

PowerPC 平台下基于 Y'CbCr 域的图像合成引擎的设计与实现

杨 剑 支 琤 王 兴 东

(上海交通大学图像通信与信息处理研究所, 上海 200240)

摘 要 通过对传统的基于 RGB 域的图像合成方式在视频图像合成应用上的缺点的分析, 提出了基于 Y'CbCr 域的图像合成方法, 以实现视频图像进行快速合成, 同时讨论了在 PowerPC 平台上利用 Altivec 提高运算效率的可行性和有效性, 并描述了具体的实现过程。

关键词 Y'CbCr 视频 图像合成 PowerPC Altivec

中图分类号: TN911.73 **文献标识码**: A **文章编号**: 1006-8961(2006)11-1724-04

Design and Implementation of Y'CbCr Image Compositing Engine on PowerPC

YANG Jian, ZHI Cheng, WANG Xing-dong

(Institute of Image Communication and Information Processing, Shanghai Jiao Tong University, Shanghai 200240)

Abstract Disadvantages of traditional image compositing when applying to video image are analyzed, and a method of video image compositing in Y'CbCr space is propounded to composite video image fast. It also discusses the possibility and effectivity of implementing the video image compositing engine with Altivec on PowerPC, and describes the implementation progress in details.

Keywords Y'CbCr, video, image compositing, PowerPC, Altivec

1 引 言

图像合成是数字图像处理常用的技术之一, 在实际的视频图像处理中具有广泛的应用, 例如视频添加字幕、时码、台标、视频淡入淡出特效等都是用图像合成技术实现的。

RGB 是计算机中最常用的色彩空间。它是通过红、绿、蓝三基色的相加来产生其他的颜色。由于 RGB 颜色与设备的独立性的原因, 因而广泛应用于计算机图形、成像系统中。但是, 由于 RGB 颜色空间在应用于实际图像, 特别是视频图像时, 具有一定的局限性, 例如, RGB 分量具有相同带宽, 其亮度、色彩与 RGB 分量都具有相关性等, 因此, 视频数据

一般都采用亮度和两个色彩分量的表示方法, 如 YUV、YIQ、Y'CbCr 等颜色空间^[1]。其中, Y'CbCr 颜色空间是 ITU-R BT. 601 标准的一部分, 应用十分广泛。Y'CbCr 具有几种不同的抽样方式, 如 4:4:4, 4:2:2, 4:1:1, 4:2:0 等。常用的 RGB 颜色空间与 Y'CbCr 颜色空间之间的转换关系为^[2]

$$\begin{pmatrix} Y' \\ Cb \\ Cr \end{pmatrix} = \begin{bmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{bmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} + \begin{pmatrix} 128 \\ 128 \\ 128 \end{pmatrix} \quad (1)$$

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{bmatrix} 1.164 & 0.0 & 1.596 \\ 1.164 & -0.391 & -0.813 \\ 1.164 & 2.018 & 0.0 \end{bmatrix} \begin{pmatrix} Y' \\ Cb \\ Cr \end{pmatrix} - \begin{pmatrix} 16 \\ 128 \\ 128 \end{pmatrix} \quad (2)$$

其中, RGB 分量的数值范围为 [0, 255], Y' 分量的数值范围为 [16, 235], CbCr 分量的数值范围为 [16, 240]。

基金项目: 上海市科委基金项目 (03DZ15022)

收稿日期: 2006-08-03; 改回日期: 2006-08-12

第一作者简介: 杨剑 (1982 ~), 男, 2004 年毕业于上海交通大学获电子信息工程专业硕士学位, 现为上海交通大学图像通信研究所硕士研究生。主要研究方向为高清晰度数字电视技术。E-mail: yang.jian@situ.edu.cn

与其他图像不同,由于视频图像数据一般以 4:2:2 采样的 Y'CbCr 分量为主,因此应用于视频处理的图像合成引擎的设计与实现必须考虑到这一点,以最大化地提高处理的能力。本文提出了一种基于 Y'CbCr 域的图像合成方法,同时设计和实现了 PowerPC 平台上基于 Y'CbCr 域的图像合成引擎,并成功应用于苹果平台的字幕与非线性编辑软件中,使得软件性能大为提高。

2 基于 Y'CbCr 域的图像合成引擎

2.1 传统图像合成方式与视频图像合成

传统的图像合成方式是以 RGB 域的图像为处理对象,并采用 Alpha 叠加原理实现图像合成。一般说来,计算机中的图形系统都是以 RGB 颜色空间为基础,提供对图像合成处理的支持,如 3 维图形系统 OpenGL。而视频图像则以 Y'CbCr 域的数据为主。在这样的情况下,如果采用传统的图像合成方式对视频图像进行合成处理,则需要首先把 Y'CbCr 域的数据转换成 RGB 域的数据,然后才可进行图像合成,最后还需要将其转换为 Y'CbCr 域的数据。整个合成过程如图 1 所示。

