

一种计算图象处理速度的新方法

孔俊宝

(南京邮电学院 南京 210003)

刘梦周

(北方交通大学 北京 100044)

摘要 本文描述的是一种用于计算图象处理速度的新技术。如果图象面积已知,那么利用一组相互正交的延迟线就可以分析计算出图象处理速度矢量的两个垂直分量;而计算一个未知图象面积的处理速度,则需要以两组相互正交的延迟线辅助之。该方法适用于作处理模拟大规模集成电路的工具。

关键词 图象处理,处理速度,计算方法

1 引言

在很多情况下,数字图象处理一般需要大量的高度专门化的高速硬件。用常规的数字信号处理技术来计算图象速度,即使是采用固态高速数字硬件,其计算过程也是很慢的。传统的数字信号处理技术两帧图象的数字化信息,在作处理以前必须在存储器中至少存储一帧的数字化数据。这就要求使用高速的 A/D 转换器和存储器。此外,由于难以做到将像素和 A/D 转换器集成在一起,就需要连续地对图象进行数字化。这就严重制约了数字处理器的并行处理能力,在大多数情况下,导致了动态图象的失真。专门化的数字硬件设备用于完成计算光通量的算法虽然是高度精练的^[1],但是用来达到实现计算图象处理高速度的目的确是困难的,在很多情况下是不可能的。

使用模拟超大规模电路系统的聚焦平面图象处理方法被证明是一种行之有效的方法。尽管具有内聚焦平面图象处理能力的成象器的像素密度并不比传统 CCD 成象器的像素密度高(以下我们简称前者为精巧成象器),但其所完成的预处理功能却能大大减低后处理器的计算工作量。在某些情况下,精巧成象器可单独工作。而精巧成象器的这种可能性就使其能应用于某些没有多余计算能力和没有多余内存资源的新领域。

模拟超大规模集成电路用于聚焦平面图象处理速度计算,这早已有所论述,它能产生一个动态图象的速度空间图。但其每个像素所需的晶体管数量超过 60 个,而且,像素的密度也不高,此外,还需要电压多路器,而此多路器在没有大输入偏压的 CMOS 电路中是

1993-08-09 收到,1994-06-30 定稿

孔俊宝 男,1929年生,教授,计算机与通信电子电路系统的研究。

刘梦周 男,1963年生,硕士,计算机与通信专业。

难以正常工作的。

本文介绍的是一种新颖的图象处理速度计算方法,该方法适合于与精巧成象器结合在一起使用,所描述的详细分析可作为使用 CMOS 工艺的硅成象的基础。

2 以延时为基础的图象处理速度计算

这里推荐一种新的图象速度的计算方法,该方法是基于在并联延迟线上传传的多路信号。一个延迟线包含各个延迟因素的串联连接和在线信号的快速传输,由于在延迟线中控制的信号快速传输,该延迟因素必须全部正确地控制延迟。理想地,全部信号应该同时离开延迟线的一端并立即达到延迟线的另一端。

当一个动态图象设计成一个交叉延迟线上时(如图 1 所示),一些信号将勾划该图象边缘线和进入图象边缘线范围以内。信号在图象边缘线内延迟时间的长度依赖于该图象信号的相对速度。如果只考虑延迟线的 x 方向情况,则全部信号在图象边缘线内停留的时间长度总和,即总的时间为:

$$T_{1x} = \frac{A_{IM}}{n(V_s - V_x)}, \quad V_s > V_x, \quad (1)$$

上式中 A_{IM} 为图象的面积, n 是两个象素之间的距离, V_s 为信号速度, V_x 为图象速度 V_{IM} 的 x 分量,由于 T_{1x} 不是 V_y (图象速度 V_{IM} 的 y 分量)的函数,如果 T_{1x} 能被测量并且全部参数 (A_{IM} , n 和 V_s) 是已知的,则 V_{IM} 的 x 分量是可以计算出来的。

