

# 船舶稳像系统算法研究

王贻先, 张维华

(武汉理工大学信息工程学院, 武汉 430070)

5 **摘要:** 由于在视频采集过程中因为一些原因, 摄像机出现晃动, 会造成视频图像质量的下降, 因此视频稳像功能就显得尤为重要。本文针对船舶的特殊环境, 提出一种硬件稳像解决方案, 此系统基于 TI DM6437 达芬奇视频开发系统。本文阐述了稳像系统的系统组成、软件设计以及稳像算法, 并且详细介绍了稳像算法及对算法的改进等。实验表明, 此稳像系统具有较好的实际应用价值。

10 **关键词:** 视频稳像; 船舶; DM6437; 双线性插值; DSP/BIOS  
中图分类号: TP391

## Ship Video Stabilization System Algorithm

WANG Yixian, ZHANG Weihua

15 (Information Engineering, Wuhan University Of Technology, Wuhan 430070)

**Abstract:** The collection process in the video for some reason, the camera shake occurs, will cause a decline in the quality of the video image, so the video image stabilization feature is particularly important. In this paper, the special circumstances of the ship, a hardware image stabilization proposed solutions, this system development based on TI DM6437 DaVinci video system. Image Stabilization System This paper describes the system components, software design and stabilizing algorithm, and introduces the algorithm and image stabilization and improvement of the algorithm. Experiments show that the stabilization system has good practical value.

20 **Key words:** video image stabilization; ship; DM6437; bilinear interpolation; DSP / BIOS

## 25 0 引言

在摄像的过程中, 经常会由于摄像载体的不稳定造成输出视频的抖动, 这对于视频的观看或处理都是不利的。视频稳像技术的目的就是要将这些不稳定视频当中的抖动分量消除, 形成较为稳定的视频输出。稳定视频图像的基本方法可以分为三大类: 光学稳像、机械稳像和电子(数字)稳像。与其它两种方法相比, 电子稳像结合了电子技术、计算机技术和数字信号处理技术, 具有更小的体积、更低的成本, 以及更广泛的应用场合, 因此得到了广泛的研究<sup>[2]</sup>。

30 视频稳像技术在船舶监控领域中具有重要的意义。在信息获取方面, 随着世界各国视频处理技术的发展, 监控系统获取图像能力的增强, 特别是视频图像所具有的直观、丰富、便于获取、便于存储和便于传输等特点, 使视频稳像和跟踪技术的应用成为了未来船舶监控中获取信息的重要手段。

35 在船舶监控领域中, 由于成像系统受到船体姿态变化、设备振动等影响, 将会导致在监视器上输出的视频出现波动、模糊等现象。比如对于机舱内设备的监控过程中, 由于载体的振动, 导致在集控室或驾驶台的监视器上看到的视频图像中, 各仪表的表盘是模糊不清的, 无法准确地读取各设备的运行数据, 因此失去了监控的意义。同时, 这种不稳定的视频输出还会使机舱和驾驶台的工作人员产生疲劳感或错觉, 从而导致误判或漏判, 给船舶的运行和

---

作者简介: 王贻先(出生于1981年), 男, 武汉理工大学信息工程学院硕士研究生, 主要研究方向: 视频图像处理. E-mail: snowflakes8183@163.com

管理带来很大的风险。因此,本文针对这一点,通过一定的视频处理技术来消除或减轻载体运动对视频质量的影响,能够在很大程度上提高船舶监控的效率,为船舶的运动、管理提供了安全保障<sup>[1]</sup>。

## 1 系统组成

45 本系统由 DM6437 达芬奇开发平台组建而成,包括视频采集、视频显示、时序控制电路以及电源电路等部分组成。系统结构如图 1 所示。

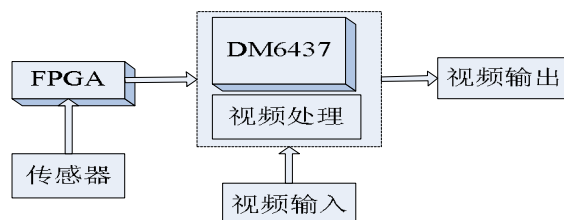


图 1 视频稳像系统组成框图

50 系统以 TI DM6437 为核心算法处理器,充分利用达芬奇开发平台的技术优势,采用数字摄像机采集视频图像,视频图像以 YUV422 格式存储及显示,FPGA 利用陀螺仪、传感器等测出图像帧需要旋转的角度、平移的像素,通过串口发送参数给 DSP 做图像处理,图像处理算法在 DSP 中负责视频处理,最终将图像在显示器上显示出来。

