即第 i 个 v_0 脉冲进入解算装置后, 从(3-4)式可得

$$\eta[i] = \eta[0] + i\Delta\eta[0] + \frac{i(i-1)}{2!} \Delta^2\eta[0] + \dots + \frac{i(i-1)\dots(i-n+1)}{n!} \Delta^n\eta[0]. \quad (3-5)$$

令(3-5)式中 i 各次项的系数和(2-5)式中 y 各对应次项的系数相等, 即可求出解(2-5)式所需的各次差分初始值 $\Delta^k \eta[0]$ 和多项式(2-5)的系数 $B_k (k = 0, 1, \dots, n)$ 之间的关系^[9], 对于 4 次以下的多项式

$$\left. \begin{aligned} \eta[0] &= B_{0s} \text{ 的小数部份} \\ \Delta\eta[0] &= B_{1s} + B_{2s} + B_{3s} + B_{4s} \\ \Delta^2\eta[0] &= 2B_{2s} + 6B_{3s} + 14B_{4s} \\ \Delta^3\eta[0] &= 6B_{3s} + 36B_{4s} \\ \Delta^4\eta[0] &= 24B_{4s} \end{aligned} \right\} \quad (3-5)$$

其中 B_{ks} 为第 s 切段逼近多项式 $\eta = \sum_{k=0}^n B_k y^k$ 中 y^k 项的系数.

在图 5(a) 所示的装置中, v_0 脉冲经分频器 D_y 分频 K_y 倍后使寄存器 R_y 的内容减 1, 而解算装置用(3-4)式迭代过程中由 R_0 溢出的脉冲经分频器 D_η 分频 K_η 倍后得到扫描采样脉冲 v_s . 由于置入解算装置的各次差分的初始值和逼近多项式的系数间是按自变量同次项系数相等的关系求得的, 因此, 当 $\Sigma(v_0/K_y) = y$ 时

$$\Sigma(V_s/K_\eta) = \eta. \quad (3-6)$$

对该切段的任一脉冲均成立, 因此可确切地完成(2-5)式要求的计算.

当 R_y 中的内容减到零时, 发出“切段终了”讯号. 该讯号把 RAM 中下一切段的 y 和 η 切段长度 l_{ys+1} 和 $l_{\eta s+1}$ 及各次差分初始值分别送入各有关寄存器中并开始下一切段的迭代运算. 上述过程从该扫描行第 1 切段 ($S = 1$) 开始, 到第 m 切段 ($S = m$) 为止, 下一扫描行重复这一过程, 直到整张图像扫描完毕, 完成图像 $F(x, y)$ 到 $G_1(x, \eta)$ 的变换为止.

在输出记录过程中, 可以用同样的装置来解算(2-9)式或其逼近多项式, 但 η, y, v_0, v_s 等分别用 x, ξ, u_0, u_s 作对应的替代.

加入分频器 D_y, D_η 及 D_ξ, D_x 并相应地提高空间频率 v_0 和 u_0 可得到 $\frac{d\eta}{dy}$ 和 $\frac{d\xi}{dx}$ 大于, 等于或小于 1 的变换; 使像元尺寸改变更近于连续而减少图像在几何变换过程中的讯息损失. 通常 K_y 和 K_ξ 可取常数如 4 分频.

由于计算机给出的各次差分初始值存在截断误差, 多项式次数愈高, 切段愈长因而迭

代次数愈多时,解算装置解得的 η 值(v_s 脉冲个数)的误差愈大。每一切段的几何变换式用二次多项式来逼近,各次差分初始值取二进制 32 位并采用 4 分频时, η 误差小于 ± 1

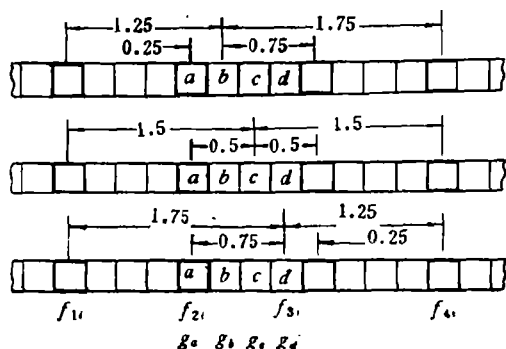


图 6 图像数据的加密插值

只要图像本身是行扫描方式数字化的并且一个扫描行是一个记录段,上述几何变换方法也适用于已经记录在计算机磁带上的图像数据。直接使用上述方法相当于几何变换时用最近邻点法插值。从磁带机到计算机接口间加一个简单的装置,可以实现 $\sin c$ 函数的近似插值。