同样,在延迟线的 y 方向全部信号在图象边缘线内,延迟线上传传的时间长度总和为

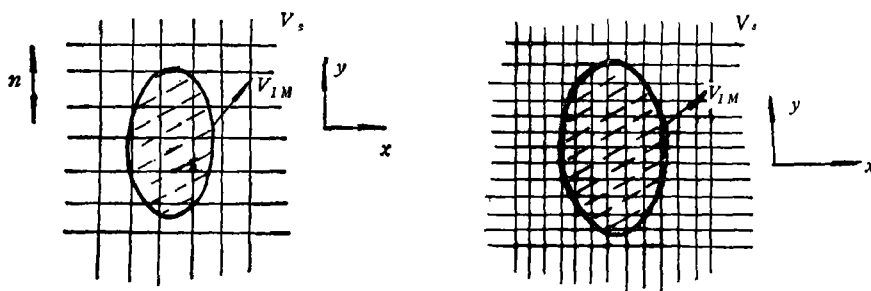


图 1 在交叉延迟线上的动态图象

图 2 使用两个交叉延迟线计算图象速度

$$T_{1y} = \frac{A_{IM}}{n(V_s - V_y)}, \quad V_s > V_y. \quad (2)$$

这样,如果 T_{1x} 和 T_{1y} 能够被测量,图象速度 V_{IM} 即可算出。

使用(1)和(2)式去计算 V_{IM} 需要知道 A_{IM} , 如果 A_{IM} 不知道,就需要知道交叉延迟线的两种情况:一种包含延迟线上的 x 方向和 y 方向的情况;而另一种则包含延迟线上负 x 方向和负 y 方向的情况,如图 2 所示。

T_{1xN} 在延迟线上负 x 和负 y 方向,分别表示为

$$T_{1xN} = \frac{A_{IM}}{n(V_s + V_x)}, \quad (3)$$

$$T_{1yN} = \frac{A_{IM}}{n(V_x + V_y)}, \quad (4)$$

上式中下脚 xN 和 yN 分别表示延迟线上的负 x 和负 y 方向。

结合(1)至(4)式,可得结果:

$$V_x = \frac{(T_{1x} - T_{1xN})}{(T_{1x} + T_{1xN})} V_s, \quad (5)$$

$$V_y = \frac{(T_{1y} - T_{1yN})}{(T_{1y} + T_{1yN})} V_s. \quad (6)$$

这样,对于计算 V_x 和 V_y , 只需知道 $T_{1x}, T_{1xN}, T_{1y}, T_{1yN}$ 和 V_s 即可。

3 图象面积的量化和计算误差

我们分析至此,前曾假设 T_{1y} 是 A_{IM} 的连续函数,但是图象是有限的,并且用成象器看图象面积是一个实际图象面积的量化值,该图象面积的量化值用 A_{IMg} 来表示。

如果仅使用交叉延迟线的一种情况,则计算图象面积的量化误差 $E_r(V_x)$ 或 $E_r(V_y)$ 为

$$E_r(V_x) = \left(\frac{V_x}{V_s} - 1 \right) \left(1 - \frac{A_{IMg}}{A_{IM}} \right). \quad (7)$$

上式表示当 $A_{IMg} \approx A_{IM}$, 且 V_x 较低时, $E_r(V_x)$ 是大的,则所计算的速度是慢动作的图象速度。若用一个很高象素密度的成象器,则 $A_{IMg} \approx A_{IM}$, 或者信号速度 V_s 置于一个低值,其结果是加长了计算周期。

如果使用交叉延迟线的两种情况,从延迟线的 x 方向来看量化图象面积不等于从延迟线负 x 方向所看的结果。令 A_{IMg}^+ 和 A_{IMg}^- 分别表示从延迟线 x 方向和负 x 方向所看的量化图象面积。在这种情况下计算误差 $E_r(V_x)$ 为

$$E_r(V_x) = \frac{[1 - (A_{IMg}^-/A_{IMg}^+)] [1 - (V_x/V_s)^2]}{[1 - (A_{IMg}^-/A_{IMg}^+)] + [1 + (A_{IMg}^-/A_{IMg}^+)] [V_x/V_s]}. \quad (8)$$