## 2 软件设计

### 2.1 算法设计

电子稳像的前提条件是先获得图像的全局运动矢量,根据获得的图像运动矢量方法的不同,电子稳像的基本方法有两种<sup>[3]</sup>:

1. 利用传感器检测摄像机的运动矢量再转化为图像的运动量即在摄像机上设置两个角度传感器用以检测上下、左右方向上的角速率,再通过放大、滤波去除对图像质量影响大的高频成分,然后经过 A/D 转换存入计算机中作为控制依据。采用像移补偿技术补偿图像运动,从而实现图像的稳定。此方法易于实现,但其检测精度要依赖于具有高精度、对振动敏感的陀螺仪。

2. 利用稳像算法实现电子稳像图像采集后,根据要求提取场或帧的图像,当物体做匀速运动时,帧处理优于场处理,这是因为奇场和偶场运动相同,可以共用一个运动矢量,帧处理时只需传送一个运动矢量,但当物体运动是非匀速运动时,场处理则优于帧处理。选取图像后与参考图像利用稳像算法获取图像的运动矢量,参考图像可以是存储器中的固定图像,也可能是图像缓存中的相邻帧图像。此稳像方法的关键是利用稳视频图像的数字稳像系统设计与目标识别像算法计算图像的运动量。

70 本文对比两种不同的视频处理算法,针对船舶的特殊环境,考虑到后一种方法对处理器的要求比较高,其实时性和精度难以得到保障,因此在本文中采用了前一种方法,采用了传感器检测图像的运动矢量,降低了对处理器的负担,视频的实时性和质量也得到保证。

#### 2.1.1 图像旋转算法

旋转有一个绕着什么转的问题,通常的做法是以图像的中心为圆心旋转。在图 2 所示的坐标系 A 中,以图像的中心为原点 O,向右为 x 轴正方向,向下为 y 轴正方向,点(x<sub>0</sub>, y<sub>0</sub>)

75 顺时针旋转  $\alpha$  角度后坐标变换为  $(x_1, y_1)$ 。设  $r$  为点  $(x_0, y_0)$  到原点  $(0, 0)$  的距离，在旋转过程中， $r$  保持不变， $\beta$  为  $r$  与  $x$  轴之间的角度，则有：

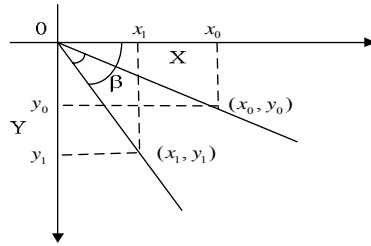


图2 图像旋转示意图

80 旋转  $\alpha$  角度后：

$$x_1 = r * \cos(\beta) ; y_1 = r * \sin(\beta) ; \quad (1)$$

旋转前：

$$\begin{aligned} x_0 &= r * \cos(\beta - \alpha) \\ &= r * \cos(\beta) * \cos(\alpha) + r * \sin(\beta) * \sin(\alpha) \\ &= x_1 * \cos(\alpha) + y_1 * \sin(\alpha) ; \end{aligned} \quad (2)$$

85

以矩阵的形式表示为：

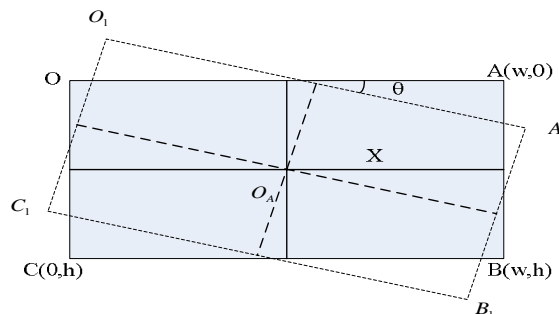
$$\begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} \quad (3)$$

公式 (3) 的逆变换为：

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} \quad (4)$$

90 如图3中所示，图形  $OABC$  以  $O_A$  为中心顺时针旋转  $\theta$  角度，图形变为  $O_1A_1B_1C_1$ ，可以看出旋转后图形变大了，因为显示设备固定，因此在本文中需要将扩大的部分去除，仅显示以  $O_A$  为中心的原图形大小。