图 5(b) 为这一插值装置的示意图。每隔 4 个 v_0 脉冲从磁带读出一个像元的灰度值并置入寄存器 R_{11} , 4 个相邻的像元灰度值保存在 4 个顺序传送的寄存器 R_{11} , R_{12} , R_{13} 和 R_{14} 中。第 1, 2, 3, 4 个 v_0 脉冲分别要求得到两个相邻像元 f_{2i} 和 f_{3i} 间的 4 个插值点 a_i , b_i , c_i , d_i 的灰度值 g_{ai} , g_{bi} , g_{ci} , g_{di} (见图 6)。用 $\sin c$ 函数插值时

$$\begin{aligned} g_{ai} &= f_{1i} \sin c(1) + f_{2i} \sin c(0) + f_{3i} \sin c(1) + f_{4i} \sin c(2) = f_{2i}, \\ g_{bi} &= f_{1i} \sin c(1.25) + f_{2i} \sin c(0.25) + f_{3i} \sin c(0.75) + f_{4i} \sin c(1.75), \\ g_{ci} &= f_{1i} \sin c(1.5) + f_{2i} \sin c(0.5) + f_{3i} \sin c(0.5) + f_{4i} \sin c(1.5), \\ g_{di} &= f_{1i} \sin c(1.75) + f_{2i} \sin c(0.75) + f_{3i} \sin c(0.25) + f_{4i} \sin c(1.25). \end{aligned} \quad (3-7)$$

对于陆地卫星计算机兼容磁带影像数据,像元有 64 个灰度等级,在 a_i , b_i , c_i 和 d_i 4 点上分别只有 4 个固定的 $\sin c$ 函数值,因此可以用硬件查表法来插值。在需要高速处理时可用 4 个 256×8 的 ROM, 而由 R_i 和 D_i 中的内容组合成地址, 从 4 个 ROM 中分别读出 4 个 $f_{ji} \sin c(x)$ 值, 按(3-7)式相加, 就能得到 a_i , b_i , c_i 和 d_i 点上的灰度插值 g_{ai} , g_{bi} , g_{ci} 和 g_{di} 。这时采样脉冲 v_s (或记录脉冲 u_r) 可取得相邻像元间以 $1/4$ 距离为单位的点上用 $\sin c$ 函数插值的灰度值。记录时可用同样的方法来实现灰度插值。

四、图像数据的向量和子矩阵表达方法的变换

在计算机图像处理中,图像数据经常用矩阵来表示。当图像以行扫描方式数字化时,每一扫描行对应于矩阵的一个列向量。

有时,为了计算机处理的方便,常常把矩阵 G 分成若干子矩阵

个像元的最大切段长度为 16384 个像元。在同样条件下,用三次多项式时最大切段长度为 925 个像元。为了避免累积误差,每一切段终了,将 R_i 中可能出现的迭代误差送到误差寄存器 R_{Er} 中,控制下一切段各次差分初始值从 RAM 传送到各相应寄存器时的舍入,在下一切段中进行补偿。 R_i 中的误差还和下一切段的 η 切段长度 $l_{i,s+1}$ 相加,使累积误差能检测出来并得到补偿。

4. 磁带图像数据的几何变换

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_{11} & \mathbf{G}_{12} & \cdots & \mathbf{G}_{1Q} \\ \mathbf{G}_{21} & \mathbf{G}_{22} & \cdots & \mathbf{G}_{2Q} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{G}_{P1} & \mathbf{G}_{P2} & \cdots & \mathbf{G}_{PQ} \end{bmatrix}.$$

若子矩阵 \mathbf{G}_{ij} 的维数相同且均为 $m \times n$, 则 $P = M/m$, $Q = N/n$ (见图 7(a), 7(b)).

要将外存中列向量形式的图像数据组织成子矩阵形式, 需有较大的计算机内存容量, 频繁的内外存交换并占据一定的计算时间。

图像数据和计算机之间一般通过直接数据通道来传送。利用一般小型计算机直接数据通道结构上的特点, 图像或记录在磁带上的影象数据以扫描行方式输入计算机过程中, 可自动组织成子矩阵; 在输出记录时, 子矩阵形式的图像数据可以同样自动转换成扫描行数据。

一般小型计算机的直接数据通道接口中, 通常有一个“内存地址计数器”MA。在计算机内存与外围设备交换数据前, 把要交换的数据在内存中的起始地址置入 MA 中。启动直接数据通道后, 每传送一个机器字, MA 中的内容加 1, 因此传送的数据在内存中的地址是顺序的。