当 $A_{IMg}^- = A_{IMg}^+$ 时, $E_r(V_x)$ 则为零。(7)式与(8)式比较,可得出下面的结论。当使用交叉延迟线的一种情况时,则该计算误差仅是图象面积量化的失配函数。

量化图象面积表示为

$$A_{IMg} = \left(1 - \frac{V_x}{V_s} \right) \sum_{A_{IM}} \left[n^2 Q \left(\frac{L_j^i}{n \{ (V_y/V_x)^2 + [1 - (V_x/V_s)]^2 \}^{1/2}} \right) \right], \quad (9)$$

上式中的 L_j^i 是图象覆盖的第 i 个延迟线的长度,算子 $Q(x)$ 是已给定 x 的量化值, $\sum_{A_{IM}}$ 是图象的总的交叉延迟线。

图象面积的量化误差是图象方位的函数。当图象的形成是不规则的或者 $V_x \approx 0$ 时,如不使用数值方法计算量化图象面积是很困难的,但估算图象面积的量化误差的极限是可能的。任意形状的一个图形,如图3所示。当图象在垂直平面上投影时,形成一条线,用 L_M 和 L_m 分别表示其投影线的最长和最短;用 θ_M 和 θ_m 分别表示最长线和最短线与 y 轴之间的夹角。假定每一延迟线产生一个(单位面积 n^2) 图象面积量化误差的最坏

情况为 $A_{IM}, E_q(A_{IMq}) = (A_{IM} - A_{IMq})/A_{IM}$, 位于下式的极限, 即

$$\frac{L_m n}{A_{IM} \cos \theta_m} < E_q(A_{IMq}) < \frac{L_M n}{A_{IM} \cos \theta_M}. \quad (10)$$

在指定的情况下, 如一个矩形图象, 其尺寸为 $l \times b$, 边缘是与 x 和 y 轴平行, 如图 4 所示. 该图象的速度 $V_{IM} = V_x$ 和 $V_y = 0$, 在这种情况下, 图象的视在宽度是 b 和图象的视在长度是 $l/(1 - V_x/V_s)$, 其量化视在图象面积为

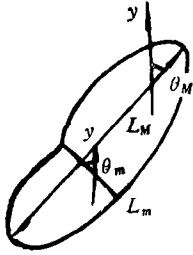


图 3 任意形状的图形

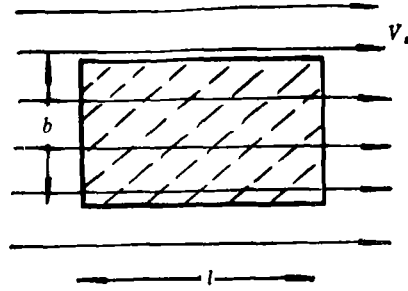


图 4 矩形图形

$$A_{IMq} = n^2 Q \left[\frac{b}{n} \right] Q \left\{ \frac{1}{n[1 - (V_x/V_s)]} \right\}; \quad (11)$$

而量化图象面积则为

$$A_{IMq} = n^2 \left(1 - \frac{V_x}{V_s} \right) Q \left[\frac{b}{n} \right] Q \left\{ \frac{1}{n[1 - (V_x/V_s)]} \right\}. \quad (12)$$

如果 $V_y \approx 0$, 则(10)式可用来估算图象面积量化误差的极限. 在这样的特定情况下, (10)式可写成

$$\frac{n}{l} < E_q(A_{IMq}) < \frac{n}{l} \left[1 + \left(\frac{l}{b} \right)^2 \right]. \quad (13)$$

结合(7)式和(13)式, 得到

$$\left(\frac{V_s}{V_x} - 1 \right) \frac{n}{l} < E_s(V_x) < \left(\frac{V_s}{V_x} - 1 \right) \frac{n}{l} \left[1 + \left(\frac{l}{b} \right)^2 \right]. \quad (14)$$

这样, 如果有一个象素为高密度的, 并选择一个适当的 V_s 值, 则得到图象面积量化的计算误差将是低的.