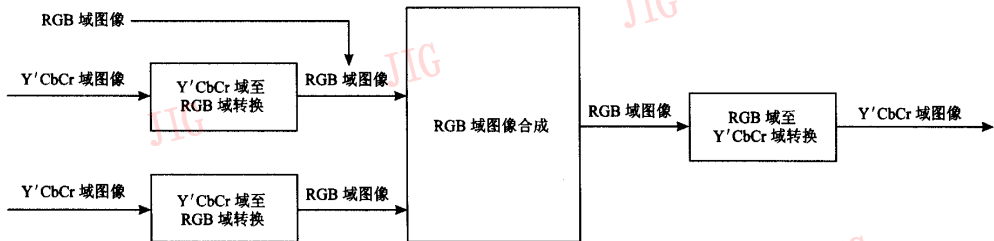


图 1 RGB 域图像合成应用于视频图像合成
Fig.1 Image compositing in RGB color space

2.2 基于 Y'CbCr 域的图像合成引擎的设计

根据以上分析,传统图像合成是在 RGB 域进行,其应用于视频图像合成时,由于与 Y'CbCr 视频数据不一致,需要进行额外的转换和处理,因此导致了图像质量下降和数据处理能力降低。基于 Y'CbCr 域的图像合成,则可以从根本上克服传统图像合成方式在视频图像处理方面的缺点,在合成处理过程中,它不会造成视频图像质量的损失,同时能够减少额外操作,还可以充分利用计算资源进行合成处理,在视频图像合成处理中具有重要的意义。

基于 Y'CbCr 域的图像合成引擎,还可以实际应

用图 1 可以看出,在这个处理过程中,需要对视频图像数据进行两次 Y'CbCr 域与 RGB 域之间的转换,其具有以下两个明显的缺点:

(1) Y'CbCr 域与 RGB 域之间的转换会导致视频图像质量的损失。由于 Y'CbCr 与 RGB 数据都是用整数表示,而转换矩阵却是小数形式的。通常在转换过程中,需要先把小数形式的转换矩阵用整数表示,然后进行整数运算,并截断到适当的数据范围来得到转换结果。其中,由于小数用整数表示会引入一定的误差,而且截断过程中也可能引入误差,因此两次 Y'CbCr 空间与 RGB 空间之间的颜色空间转换,会使得图像质量的损失更为严重。

(2) Y'CbCr 域与 RGB 域之间的转换会因消耗大量的计算时间,而降低系统的处理性能。合成计算和 Y'CbCr 域与 RGB 域之间的转换,都需要对每个像素进行计算,其中合成计算是进行 Alpha 叠加运算,而颜色空间转换是转换矩阵乘法运算,由于其计算量是 Alpha 叠加计算量的 3 倍左右,因此,颜色空间转换计算量是图像合成计算量的 3 倍左右。而采用传统图像合成方式对视频图像进行合成处理,则大部分的计算时间都会将消耗在 Y'CbCr 域与 RGB 域相互之间的转换上,这对于系统的性能具有很大的影响。

用于视频图像的字幕叠加等。根据实际情况,提出以下功能需求:

- (1) 输入:底层图像为 Y'CbCr 域的图像,顶层图像是包含 Alpha 通道的 Y'CbCr 域或者 RGB 域的图像,如字幕、台标、时码等;
- (2) 处理:采用图像合成处理中常用的 Alpha 叠加原理,对顶层图像与底层图像进行合成;
- (3) 输出:合成处理以后的 Y'CbCr 域图像;
- (4) 性能:尽可能提高处理效率,最好做到实时处理。

根据以上需求,设计的基于 Y'CbCr 域的图像合

成引擎的基本框图如图 2 所示。

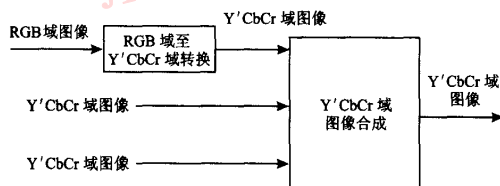


图 2 基于 Y'CbCr 域的图像合成引擎

Fig. 2 Image compositing engine in Y'CbCr color space

整个引擎主要包含以下两个功能模块：

(1) Y'CbCr 域图像合成模块：其输入的顶层和底层图像都是 Y'CbCr 域图像，顶层图像带有 Alpha 通道。该模块采用图像合成处理中常用的 Alpha 叠加原理，对顶层和底层图像进行合成处理，输出 Y'CbCr 域图像。这个模块是合成引擎的核心模块。