如果以  $O$  点为坐标系原点，向右为  $X$  轴正方向，向下为  $Y$  轴正方向，记为坐标系  $B$ 。设图像的宽为  $w$ ，高为  $h$ ，两种坐标系间的转换关系如图3所示。则有：



95

图3 图像旋转及坐标系的转换关系

$$x_A = x_B - 0.5w ; y_A = y_B - 0.5h ; \quad (5)$$

以矩形的形式表现出来：

$$100 \quad \begin{bmatrix} x_A \\ y_A \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -0.5w \\ 0 & 1 & -0.5h \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_B \\ y_B \\ 1 \end{bmatrix} \quad (6)$$

公式 (6) 的逆变换为:

$$\begin{bmatrix} x_B \\ y_B \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -0.5w \\ 0 & 1 & -0.5h \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_A \\ y_A \\ 1 \end{bmatrix} \quad (7)$$

于是, 坐标系 B 中图像的旋转变换可分为 3 步: ①应用公式 (6), 将坐标系 B 变成坐标系 A; ②应用公式 (4), 在坐标系 A 中将该点顺时针旋转角; ③应用公式 (7), 将坐标系 A 变回坐标系 B。这样, 我们就得到了变换矩阵, 是上面 3 个矩阵的级联。

$$105 \quad \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -0.5w_2 \\ 0 & 1 & -0.5h_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -0.5w_1 \\ 0 & 1 & -0.5h_1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos(\theta) & -\sin(\theta) & -0.5w_1 \cos(\theta) + 0.5h_1 \sin(\theta) + 0.5w_2 \\ \sin(\theta) & \cos(\theta) & -0.5w_1 \sin(\theta) - 0.5h_1 \cos(\theta) + 0.5w_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} \quad (8)$$

公式 (8) 的逆变换为:

$$110 \quad \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0.5w_1 \\ 0 & 1 & 0.5h_1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -0.5w_2 \\ 0 & 1 & -0.5h_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos(\theta) & -\sin(\theta) & -0.5w_2 \cos(\theta) - 0.5h_2 \sin(\theta) + 0.5w_1 \\ \sin(\theta) & \cos(\theta) & 0.5w_2 \sin(\theta) - 0.5h_2 \cos(\theta) + 0.5w_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} \quad (9)$$

其中,  $w_1, h_1$  表示原图的宽和高,  $w_2, h_2$  表示新图的宽和高。

这样, 对于新图中的每一点  $(x_1, y_1)$ , 可以根据公式 (9) 求出对应原图中的  $(x_0, y_0)$ , 得到它的灰度, 如果超出原图范围, 则填成黑色。要注意的是, 原始图像的像素坐标为整数, 由于有浮点运算, 变换后的目标图像的坐标位置可能不是整数。所以, 图像的旋转处理, 除了要

115 要进行几何运算, 还需要对目标图像的坐标位置进行插值处理<sup>[4]</sup>。

一般常用的插值方法: 最近邻域法、双线性插值、双三次线性插值。

最近邻域法: 最近邻法是一种简单的插值方法, 它是通过计算于点  $(x_0, y_0)$  邻近的 4 个点, 将与点  $(x_0, y_0)$  最近的整数坐标点的像素值取为  $(x_0, y_0)$  点的像素值。

120 双线性插值<sup>[5][6]</sup>: 对于一个目的像素, 通过反向变换得到浮点坐标  $(x+u, y+v)$ , 其中  $x, y$  均为非负整数,  $u, v$  为  $[0, 1)$  区间的浮点数, 则这个像素的值可由源图中坐标为  $(x, y)$ 、 $(x+1, y)$ 、 $(x, y+1)$ 、 $(x+1, y+1)$  4 个点所对应的像素的值确定, 即:  $f(x+u, y+v) = (1-u)(1-v)f(x, y) + (1-u)v f(x, y+1) + u(1-v)f(x+1, y) + uv f(x+1, y+1)$ 。其中,  $f(x, y)$  代表源图像  $(x, y)$  处的像素值, 以此类推。

