[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06T 3/40 (2006.01)

[12] 发明专利申请公布说明书

[21] 申请号 200810051097.3

[43] 公开日 2009 年 1 月 28 日

[11] 公开号 CN 101354782A

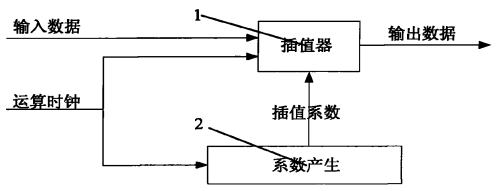
[22] 申请日 2008.8.21
[21] 申请号 200810051097.3
[71] 申请人 中国科学院长春光学精密机械与物理研究所
地址 130033 吉林省长春市东南湖大路 16 号
[72] 发明人 王德江 匡海鹏
[74] 专利代理机构 长春菁华专利商标代理事务所
代理人 王立伟
权利要求书 1 页 说明书 5 页 附图 1 页

[54] 发明名称

一种节省硬件资源的数字图像定标器

[57] 摘要

本发明一种节省硬件资源的数字图像定标器。属于视频图像处理领域。该图像定标器包括插值器模块和系数产生模块两部分。具体连接方式为：原始图像数据与时钟信号输入到插值器模块中；在外部时钟的作用下，系数产生模块生成插值运算所需的位置系数，该系数亦输入到插值器模块中；插值器模块根据输入的原始图像数据与插值位置系数进行插值运算；然后插值器模块输出经插值运算处理后的图像。该种定标器基于对缩放算法的分层次因式分解，显著降低了乘法器数量，所需硬件资源仅为一个累加器、寄存器与相应的辅助控制单元，且适用于任意比例的缩放。本发明采用多阶插值算法的图像定标器，可以完成任意比例的图像缩放功能。



1、一种节省硬件资源的数字图像定标器，其特征在于该图像定标器包括插值器模块（1）和位置系数产生模块（2）两部分；

具体连接方式为：原始图像数据与时钟信号输入到插值器模块（1）中；在外部时钟的作用下，位置系数产生模块（2）生成插值运算所需的位置系数，亦输入到插值器模块（1）中；插值器模块（1）根据输入的原始图像数据与插值位置系数进行插值运算；然后插值器模块输出经插值运算处理后的图像。

2、根据权利要求 1 所述的一种节省硬件资源的数字图像定标器，其特征在于所述的插值器模块（1）根据输入的原始图像数据与插值位置系数，采取合并同类项进行插值运算；插值器模块（1）包括负责对输入的数据进行暂存的寄存器（3）、（4）、（5）、（6），暂存的数据分别为 g_0, g_1, g_2, g_3；负责完成插值算法中的 8 次加、减法运算的加法器（7）、（8）、（9）、（10）、（11）、（12）、（13）、（14）；负责完成涉及到 μ_k 的两次乘法运算的乘法器（17）、（18）；负责完成两次固定系数乘法运算的移位相加器（15）、（16）；执行的顺序如图中所示，控制运算时序的寄存器（3）、（4）、（5）、（6）。

3、根据权利要求 1 所述的一种节省硬件资源的数字图像定标器，其特征在于所述的位置系数产生模块（2）包括加法器（19）、寄存器（20），设输入行含 X 个像素点，输出行含 Y 个像素点，Y/X 表示缩放比。

一种节省硬件资源的数字图像定标器

技术领域：

本发明属于视频图像处理领域，提出了一种显著节省硬件资源的数字图像定标器。

背景技术：

平板显示器在显示终端设备中逐渐占据主导地位，但其最佳分辨率为固定值，即为其物理显示像素点，不同的显示设备其物理显示像素点也不尽相同；同时数字电视或计算机的输入图像分辨率也是可变的，这就需要图像定标引擎将不同分辨率的输入图像映射到平板显示器上。

为了实现实时处理，传统的数字图像定标器通常采用两点插值算法。该方法通过计算插值点在原图像水平或垂直方向上两点的插值位置，得到该两点的权重，从而计算出插值点的图像灰度值。这种方法虽然实现起来非常简单，在 FPGA(Field Programmable Gate Array)或 ASIC(Application Specific Integrate Circuit)实现中一般需要两个乘法器与一个加法器，但缩放后的图像质量相对于原图像有较大的下降。如果采用高阶的插值算法，虽然能提高缩放后图像的质量，但极大的增加了硬件资源，降低了系统实时处理的能力。随着高分辨率视频资

