

doi:10.3969/j.issn.1673-9833.2014.04.019

# DM642 图像数据可视化

王鹏程, 吴金津, 龙永新

(湖南工业大学 计算机与通信学院, 湖南 株洲 412007)

**摘要:** 采用 Matlab 和 CCS 的探针导入功能, 将静态图像分解成对应的 DAT 数据文件, 再导入 DM642 进行图像处理, 通过探针的导出功能和 Graph 工具, 在硬件平台上可以观察动态图像的处理结果。介绍了 CCS 的数据文件 DAT 格式, 以及静态图像和实时图像生成 DAT 文件的方法。通过应用实例说明了本方法能给算法调试带来便利, 避免了实验环境的干扰。

**关键词:** DM642; CCS 数据文件; 图像处理

**中图分类号:** TP391

**文献标志码:** A

**文章编号:** 1673-9833(2014)04-0086-05

## The DM642 Image Data Visualization

Wang Pengcheng, Wu Jinjin, Long Yongxin

(School of Computer and Communication, Hunan University of Technology, Zhuzhou Hunan 412007, China)

**Abstract:** Applied Matlab and the probe importing function of CCS to convert the static image into data file of DAT format, and then import to DM642 for image processing. Through the probe exporting function and Graph tool, observed the processing result of dynamic image on the hardware platform. Introduced the data file DAT format of CCS and the method of generating DAT files for static image and real-time image. The application example shows that this method can bring convenience for algorithmic debugging and avoid the interference of experimental environment.

**Keywords:** DM642; CCS data file; image processing

随着 DSP (digital signal processing) 技术的不断发展, DSP 芯片的集成度、运算速度、数据吞吐率等不断地提高, 它能应用到各种领域, 如传统的控制、离散信号处理、实时视频处理与传输等领域。TMS320DM642 芯片<sup>[1]</sup>是 TI (Texas Instruments) 公司生产的 DSP C6000<sup>[2]</sup>系列中最新的定点 DSP, 其核心是 C6416 型高性能数字信号处理器, 具有极强的处理性能、高度的灵活性和可编程性, 同时外围集成了非常完整的音频、视频和网络通信等设备及接口, 特别适用于机器视觉、医学成像、网络视频监控、数字广播以及基于数字视频/图像处理的消费类电子

产品等领域。但是, 利用 DSP 进行图像处理时, 由于实验环境的变化、硬件设备的干扰等, 无法准确地观察到实验结果, 也不能对不同算法的实验结果进行比较。针对 DM642 芯片的图像数据可视化问题, 本文提出在图像处理前先将图像数据导入 CCS<sup>[3]</sup> (code composer studio) 中, 再将处理后的数据显示出来, 该操作能给研究人员提供很大的帮助。本文介绍了 CCS 数据文件格式以及如何将静态图像和视频图像存储为相应的 DAT 数据文件, 再介绍怎样将图像的 DAT 数据导入 DSP 进行算法调试, 最后在硬件平台将实时处理的图像数据导出并显示出来。

收稿日期: 2014-04-08

基金项目: 湖南省科技发展基金资助项目 (2011GK3145)

作者简介: 王鹏程 (1989-), 男, 湖南岳阳人, 湖南工业大学硕士生, 主要研究方向为图像处理,

E-mail: wang\_pc\_10@126.com

## 1 CCS 数据文件

CCS是由TI生产的DSP开发工具。在Windows操作系统下, CCS采用图形接口界面, 提供了环境配置、源文件编辑、程序调试、跟踪和分析等工具。CCS可识别的数据文件格式为DAT<sup>[3]</sup>, 其具有特定的文件格式, 由21个字符构成文件头信息, 文件头信息之外的数据即为图像数据文件。文件头由固定标识、数据格式、起始地址、页类型和数据块大小5部分组成<sup>[4]</sup>。固定标识为1651, 数据格式包括十六进制、十进制、十进制长整形和十进制浮点型4种格式, 分别用1, 2, 3, 4表示。起始地址为存储数据的起始地址。页类型包括数据和程序2种类型, 分别用0和1表示。数据块大小表示数据的长度<sup>[4]</sup>。一个CCS数据文件格式如下所示:

```
165110x800000003
0x08080863
0x84101A0A
0xFFFFFFFF
```