4 用光强调制延迟线测量 T_1

这里是几种测量 T_1 的有效方法. 不同的象素密度就意味着有不同的测量精度. 本文描述的这个方法是用落在象素上的光强去选择延迟因素的延时. 该延迟因素的延时用 τ_1 表示, 这是图象边界内的; 而图象边界外的延时用 τ_0 表示 ($\tau_1 \gg \tau_0$).

如果图象沿着 x 方向移动的速度为 V_x , 则延迟线的总延时 T_2 可表示为

$$T_{2x} = \frac{A_T}{n^2/\tau_0} + \left(1 - \frac{\tau_0}{\tau_1} \right) \frac{A_{IM}}{(n^2/\tau_1)[1 - (V_x \tau_1/n)]}, \quad (15)$$

上式中 A_T 表示沿着延迟线的方向的图象, 并属于 x 分量的总和.

同样

$$T_{2y} = \frac{A_T}{n^2/\tau_0} + \left(1 - \frac{\tau_0}{\tau_1}\right) \frac{A_{IM}}{(n^2/\tau_1)[1 - (V_x\tau_1/n)]} \quad (16)$$

如果使用交叉延迟线的两个方位, V_x 和 V_y 可以表示为

$$V_x = \frac{(T_{2x} - T_{2xN})(n/\tau_1)}{T_{2x} + T_{2xN} - 2A_T/(n^2/\tau_0)} \quad (17)$$

和

$$V_y = \frac{(T_{2y} - T_{2yN})(n/\tau_1)}{T_{2y} + T_{2yN} - 2A_T/(n^2/\tau_0)}, \quad (18)$$

上式中约定 N 是属于 T_2 , 表示 x 或 y 的负方向的延迟线。

使用光强调制测量延迟线的 T_1 主要不利的地方是图象边界延迟线的“弯曲”, 这是由于图象边界内外的传播速度的不同而引起。这种情况类似于在空气与玻璃之间界面上的光交叉, 如果入射角不等于全部信号, 正方图象在成象器上运动的情况如图 5 所示。在延迟线的 x 方向的总延时可表示为

$$T_{2x} = \frac{A_T}{n^2/\tau_0} + \left(1 - \frac{\tau_0}{\tau_1}\right) \cdot \frac{A_{IM}}{(n^2/\tau_1)[1 - (V_x\tau_1/n)]} + \varepsilon_T, \quad (19)$$

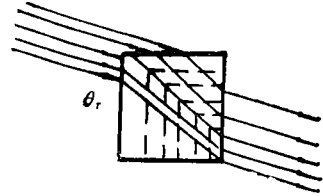


图 5 正方图象的折射

上式中的 ε_T 是下面给出的误差项:

$$\varepsilon_T = -\left(1 - \frac{\tau_0}{\tau_1}\right) \frac{A_{IM}[(V_y\tau_1/n)]}{2(n^2/\tau_1)[1 - (V_x\tau_0/n)][1 - (V_x\tau_1/n)]^2} \quad (20)$$

假定折射角 θ , 在 45° 与 90° 之间, 即 $45^\circ < \theta < 90^\circ$ 。

忽略 ε_T 后 V_x 的近似值为 V_{xc} , 则

$$V_{xc} = \frac{n}{\tau_1} \left(1 - \frac{[1 - (\tau_0/\tau_1)]A_{IM}}{(n^2/\tau_1)\{T_{2x} - A_T/(n^2/\tau_0)\}}\right) \approx V_{xc} \quad (21)$$