125 这就是双线性插值法, 它考虑到了点  $(x, y)$  的 4 个直接邻点对它的影响, 因此一般可

以得到令人满意的插值效果。

双三次线性插值<sup>[5][6]</sup>：双线性插值法考虑到浮点坐标 (x+u, y+v) 的 4 个直接邻点对它的影响，而双三插值法则考虑到该点周围 16 个邻点对它的影响。目标坐标的像素值可由如下的公式得到： $f(x+u, y+v) = ABC$ 。

$$130 \quad A = [S(u-1) \quad S(u) \quad S(u+1) \quad S(u+2)]$$

$$B = \begin{bmatrix} f(x-1, y-1) & f(x-1, y) & f(x-1, y+1) & f(x-1, y+2) \\ f(x, y-1) & f(x, y) & f(x, y+1) & f(x, y+2) \\ f(x+1, y-1) & f(x+1, y) & f(x+1, y+1) & f(x+1, y+2) \\ f(x+2, y-1) & f(x+2, y) & f(x+2, y+1) & f(x+2, y+2) \end{bmatrix}$$

$$C = [S(v+1) \quad S(v) \quad S(v-1) \quad S(v-2)]^T$$

插值函数  $S(x) = \sin(\pi x) / \pi x$  可以采用以下多项式近似。

$$S(x) = \begin{cases} 1 - 2|x|^2 + |x|^3, & (0 \leq |x| < 1) \\ 4 * 8|x| + 5|x|^2 - |x|^3, & (0 \leq |x| < 2) \\ 0, & (|x| \geq 2) \end{cases}$$

135 三种插值方法按照相同角度旋转的效果图：

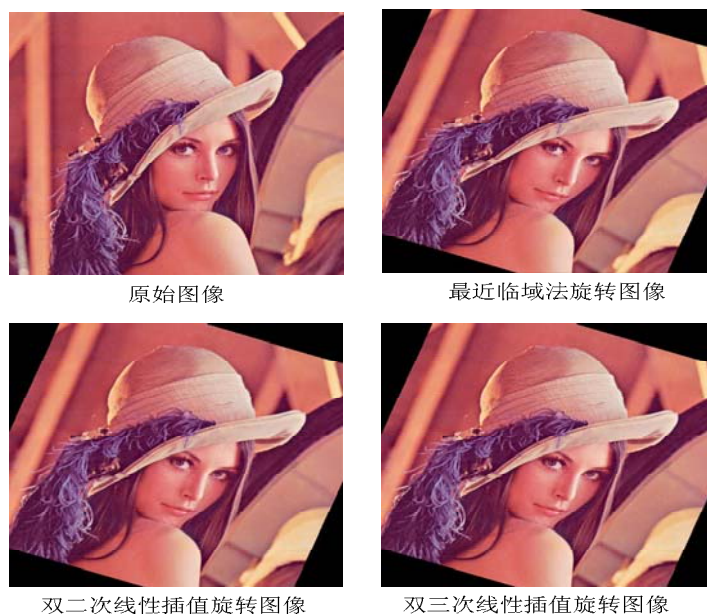


图 4 不同插值算法的旋转效果图

由以上几幅图可以看到，用最近邻法得到的图像质量最差，其图像上面呈现出明显的锯齿状纹理，特别在像素差别比较大的地方明显，用双三次插值法得到的图像纹理最清晰，质量最好，双线性插值法得到的图像的质量介于两者之间。

140

表 1 三种插值算法的比较

插值方法	性能特点
最近邻域法	处理时间最短，实时性比较好，但得到的图像质量不高，在旋转的过程中容易产生锯齿状边界。适用于要求旋转实时性高，但对图像质量要求不高的程序
双二次线性插值	相对最近邻插值法，其速度慢，但是得到的图像质量高，纹理较清晰，能满足大多数程序的要求。
双三次线性插值	处理速度最慢，但相应的得到的图像的质量最高，适用于对图像质量要求比较高的程序，一般很少使用。



因此本文中针对船舶的特殊环境，视频图像对实时性和图像质量都有要求，选择双二次线性插值作为图像旋转的处理算法，能够满足系统的需求。

### 2.1.2 图像放大算法

在上一节中介绍了图像的旋转算法，从给出的效果图中可以看出，当图像旋转的时候，整个图像会变大，而显示器屏幕是固定大小的，因此会留下没有图像内容的空隙，这样就不便于人们观察视频，于是我们需要将旋转后的图像进行放大，这样放大后的图像可以覆盖住以前的空隙部位，以形成一副完整的视频图像。

如图 5 所示，图形 ABCD 以 O 点为中心进行放大，得到图形  $A_1B_1C_1D_1$ 。点 A 映射到  $A_1$  位置，点 B 映射到  $B_1$ ，点 C 映射到  $C_1$ ，点 D 映射到  $D_1$ 。