从上面的示例可以看出, 该数据的格式为1, 即十六进制; 数据存储的起始地址为0x8000000; 数据长度为3。CCS的探针点可以从个人电脑的文件中读取DAT数据, 并将读取到的数据保存到对应的目的地址, 供算法使用, 也可以将对应地址的数据保存到个人电脑对应的文件中。上述操作需用到CCS中数据提取和保存函数, DATA LOAD<sup>[5]</sup>函数读取DAT数据, DATA SAVE<sup>[5]</sup>函数完成DAT数据保存。

## 2 静态图像的数据处理及应用

色彩空间是颜色集合的数学表示。最常用的颜色模型是: RGB<sup>[6]</sup>(常用于计算机图形学中); YIQ, YUV或YCbCr(常用于视频系统中); CMYK(常用于彩色打印)。而CCS能处理和显示的颜色模型是RGB和YUV, 前者针对的是静态图像数据, 后者针对的则是视频数据。静态图像数据的处理过程如图1所示<sup>[7]</sup>。利用DM642芯片对静态图像进行处理前, 需要先将静态图像数据转化为对应的DAT数据文件。CCS提供了2种读取RGB数据的方式: 一种是分开读取RGB的3个分量, 另一种是读取交叉存取的RGB数据<sup>[8]</sup>。

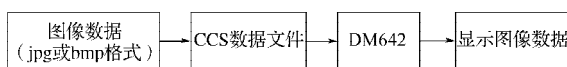


图1 静态图像数据处理过程

Fig. 1 Static image data processing

在对图像进行处理前, 先将静态图像通过Matlab转化为对应的DAT文件, 然后导入DM642进

行处理, 图像的处理结果可以通过Graph—>Image命令显示。

### 2.1 RGB分量的生成

CCS对图像进行处理前, 可以使用VC++6.0或Matlab软件先将图像转化成相对应的DAT文件, 本文使用Matlab软件。Matlab<sup>[9]</sup>是一种强大的分析、计算和可视化工具, 特别适合对数字信号处理算法进行模拟和分析。

#### 2.1.1 R, G, B分量的生成

以jpg格式的图像为例, 利用Matlab提取该图像的R, G, B 3个分量, 并分别保存在R.DAT, G.DAT, B.DAT文件中。具体算法过程如下:

- 1) 读入一幅jpg格式的图像;
- 2) 提取图像的R, G, B 3个颜色分量;
- 3) 打开文件R.DAT, G.DAT, B.DAT, 并写入数据头文件;

4) 将R, G, B分量分别存入对应的DAT文件中, 并压缩成32位;

- 5) 关闭文件。

算法核心代码如下。

```
fprintf(fid1,'16511800000000%\n',M*N/4);
% 输出 CCS 的数据格式 DAT 文件的文件头
fprintf(fid2,'16511800000000%\n',M*N/4);
% 输出 CCS 的数据格式 DAT 文件的文件头
fprintf(fid3,'16511800000000%\n',M*N/4);
% 输出 CCS 的数据格式 DAT 文件的文件头
form=1:M
for n=1:N/4
    fprintf(fid1,'0x%s%s%s%s\n',dec2hex(R(m,n*4),2),
        dec2hex(R(m,n*4-1),2), dec2hex(R(m,n*4-2), 2),
        dec2hex(R(m,n*4-3),2));
    % 将 4 个 R 分量像素数据(无符号 8 位数据)压缩
    成一个数据(32 位数据)
    fprintf(fid2,'0x%s%s%s%s\n',dec2hex(G(m,n*4),2),
        dec2hex(G(m,n*4-1),2),dec2hex(G(m,n*4-2),2),
        dec2hex(G(m,n*4-3),2));
    % 将 4 个 G 分量像素数据(无符号 8 位数据)压缩
    成一个数据(32 位数据)。
    fprintf(fid3,'0x%s%s%s%s\n',dec2hex(B(m,n*4),2),
        dec2hex(B(m,n*4-1),2),dec2hex(B(m,n*4-2),2),
        dec2hex(B(m,n*4-3),2));
    % 将 4 个 R 分量像素数据 (无符号 8 位数据) 压
    缩成一个数据 (32 位数据)
```

#### 2.1.2 交叉RGB数据的生成

在CCS的数据显示中, 提供了RGB交叉存取的

方式。因此,在存储数据时,可以将数据以RGB交叉存取的方式存取在同一个数据DAT文件中。该算法的过程与R、G、B分量的生成过程大体一致,只是步骤3)为将R、G、B3个分量交叉存入DAT文件中。