如果 $V_x, V_y \ll n/\tau_1$, 由于反射信号而产生的 V_{xc} 的误差与 V_y/V_x 的关系较强, 而与 V_x 的绝对值的关系较弱。例如, 当 $n = 127.5\mu\text{m}$, $A_T = 14.44 \times 10^6\mu\text{m}^2$, $A_{IM} = 1 \times 10^6\mu\text{m}^2$, $n/\tau_0 = 12.75 \times 10^6\mu\text{m/s}$ 和 $n/\tau_1 = 25.56 \times 10^3\mu\text{m/s}$, V_{xc} 的误差与 V_y/V_x 和 V_x 成函数关系。从该特性可看出, V_{xc} 的误差近似等于 $V_y/(2V_x)$ 。如果 V_x 近似为 n/τ_1 , 则 V_{xc} 的误差变得比 $V_y/(2V_x)$ 更为重要。

实际上 V_{xc} 和 V_{yc} , 其误差校正基于下面的方程

$$V_{xcc} = \frac{4}{3} V_{xc} \left(1 + \frac{V_{yc}}{2V_{xc}}\right), \quad (22)$$

$$V_{ycc} = \frac{4}{3} V_{yc} \left(1 + \frac{V_{xc}}{2V_{yc}}\right). \quad (23)$$

上两式中的 V_{xcc} 和 V_{ycc} 分别为 V_{xc} 和 V_{yc} 的校正值。误差校正之后, V_x 的误差 V_{xc} 和 V_y 的误差 V_{yc} 与 V_y/V_x 均呈函数关系。在校正误差之后, V_x 的误差 V_{xc} 从 100% 降到小于 7%。这里要注意的是, V_x 的误差 V_{xc} 是随着 V_y/V_x 的增大而增大; 同样,

V_{ye} 也随着 V_x/V_y 的增大而增大;或者相反地, V_{ye} 是随着 V_y/V_x 的增大而减小。如果 V_y/V_x 大于 0.5 以上时,则 V_{ye} 就小于 5%。

如果使用延迟线,则 V_{xe} 和 V_{ye} 分别为

$$V_{xe} = \frac{(T_{2x} - T_{2xN})(n/\tau_1)}{T_{2x} + T_{2xN} - 2A_T/(n^2/\tau_0)}, \quad (24)$$

$$V_{ye} = \frac{(T_{2y} - T_{2yN})(n/\tau_1)}{T_{2y} + T_{2yN} - 2A_T/(n^2/\tau_0)}. \quad (25)$$

当 $V_x\tau_1/n, V_y\tau_1/n \ll 1$ 时,由于信号反射所形成的 V_{xe} 是比较小的。如果 V_x 和 V_y 小于 $10^3\mu\text{m/s}$,则 V_{xe} 和 V_{ye} 就小于 5%, V_x 和 V_y 值加大时, V_{xe} 和 V_{ye} 可加大到 25% (在 $V_x = 5 \times 10^3\mu\text{m/s}$ 和 $V_y = 10^4\mu\text{m/s}$)。误差校正处理可减小误差,因此具有重要的意义。

如果 $(V_x\tau_1/n)^2 \ll 1, V_x$ 与 V_{xe} 之间的关系可近似为

$$V_{xe} \approx 2V_x \left[\frac{1 - (V_y\tau_1/n)}{2 - (V_y\tau_1/n)} \right]. \quad (26)$$

用 $V_{x'}$ 表示 V_{xe} 的校正值,可用下式表达为

$$V_{x'c} = \frac{V_{xe}[2 - (V_y\tau_1/n)]}{2[1 - (V_y\tau_1/n)]}. \quad (27)$$

(27)式是用 $V_{x'c}$ 和 $V_{y'c}$ 分别代替(26)式中的 V_x 和 V_y 而获得。同样地

$$V_{y'c} = \frac{V_{ye}[2 - (V_x\tau_1/n)]}{2[1 - (V_x\tau_1/n)]}. \quad (28)$$

用(27)式在 $V_x = 5 \times 10^3\mu\text{m/s}$ 和 $V_y = 10^4\mu\text{m/s}$ 的情况下,可将误差从 24%减小到小于 5%。