为了输入内存中的 n 扫描行数据能自动组成 P 个 $m \times n$ 维子矩阵, 并设 $m = 2^{k_1}$, $n = 2^{k_2}$, 只要修改一下 MA 的结构, 使它由 MA1 和 MA2 两部份组成, 其中 MA2 占 $\log_2 n$ 位, 而每传送一个字, MA1 中的内容加 1 (见图 8(a))。当输入第 1 扫描行图像数据时 MA2 置零, 因此, 矩阵 \mathbf{G} 的第 1 列第 i 像元将写入第 $(i-1)n$ 存储单元; 输入第 2 扫描行数据时 MA2 置 1, 因此该列第 i 像元将写入第 $(i-1)n+1$ 存储单元。当各扫描行数据在内存中顺序排列时, 第 j 列第 i 像元应写入第 $(j-1)M + (i-1)$ 存储单元 ($j \leq n$), 但由于进行了上述修改, 将写入第 $(i-1)n + j - 1$ 存储单元 (见图 9)。当 n 个扫描行输入完毕后, 按常规的顺序写入外存。这时将能自动组成 P 个子矩阵 \mathbf{G}_{ij} ($i = 1, \dots, P$), 且子矩阵按行向量排列如图 10 所示。内存中开辟两个缓冲区, 每个缓冲区可放 n 扫描行数据, 两个缓冲区以交替方式工作。

子矩阵形式的图像数据可用同样方法在输出时恢复成记录行形式的数据。按 $0, n, \dots, (m-1)n$ 地址顺序输出的像元 $g_{11}, g_{21}, \dots, g_{M1}$ 正好构成第 1 记录行, 按 $k, n+k, \dots, (m-1)n+k$ 地址顺序输出的像元正好构成第 k 记录行数据。

子矩阵 \mathbf{G}_{ij} 的列数 n 还可以由计算机发出命令, 通过直接数据通道接口中增加的控制寄存器 K 和有关线路来控制 (见图 8(b)), 使 n 可以是小于 2^{k_2} 的任何整数。若计算机的内存有效工作单元为 MC 字节而图像的每一扫描行有 M 个像元, 则能取的最大 n 值为

$$n = MC/2M.$$

自动组成的子矩阵可用常规方法合并成较大的或分割成更小的子矩阵。

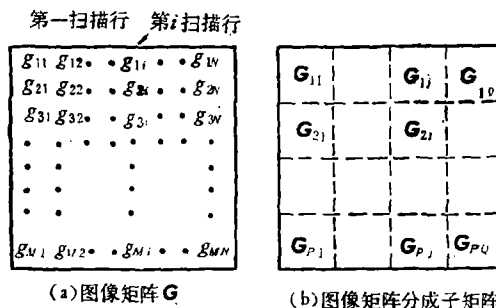


图 7 图像矩阵 \mathbf{G} 分成若干个子矩阵

这样组织图像数据时前面提出的几何变换方法仍然是合用的,只是与图像边缘 AB 或 DA 变换有关的位置值即 B_{0r} 和 A_{0r}' 的整数部份应加到直接数据通道接口中的内存地址计数器的 MA1 部份。

五、结 论

图像的几何变换或纠正在计算机图像处理中占的工作量往往很大。在陆地卫星影像的预处理系统或高性能的计算机图像处理系统中,通常用专门设计的阵列处理器来完成并且还需有较大容量的计算机内存。本文提出的方法适用于小型计算机图像处理系统中,因此能降低这一处理的代价。

在计算机图像处理中,常常要用子矩阵形式来表示图像数据。稍稍改动计算机的直接数据通道接口,就能自动进行扫描行向量和子矩阵数据形式的相互变换,随着遥感和其它计算机图像处理的理论和技术的发展,计算机图像处理的工作量将越来越大。采用软硬件相结合的方法来解决其中一些工作量较大的任务,将是计算机图像处理系统的发展途径之一。