算法核心代码如下。

```
fprintf(fid,'16511800000000%\n',M*N*4);
for i=1:M
for j=1:N
    fprintf(fid,'0x%s%s%s%s%\n',dec2hex(B(i,j),2),
        dec2hex(G(i,j),2),dec2hex(R(i,j),2),dec2hex(z(i,j),2))
```

### 2.1.3 转化为Y, U, V三个分量

Y, U, V分别指的是亮度、红色色度和蓝色色度值。YUV的数据格式是常用的数据格式,能将亮度与色度信号分离,与RGB信号传输相比,它最大的优点在于只需占用极少的频宽。它与上面保存R、G、B3个分量的方法类似,将3个值保存下来。RGB转化为YUV公式<sup>[10]</sup>如下:

$$Y=0.3R+0.59G+0.11B;$$

$$U=(B-Y)0.493;$$

$$V=(R-Y)0.877.$$

通过Matlab获得图像的DAT数据后,经CCS的探针点DATA LOVD将数据导入到DM642 simulate相应的地址中。DM642就可以对采集到的静态图像数据进行处理,并通过Graph命令显示出来。

## 2.2 RGB DAT数据的显示

CCS中提供了多种方法将经程序处理后的数据显示出来,包括时间/频率、星座图、眼图和图像等。选择View→Graph命令<sup>[11]</sup>,可以得到所需要的图表。图2是Graph设置框图,其中Image功能可以用来显示图像数据。Image菜单包括很多的参数选择,根据不同的参数设定,有不同的显示效果。Graph Title指定图像窗口的标题。Color Space Operations指定需要显示的图像的色彩空间RGB或YUV。如果选择YUV色彩空间,接下来还需要选择对应的YUV色彩关系4:2:2或4:2:0或4:1:1<sup>[7]</sup>,Start Address中的Y Source, Cb Source, Cr Source是3个分量对应的起始地址。如果选择RGB色彩空间,首先判断图像数据是否是交叉存取,如果是交叉存取,则在Interleaved里选择Yes,在RGB里输入对应的RGB地址,并在Image RGB Order中选定对应的RGB顺序,如果RGB数据是以32位存取的,则选择32,如果是24位,则选择24位;如果不是交叉存取,则分别将R、G、B的起始地址输入Start R Address, Start G Address, Start B Address中。Lines Per Display输入对应的垂直像素。Pixels Per Line输入对应的水平像素<sup>[10]</sup>。图3是Lena图像的RGB交叉显

示的参数设置。通过图3的设置,可将读入DSP simulate的静态图像显示出来。Lena显示结果如图4所示。

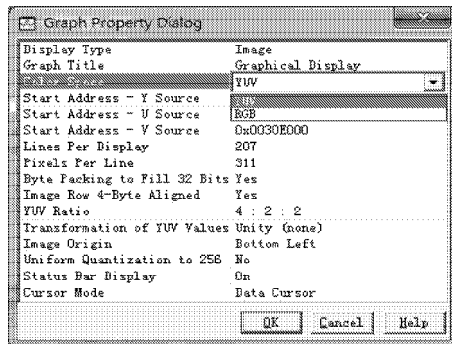


图2 Graph属性设置框

Fig. 2 Graph Property Dialog

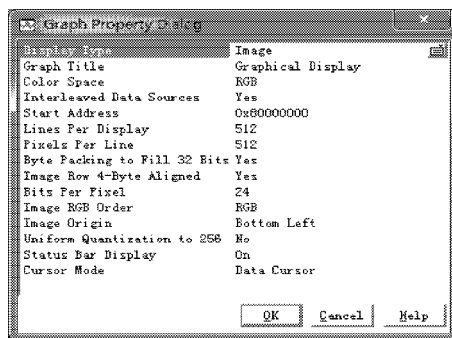


图3 Lena显示的设置

Fig. 3 Lena display setting

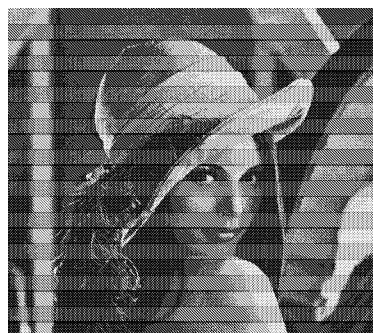


图4 Lena显示图

Fig. 4 Lena display

## 2.3 应用

将静态图像数据的DAT文件存储在DM642对应的地址上。通过这些地址,可以对静态图像数据进行访问和处理。图5是在DM642 simulate环境中,检测静态灰度图像Lena的Harris角点的结果图。如果读入的静态图像数据是YUV,则可以直接对Y数据进行处理;如果是RGB,则需先将RGB数据转化为灰度数据Gray,转化公式为