5 在延迟线失配下测量速度精确度

如果各延迟线上有不同的信号速度,则图象在各延迟线上的视在长度 L'_i 可表示为

$$L'_i = \frac{L_i}{1 - (V_x\tau_{ie}/n)}, \quad (29)$$

上式中 τ_{ie} 是延迟线中的延迟因素的平均时延。

使用(29)式可将图象的视在面积表达为

$$A'_{iM} = \frac{A_{iM}}{1 - (V_x\tau_{ie}/n)} + n \sum_{A_{iM}} (\varepsilon_d), \quad (30)$$

上式中

$$\varepsilon_d = L_i \left\{ \frac{(V_x/n)(\tau'_{ie} - \tau_1)}{[1 - (V_x\tau_{ie}/n)][1 - (V_x\tau'_{ie}/n)]} \right\}. \quad (31)$$

上式中 τ'_{ie} 是在图象面积 A_{iM} 内侧第 i 个延迟线上延迟因素的平均速度。

对(31)式的近似研究表明,在延迟线中失配成分所产生的 A'_{iM} 的误差正比于 L_i ($\tau'_{ie} - \tau_1$)。如果处理参数是较高随机变化的,图象上的延迟线较长, τ_1 是统计意义的 τ'_{ie} 的近似值,则可用 $(\tau'_{ie} - \tau_1)$ 的减小来补偿 L_i 的增大。但是,如果处理参数是有次

序的变化, 一个长 d_i 的延迟线可以导致 A'_{IM} 产生误差。简言之, 在这里建议使用成象器以完成高精度的速度计算。对于有次序变化的处理参数, 可能减少许多误差。

用(30)式计算速度的表达式为

$$V_s = \frac{(T_{2x} - T_{2xN})(n/\tau_1)}{T_{2x} + T_{2xN} - 2A_T/(n^2/\tau_0)} + \epsilon_v, \quad (32)$$

上式中的误差项 ϵ_v 为

$$\epsilon_v = \frac{-2[1 - (\tau_0/\tau_1)] \sum_{A_{IM}} (\epsilon_d)}{T_{2x} + T_{2xN} - 2A_T/(n^2/\tau_0)}. \quad (33)$$

如果 $\tau_0 \ll \tau_1, V_s \tau_1/n \ll 1$ 并且 $\tau'_{ie} = \tau_{ie}$ 为 j 的全部值, 则(33)式可简化为

$$\epsilon_v = V_s \left(1 - \frac{\tau_{ie}}{\tau_1}\right). \quad (34)$$

这样, 该速度计算误差的百分数等于延迟线的平均延时的不匹配百分数。

6 结 束 语

本文描述了一种计算图象处理速度的新方法, 按照它的限度进行了详细的分析, 使用通常有效的方法分析, 最大的计算误差源是延迟线的不匹配。本文的分析中关于成象器的硅特性和内在的图象处理速度的计算容量, 将另文描述。

参 考 文 献

- [1] Mead C A. Analog VLSI and Neural Systems. New York, MA:Reading. Addison-Wesley, 1989, Ch.3.
- [2] Kong J B. On Data Network Accurate Measurement of Group-Delay by Microprocessor. ICC, Beijing: 1986, 613-618.
- [3] 孔俊宝. 电信科技情报, 1982, (2): 29.

A NEW METHOD FOR IMAGE-VELOCITY COMPUTATION

Kong Junbao

(Nanjing Institute of Posts and Telecommunications, Nanjing 210003)

Liu Mengzhou

(Northern Jiaotong University, Beijing 100044)

Abstract A new method for image-velocity computation is described. If the image area is known, then by using a set of intersecting delay lines, the two perpendicular components of image velocity can be analysed and computed, but if the image area is not known, then two sets of intersecting delay lines are required. This method is a suitable tool for simulating very large-scale integrated circuits.

Key words Image process, Processing velocity, Method for computation