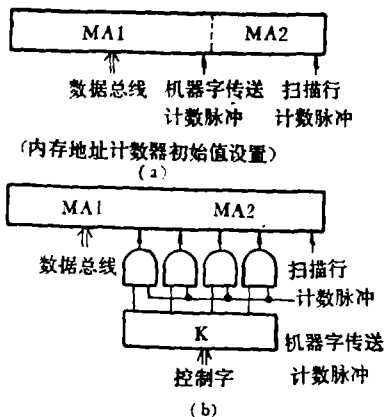


图 8 内存地址计数器 MA 的修改

通常的次序	修改后次序	像元	
0	0	g_{11}	G_{11} 第 1 列
1	n	g_{21}	
2	$2n$	g_{31}	
.	.	.	
$m-1$	$(m-1)n$	g_{m1}	
m	mn	$g_{m+1,1}$	G_{21} 第 1 列
$m+1$	$(m+1)n$	$g_{m+2,1}$	
.	.	.	
$2m-1$	$(2m-1)n$	$g_{2m,1}$	
\vdots	\vdots	\vdots	
$M-m$	$(M-m)n$	$g_{M-m+1,1}$	G_{P1} 第 1 列
$M-m+1$	$(M-m+1)n$	$g_{M-m+2,1}$	
.	.	.	
$M-1$	$(M-1)n$	$g_{M,1}$	
M	1	g_{12}	G_{11} 第 2 列
$M+1$	$n+1$	g_{22}	
.	.	.	
$M+m-1$	$(m-1)n+1$	g_{m2}	
\vdots	\vdots	\vdots	
$2M-m$	$(M-m)n+1$	$g_{M-m+1,2}$	G_{P1} 第 2 列
$2M-m+1$	$(M-m+1)n+1$	$g_{M-m+2,2}$	
.	.	.	
$2M-1$	$(M-1)n+1$	g_{M2}	
\vdots	\vdots	\vdots	
$(n-1)M$	$n-1$	g_{1n}	G_{11} 第 n 列
$(n-1)M+1$	$2n-1$	g_{2n}	
.	.	.	
$(n-1)M+m-1$	$mn-1$	g_{mn}	
\vdots	\vdots	\vdots	
$nM-m$	$(M-m)n+n-1$	$g_{M-m+1,n}$	G_{P1} 第 n 列
$nM-m+1$	$(M-m)n+2n-1$	$g_{M-m+2,n}$	
.	.	.	
$nM-1$	$nM-1$	g_{Mn}	
\vdots	\vdots	\vdots	

图 9 “内存地址计数器”修改后像元在内存中的排列

g_{11}	g_{12}	g_{1n}
g_{21}	g_{22}	G_{11}	g_{2n}
.	.	.	.
g_{m1}	g_{m2}	g_{mn}
$g_{m+1,1}$	$g_{m+1,2}$	$g_{m+1,n}$
$g_{m+2,1}$	$g_{m+2,2}$	G_{21}	$g_{m+2,n}$
.	.	.	.
$g_{2m,1}$	$g_{2m,2}$	$g_{2m,n}$
.	.	.	.
.	.	.	.
$g_{M-m+1,1}$	$g_{M-m+1,2}$...	$g_{M-m+1,n}$
$g_{M-m+2,1}$	$g_{M-m+2,2}$	G_{P1}	$g_{M-m+2,n}$
.	.	.	.
$g_{M,1}$	g_{M2}	...	g_{Mn}

图 10 像元在子矩阵中的排列

参 考 文 献

- [1] Yang Shih-zen, The Digital Scanning and Writing System in the Institute of Geography, Chinese Academy of Sciences. *Joint Symposium: Geographic Perspectives on the Environment: American and Chinese Views*, OCT. (1978).
- [2] Yang Shih-zen, Geometric Transformation and Data Organization in Digital Image Processing. *Joint Symposium: Geographic Perspectives on the Environment: American and Chinese Views*, OCT. (1978).
- [3] Paul E. Anuta, Geometric Correction of ERTS-1 Digital Multispectral Data, *LAES Information Note 103073 Purdue University*, (1973).
- [4] 村井俊治, 前田紘: 地球资源卫星 MSS テータの几何学的补正に関する研究. 东京大学生产技术研究报告 第 27 卷第 5 号(昭和)53 年 11 月.
- [5] А. А. Воронов, А. Р. Гарбузов, Ян Си-зен и др. Цифровые аналоги для систем автоматического управления. *Издательство АН СССР, Москва* (1960).
- [6] A. Klinger, M. L. Rhodes, V. T. To, *Accessing Image Data, Pattern Recognition and Image Processing* (1976).

GEOMETRIC TRANSFORMATION AND DATA ORGANIZATION IN DIGITAL IMAGE PROCESSING

YANG SHI-BEN

(Institute of Remote Sensing Applications, Space Science and Technology Center, Academia Sinica)

ABSTRACT

In this paper the algorithms for geometric transformation of image data and the decomposition of the image matrix into submatrices are introduced. When the image is digitized in the line scanning mode, the geometric transformation of the two dimensional image matrix can be realized by two one-dimensional geometric transformations, i.e., first carrying out the geometric transformation along the scanning lines and then rotating the image matrix through 90 degrees and carrying out the geometric transformation in the other direction. These algorithms can be implemented in a minicomputer system with an appropriate software and a special-purpose hardwired device, so that it can reduce the computing time effectively and is adaptable in minicomputer digital image processing systems.