$$Gray=R0.299+G0.587+B0.114.$$



图5 Harris 角点检测结果

Fig. 5 Harris corner detection results

### 3 实时图像数据采集

#### 3.1 采集过程

利用实验平台进行实时图像处理<sup>[11]</sup>时, 实验环境的变化对实验结果会产生干扰。如果能将 DSP 处理过程中的数据通过图像的形式显示出来, 将有利于研究人员对结果进行观察及验证算法的好坏。图6是 DM642 实时图像采集过程。

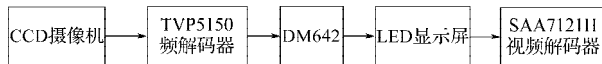


图6 DM642 视频采集系统

Fig. 6 DM642 video acquisition system

CCD 摄像机采集视频信号, 经过 TVP5150 视频解码器解码后, 生成 BT.656<sup>[12]</sup>信号, 获得对应格式为 YCbCr 的数字信号, 将其存储在对应的地址; DM642 处理完后, 处理结果通过 SAA7121H 视频解码器解码, 转化为对应的 PAL/NTSC 格式数据, 输出在 LED 显示屏。由于受实验环境的变化和外界的干扰, 经 DSP 处理完后, 输出在 LED 显示屏的数据会出现很大的变化, 不利于观察算法的结果, 但是, DATA SAVE 提供了图像数据保存功能, 通过设置对应的起始地址和数据长度, 即可将视频处理过程中的某一帧视频数据保存在对应的 DAT 文件中, 并通过上文的方法将帧数据显示出来。由于实时图像数据是隔行扫描<sup>[13]</sup>, 因此获得图像数据后, 需先对奇偶场进行合并处理, 再显示图像数据。图7是图像合并过程。

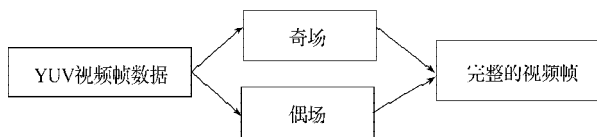


图7 图像数据的合并

Fig. 7 Image Data merging

合并算法的核心代码如下。

```
for(i=0;i<NumLines/2;i+=2)
```

```

for(j=0;j<2*NumPixels;j++)
{
*(MergeY+i*NumPixels/6+j)=*(Y+i/2*NumPixels+j);
*(MergeY+(i+1)*NumPixels/6+j)=*(Y+(i/2+NumLines/2)*NumPixels+j);
if(j<NumPixels/2)
{
*(MergeCr+i*NumPixels/2+j)=*(Cr+(i/2)*NumPixels/2+j);
*(MergeCr+(i+1)*NumPixels/2+j)=*(Cr+(i/2+NumLines/2)*NumPixels/2+j);
*(MergeCb+i*NumPixels/2+j)=*(Cb+(i/2)*NumPixels/2+j);
*(MergeCb+i*NumPixels/2+j)=*(Cb+(i/2+NumLines/2)*NumPixels/2+j);
}
}
合并前后的图像如图8~9所示。

```



图8 合并前的图像

Fig. 8 Image before merging



图9 合并后的图像

Fig. 9 Image after merging

#### 3.2 应用

在基于特征点的实时视频拼接过程<sup>[13]</sup>中, 首先需要检测出每一帧图像的特征点, 由于外界实验环境的干扰以及帧数据的变化, 帧图像检测完角点后, 无法在 LED 显示屏上精确显现出来, 但是, 通过 CCS 的处理, 则可以清晰地显示结果。图10为对图像9