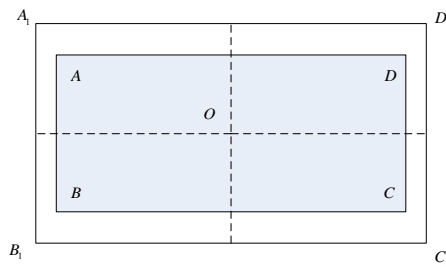


图 5 图像放大示意图

同样依据上节介绍的插值算法，考虑到系统的实时性，以及 DSP 处理器的性能等，本文在图像放大部分任然选用双线性插值算法，如图 6 所示为图像放大效果图。

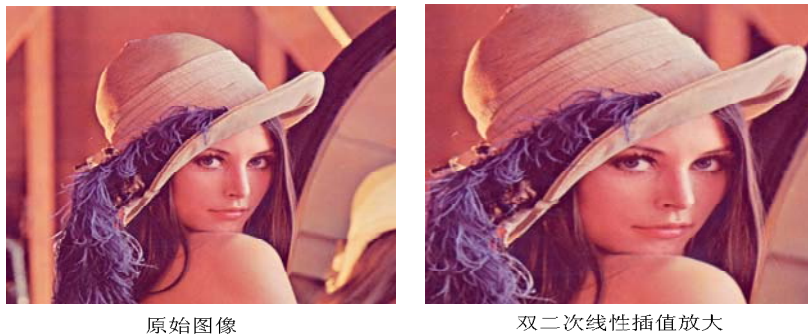


图 6 放大效果图

此放大图像是在横向放大 1.5 倍，纵向放大 1.3 倍的条件下放大的，可以观察到采用双线性插值算法放大图像的效果是比较好的，能够满足系统的需求。

### 2.1.3 图像平移算法

图像的平移实际上可以参考两个二维坐标系原点在 X 坐标、Y 坐标上的值不同,进行平移得到。如图 7 所示，图形 ABCD 横向平移 w 单位，纵向平移 h 单位，就得到了图形  $A_1B_1C_1D_1$ 。

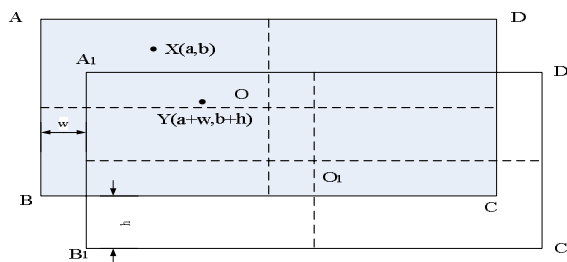


图 7 平移示意图

对于图像的平移算法，我们可以从上面的示意图中得出一个结论：在平移以前的图像 ABCD 中存在一点  $X(a,b)$  同平移以后的图像  $A_1B_1C_1D_1$  中的点  $Y(a+w,b+h)$  的像素值是一样的。我们可以将这两个点看作是相同点。将图像  $A_1B_1C_1D_1$  中的点  $Y(a+w,b+h)$  反向移动（横向  $w$  单位，纵向  $h$  单位）即可以得到图像点 ABCD 中的点  $X(a,b)$ 。

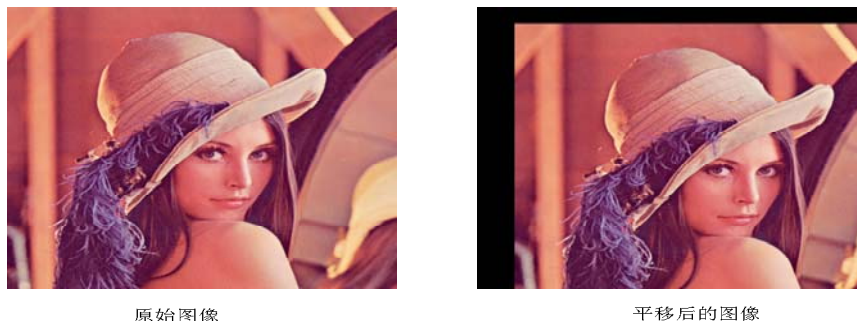


图 8 平移效果图

此平移效果图是横向平移了 50 个像素，纵向平移了 30 个像素，因为此平移算法没有对像素值作任何处理，只是进行了几何的移动，由此效果图可以看出，图像的质量没有任何降低。