(2) RGB 域至 Y'CbCr 域转换模块：输入图像为带 Alpha 通道的 RGB 域图像，输出相应的带 Alpha 通道的 Y'CbCr 域图像。这个模块由于解决了顶层输入图像的格式问题，从而扩大了基于 Y'CbCr 域的视频图像合成引擎的应用范围。

3 Alpha 叠加算法

在现代计算机图形系统中，每个像素除了基本的颜色分量外，还有一个相关联的 Alpha 分量，其表示每个像素的不透明度。多个图像可以通过特定运算生成新的图像，以实现多个图像依次叠加的特殊视觉效果。Alpha 叠加就是实现这种特殊视觉效果的基本操作。

对每个像素的某一个分量值，设前景值为 A ，背景值为 B ，则 Alpha 叠加合成值 S 的基本计算公式如下^[3]：

$$S = A \times \alpha + B \times (1 - \alpha) \quad (3)$$

其中， α 表示前景像素颜色的不透明度，取值范围为 $[0, 0.1, 0]$ 。在实际应用中， α 通常用 8bits 无符号整数表示，取值范围为 $[0, 255]$ ，相应的计算公式为

$$S = A \times \frac{\alpha}{255} + B \times \frac{(255 - \alpha)}{255} \quad (4)$$

以上计算式中对 255 的除法操作可以通过级数展开转化为简单的移位操作^[4]。假设整数变量 a 和 b 的取值范围都是 $[0, 255]$ ，整数变量 $t = a \times b$ ，则级数展开为

$$\frac{t}{255} = \left(\frac{t}{256}\right) \left(1 + \frac{1}{256} + \dots\right) \approx (t \gg 8) + (t \gg 16) \quad (5)$$

需要注意的是，以上估值计算公式有一定的误差。例如，当 $a = 255, b = 1$ 时， $t = 255, (t \gg 8) + (t \gg 16) \neq 1$ 。为此，可以对上述计算式进行如下修正^[4]：

$$t = a \times b + 128, \frac{a \times b}{255} = (t \gg 8) + (t \gg 16) \quad (6)$$

4 PowerPC 平台下基于 Y'CbCr 域的图像合成引擎的实现

根据以上设计说明，采用 CPU 标量计算方式来实现基于 Y'CbCr 域的图像合成引擎是比较直接的，主要根据式(1)、式(2)、式(4)、式(6)进行计算即可。为了进一步提高数据处理的能力，以达到实时处理的目的，可以利用 CPU 的矢量扩展来进行数据计算。现代处理器一般都提供了矢量扩展运算，即通过并行数据处理来提高数据处理吞吐量。SIMD (single instruction, multiple data) 把操作的数据对象封装在特殊的矢量寄存器中，并提供了一个指令同时对多个数据执行相同操作的功能。SIMD 能够极大地提高数据处理能力，在 3 维图形处理、视频处理、图像处理以及科学计算等领域都有广泛的应用。

PowerPC 平台提供了 AltiVec 指令扩展，并在 G4 和 G5 系列处理器中得到应用^[5]。AltiVec 提供 128bits 的矢量执行单元、32 个独立的 128bits 寄存器、162 个操作指令及流式数据预取引擎。AltiVec 矢量执行单元本身包含子单元，每个子单元处理特定的指令，这样就进一步提高了矢量单元的并行处理能力。AltiVec 的 128bits 矢量由多个封装的标量数据元素组成，其封装类型可以是：单个 128bits 数据、16 个 8bits 整数、8 个 16bits 整数、8 个 16bits 像素、4 个 32bits 整数、4 个 32bits 像素或者 4 个 IEEE-754 单精度浮点数。AltiVec 指令通常分为：矢量装载与保存、数据预取、元素排列、算术和逻辑运算、比较与控制 5 类。所有矢量处理指令可以在单个周期执行完成，单个周期内最大可执行 16 个操作。

4.1 AltiVec 实现的可行性和有效性

Alpha 叠加和 RGB 域到 Y'CbCr 域转换都是对每个像素点进行数据运算，其对每个像素点的数据运算相互之间是独立的，没有相互的数据依赖关系，

也没有特定的计算先后次序要求,而且对每个像素点的计算过程是完全相同的。由于其满足 SIMD 优化的所有条件,因而使用 AltiVec 来实现图像合成引擎中的两个功能模块是充分可行的。