局部进行角点检测的结果。截取出了中间 120\*96 像素的结果图。



图 10 角点检测结果

Fig. 10 Corner detection results

## 4 结语

针对 DM642 芯片的图像数据可视化问题, 本文提出在图像处理前先将图像数据导入 CCS<sup>[3]</sup>中, 再对图像进行处理, 最后将处理后的数据显示出来。通过实验可以发现, 利用 Matlab 与 CCS 探针功能, 可以将静态图像和视频图像数据读入 DM642, 并将相应的图像数据存储于 DM642 的对应地址, 通过这些地址信息, 可以对图像数据进行访问和处理, 这有助于研究人员调试算法, 同时, 在硬件平台环境, 可以将实时图像处理过程的数据导出, 并通过 CCS → Graph → Image 命令显示, 这将有助于观察算法的处理效果, 避免了实验环境对实验结果的干扰。

### 参考文献:

- [1] 朱晓鼎, 张东, 刘发志. 基于 TMS320DM642 芯片视频系统的设计与应用[J]. 计算机工程与设计, 2008, 29(9): 2233-2235.  
Zhu Xiaoding, Zhang Dong, Liu Fazhi. Design and Application of Video System Based on TMS320DM642 [J]. Computer Engineering and Design, 2008, 29(9): 2233-2235.
- [2] Zapata J, Ruiz R. Rapid Development of Real-Time Applications Using MATLAB/Simulink on TI C6000-Based DSP[C]//Proceedings of the 5th WSEAS International Conference on Education and Educational Technology. Tenerife: [s. n.], 2006: 104-110.
- [3] Texas Instruments. TMS320C6000 Code Composer Studio Tutorial[S]. Dallas: Texas Instruments, 2000: 20-40.
- [4] 刘巍伟. 基于 DSP 图像处理的 MATLAB 辅助研究[J]. 廊坊师范学院学报: 自然科学版, 2010, 10(3): 43-45.  
Liu Weiwei. According to DSP Picture MATLAB of the Processing the Assistance Study[J]. Journal of Langfang Teachers College: Natural Science Edition, 2010, 10(3): 43-45.
- [5] 李利. DSP 原理及应用[M]. 中国水利水电出版社, 2004: 160-170.  
Li Li. DSP Principle and Applications[M]. China Water and Power Press, 2004: 160-170.
- [6] 邵丹, 韩家伟. YUV 与 RGB 之间的转换[J]. 长春大学学报, 2004, 14(4): 51-53.  
Shao Dan, Han Jiawei. Intertransformation Between YUV and RGB[J]. Journal of Changchun University, 2004, 14(4): 51-53.
- [7] 刘松涛, 周晓东, 沈同圣. DSP 图像采集处理系统的数据可视化[J]. 海军航空工程学院学报, 2003, 18(6): 677-679.  
Liu Songtao, Zhou Xiaodong, Shen Tongsheng. The Data Visualization of DSP Image Acquisition and Processing System[J]. Journal of Naval Aeronautical Engineering Institute, 2003, 18(6): 677-679.
- [8] Texas Instruments. TMS320DM642 Video/Imaging Fixed-Point Digital Signal Processor[M]. Dallas: Texas Instruments, 2010: 68-75.
- [9] Du R J, Liu J, Li Y S. Research of DSP Code Generation Based on Matlab-DSP Intergrated Design[J]. Applied Mechanics and Materials, 2014, 551: 244-247.
- [10] Zhao Jie, Li Nan, Yang Feng. Multi-Channel Image Acquisition Preprocessing System Based on TMS320DM642[C]//2008 International Conference on Machine Learning and Cybernetics. [S. l.]: IEEE, 2008: 2944-2949.
- [11] 李小红. 基于 DSP 图像处理实验系统的开发与应用[J]. 合肥学院学报: 自然科学版, 2009, 19(4): 38-41.  
Li Xiaohong. Development and Application of Image Processing Laboratory System Based on DSP[J]. Journal of Hefei University: Natural Sciences, 2009, 19(4): 38-41.
- [12] 郭慰萱, 郭宝龙. 基于 DM642 的视频处理系统设计与实现[J]. 电子设计工程, 2009, 17(1): 79-81.  
Guo Weixuan, Guo Baolong. Design and Realization of Video Processing System Based on DM642[J]. Electronic Design Engineering, 2009, 17(1): 79-81.
- [13] 张云峰. 基于 DSP 的实时图像拼接技术[J]. 液晶与显示, 2013, 28(6): 963-967.  
Zhang Yunfeng. Real-Time Image Mosaic Technology Based on DSP[J]. Chinese Journal of Liquid Crystals and Displays, 2013, 28(6): 963-967.

(责任编辑: 邓彬)