源的推广普及，迫切的需要一种即能保证缩放后图像质量，又能以少的硬件资源实现的定标器。

发明内容：

为了降低采用高阶插值算法的图像定标器所需的硬件资源，增强图像定标器的实时处理能力，本发明提出了一种适用于线性多阶插值算法的数字图像定标器。如图 1 所示，该定标器主要由两部分构成：插值器模块和系数产生模块。具体连接方式为：原始图像数据与时钟信号输入到插值器模块中；在外部时钟的作用下，系数产生模块生成插值运算所需的位置系数，该系数亦输入到插值器模块中；插值器模块根据输入的原始图像数据与插值位置系数进行插值运算；然后插值器模块输出经插值运算处理后的图像。

插值器是定标器的核心，根据位置系数产生模块生成的被插值点位置信息，按照预先设计的算法与实现结构，完成相应的插值运算，得到缩放后图像的灰度值。与传统插值器不同，本插值器对插值算法采取了合并同类项处理，例如采用四点线性插值算法的定标器，g_0,g_1,g_2,g_3 为原图像中连续的四个点，该四个点可以为水平方向或垂直方向的连续一组；C_0,C_1,C_2,C_3 为该四个点的插值权重系数；μ_k 为位置系数(插值点 g 位于 g_1,g_2 之间，μ_k 为 g 相对于 g_1 的位置)，g 为插值后得到的新点。则四点线性插值算法如式 1 所示：

$$
\begin{aligned}
C_0 &= 0.5\times\mu_k{}^2 - 0.5\times\mu_k \\
C_1 &= -0.5\times\mu_k{}^2 + 1.5\times\mu_k \\
C_2 &= -0.5\times\mu_k{}^2 - 0.5\times\mu_k + 1 \\
C_3 &= 0.5\times\mu_k{}^2 - 0.5\times\mu_k \\
g &= g_0\times C_0 + g_1\times C_1 + g_2\times C_2 + g_3\times C_3
\end{aligned} \qquad (1)
$$

从上式中观察到，最终的插值点灰度值为g_0,g_1,g_2,g_3与相应的权重系数之积，完成一次插值运算需要 20 次乘法(8 次为固定系数乘法)和 7 次加法；另每一个权重系数都有相同的项μ_k，将C_0,C_1,C_2,C_3带入式 1 最后一项进行因式分解，提取相同的系数μ_k，可得到：

$$
g = [(g_0 - g_1 - g_2 + g_3)\times\mu_k - (-g_0 + 3\times g_1 - g_2 - g_3)]\times 0.5\times\mu_k + g_3 \qquad (2)
$$

采用这种方式仅需要 4 次乘法(2 次为固定系数乘法)与 8 次加法，其中固定系数乘法可通过移位器、加法器组合实现。

传统的位置系数模块将系数存储在固态存储器中，由查表法实现。固态存储单元无论是在 FPGA 或是 ASIC 设计中都是非常珍贵的，且每一种图像缩放格式的变换对应着一组系数，要将所有的系数都存储起来所需代价是非常大的。本发明设计的位置系数产生模块可用如下方法完成。设输入行含 X 个像素点，输出行含 Y 个像素点，Y/X 表示缩放比，而此比例即为每一行第一个插值点的位置系数μ_1，第二个插值点的系数为$[\mu_1+\mu_1]$(中括号表示对得到的和取小数部分)，依此类推，第三个点的插值位置为$[\mu_1+[\mu_1+\mu_1]]$等等。

本发明的优点：采用该种定标器所需硬件资源仅为一个累加器、寄存器与相应的辅助控制单元，且适用于任意比例的缩放。

本发明基于对缩放算法的分层次因式分解，显著降低了乘法器数量，在 FPGA、ASIC 设计中，一块芯片往往仅有十几个乘法器，采用本方法可大大增强高阶算法在实时处理芯片中的应用。同时本发明提出了一种配合定标器工作的系数产生模块，用一个加法器与少量的辅助控制单元代替了插值器系数存储单元，有效的降低了对存储容量的需求。本发明适用于采用多阶插值算法的图像定标器，可以完成任意比例的图像缩放功能。

该发明不仅节约硬件资源，而且高阶算法提高工作效率，降低成本，应用广泛。

附图说明：

图 1 是本发明的总体框图

图 2 是图 1 的插值器模块结构图

图 3 是图 1 的位置系数产生模块结构图

具体实施方式：

本发明具体实施方式如图 1 所示，主要由插值器模块 1，位置系数产生模块 2 构成。

以式（2）因式分解的结果为例，该插值器的具体实现方式如图 2 所示。其中寄存器 3、4、5、6 负责对输入的数据进行暂存，暂存的数据分别为 g_0, g_1, g_2, g_3；加法器 7、8、9、10、11、12、13、14 负责完成插值算法中的 8 次加、减法运算；乘法器 17、18 负责完成涉