AltiVec 本质上是并行数据处理,其应用于 Alpha 叠加和 RGB 域到 Y'CbCr 域转换时,可以在一个指令周期内同时对多个像素执行相同的数据计算,也就是说其在周期内完成了标量方式多个周期的工作,扣除一些矢量处理的额外消耗,总的来说,仍然可以大大减少计算所需要的时间,因此整体上能够有效地提高运算的执行效率。

4.2 AltiVec 具体实现

由于 AltiVec 是把具有某种相同属性的数据元素封装为矢量,然后通过指令同时进行相同的数据操作,因而在应用 AltiVec 进行优化时,首先需要找出同时进行相同数据操作的数据元素。例如,在 Alpha 叠加计算过程中,顶层图像具有颜色分量和 Alpha 分量,若颜色分量具有相同的属性,则对每个颜色分量进行相同方式的处理,且 Alpha 分量与颜色分量具有对应关系,若颜色分量具有不同的属性,则需要进行不同的数据操作。另外,当原始输入数据格式与矢量形式不一致时,则需要先对原始图像数据矢量进行额外的预操作,即把具有相同属性的元素分离出来,分别表示为单独的矢量,以满足矢量内部以元素具有相同属性的要求;然后再对其进行矢量运算,经过以上步骤以后,原始图像数据就已经以矢量形式存储在 128bits 矢量寄存器中,下一步就是针对具体计算过程,使用恰当的 AltiVec 运算指令对这些矢量进行数据处理。最后,需要根据要求进行计算结果的存储。与装载过程相对应,在存储过程中,如果待存储数据的矢量形式与存储格式不一致,那么就需要先进行矢量形式转换到存储格式的预操作,然后再调用存储指令存储结果。

其具体程序的部分代码如下:

```
/* 读取原始图像数据 */
bV0 = vec_ld (index_bot + 0, srcBottom);
tV0 = vec_ld (index_top + 0, srcTop);
tV1 = vec_ld (index_top + 16, srcTop);
/* 分量分离 */
tYC = vec_perm (tV0, tV1, vPermT1);
tAA = vec_perm (tV0, tV1, vPermT2);
/* 计算 */
mHigh = vec_mule (bV0, tAA);
mLow = vec_mulo (bV0, tAA);
```

```
...
vTemp1 = vec_sr (mHigh, c8_u16);
vTemp2 = vec_sr (mLow, c8_u16);
/* 分量组合 */
bV0 = vec_mergel ((vUInt8)mHigh, (vUInt8)mLow);
bV0 = vec_adds (bV0, tYC);
/* 保存结果 */
vec_st (bV0, index_bot + 0, dest);
```

其他可能用到的指令包括:vec_xor, vec_splat_u16, vec_sl, vec_lvs1, vec_mladd 等。

4.3 AltiVec 实现测试结果

在 Mac OS X 平台下,分别以标量方式和 AltiVec 矢量方式实现了 Alpha 叠加算法,测试是以无压缩 8bits4:2:2 采样的 Y'CbCr 格式的视频图像为输入,其测试结果比较如表 1 所示。

表 1 实验结果

Tab.1 Experiment result

图像大小 (pixels)	标量(单帧)		AltiVec(单帧)	
	时间 (ms)	处理速度 (ns/pixel)	时间 (ms)	处理速度 (ns/pixel)
200 × 200	0.52	13.0	0.08	2.0
400 × 400	2.13	13.3	0.38	2.4
720 × 576	2.85	6.9	1.00	2.4
1440 × 720	7.27	7.0	2.55	2.5
1920 × 1080	14.66	7.1	5.09	2.5

5 结 论

基于 Y'CbCr 域的图像合成引擎,由于充分考虑到视频数据本身的格式,同时减少了不必要的颜色空间转换,并充分利用了 AltiVec 并行数据处理的优势,因此极大地提高了运算效率,减少了计算时间,在实际的视频图像合成处理中具有重要的意义。

参考文献 (References)

- Charles Poynton. A technical introduction to digital video[M]. New York: John Wiley & Sons, 1996.
- Keith Jack. Video demystified (Fourth Edition) [M]. Burlington: Vermont, USA, Newnes, 2004.
- Thomas Porter, Tom Duff. Compositing digital images[J]. Computer Graphics, 1984, 18(3): 253 ~ 259.
- Smith Alvy Ray. Image compositing fundamentals[EB/OL]. ftp://ftp.alvyray.com/Acrobat/4_Comp.pdf.
- Apple Computer, Inc. Velocity engine resources on ADC (apple developer connection) site[EB/OL]. http://developer.apple.com/hardwaredrivers/ve/index.html.