及到μ_k的两次乘法运算；移位相加器 15、16 负责完成两次固定系数乘法运算。执行的顺序如图中所示，运算的时序由寄存器 3、4、5、6 驱动时钟控制。

位置系数产生模块由加法器 19、寄存器 20 组成，如图 3 所示。首先计算输出输入比，如该比值大于 1 则取小数部分，如小于 1 则直接取该比值。为了增加插值位置的精度，该小数可量化到 16 位甚至更高，但需要保证加法器 19 应与该比值的量化位数相同。每进行一次累加，得到一个插值点的位置信息，累加中大于 1 的部分自动舍除，所以寄存器中的值即为需要的插值位置信息。

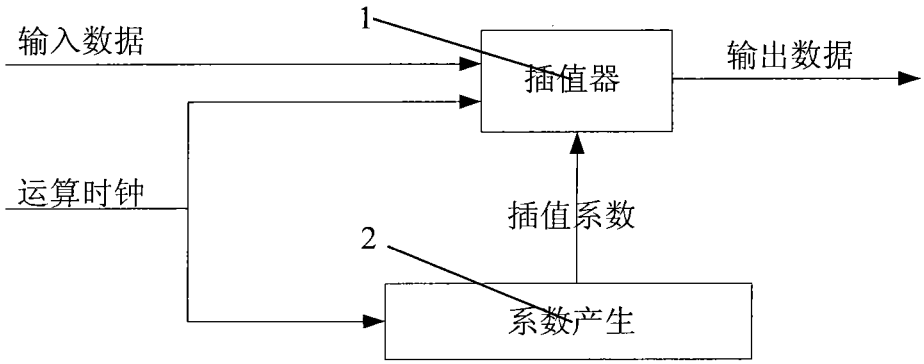

1
输入数据
插值器
输出数据
运算时钟
插值系数
2
系数产生

图 1

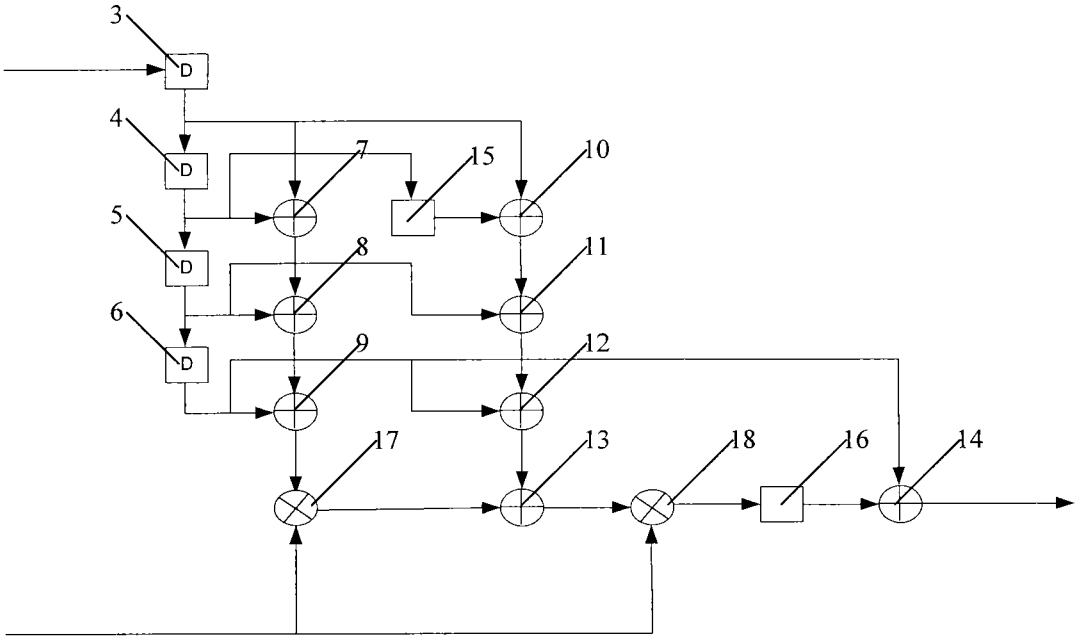

3
D
4
D
5
D
6
D
7
15
10
8
11
9
12
17
13
18
16
14

图 2

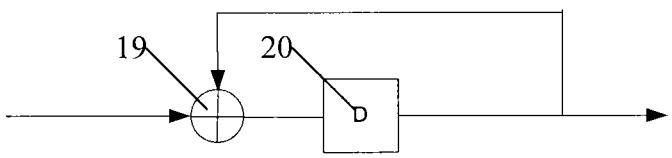

19
20
D

图 3