## 2.2 软件开发环境的建立及设计

软件系统中选定 DSP / BIOS 作为稳像系统的开发系统。通过 DSP / BIOS 来调度任务，可更加合理的利用资源并且在程序中减少动态对象的建立，使得开发者可以将更多的精力投入到特定算法的开发上<sup>[8]</sup>。

在 DSP / BIOS 开发环境中，系统采用多线程来设计稳像程序，线程对应于独立的功能单元。在基于 DM6437 的视频稳像系统中，按数据流向大致可以分为初始化模块、视频采集模块、视频处理模块以及视频输出模块。系统中对初始化模块采用 SWI 线程，而对视频采集模块、视频处理模块及视频输出模块采用 TSK 线程以简化设计。在多线程下的系统原理示意图 9 所示：

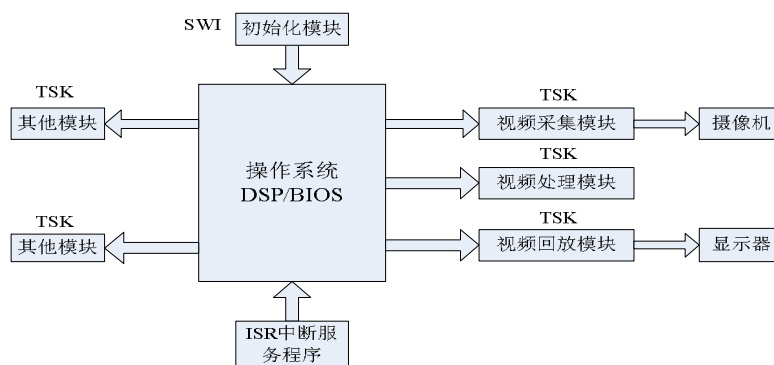


图 9 多线程系统原理图

其执行过程如下：首先进行 DSP / BIOS 系统的初始化；其后在系统中依次创建视频采集、视频处理及视频回放任务并安排其优先级，系统中为这三种任务安排相同的优先级；继而在系统中初始化这三种任务并启动它们，启动后的任务在后台执行；最后主程序进入 IDLE 状态等待其它任务或 ISR 中断程序的发生直至任务结束。该流程可以示意如下：

```
195 void main(void)
    {
    Sys_Init();          初始化
    Sys_TaskCreate(Task1); 创建视频采集任务并分配优先级
    Sys_TaskCreate(Task2); 创建视频处理任务并分配优先级
200 Sys_TaskCreate(Task3); 创建视频回放任务并分配优先级
    Sys_InitTask(Task1); 初始化视频采集任务
    Sys_InitTask(Task2); 初始化视频处理任务
    Sys_InitTask(Task3); 初始化视频同放任务
    Sys_StartTask(Task1); 启动视频采集任务
205 Sys_StartTask(Task3); 启动视频处理任务
    Sys_StartTask(Task2); 启动视频回放任务
    While( ; ; )
    {
    IDLE;                空等待
210 }
    }
```

DM6437 的程序代码使用 C 语言编写, 而且因为 DM6437 不支持浮点运算, 为了使处理器的效率最大化, 这里本文引入了 IQmath 库, 将原来的浮点数用整数来运算, 同时保证了数据的精度, 充分利用了处理器的能力。同时对代码进行了各种优化, 通过设计适合 DSP 的稳像程序, 以达到降低 DSP 运算量的目的, 还包括存储器的分配、减少内嵌循环次数、使用内联函数、使用汇编语句等<sup>[7]</sup>。

### 3 结论

文章设计了一种基于 DM6437 的电子视频稳像系统, 同时从算法级和程序级两个方面对稳像算法进行设计。阐述了稳像系统的系统组成、软件设计以及稳像算法, 对基于双线性插值算法作了详细说明。实验表明, 该系统具有很好的稳像效果, 具有较高的应用价值。

本系统具有精度高、实时性好等优点, 可应用于船舶设备中。此外, 系统丰富的接口资源及处理器强大的运算能力使得本系统可以进一步扩展, 并应用于其它相关领域。

#### [参考文献] (References)

- 225 [1] 曹辉. 船舶监控中的视频稳像与目标跟踪的研究[D].大连:大连海事大学,2008.  
[2] 赵菲. 视频稳像技术研究[D]. 长沙:国防科学技术大学,2007.  
[3] 陈硕. 视频图像的数字稳像系统设计 with 目标识别[D]. 西安:西安电子科技大学,2009.  
[4] 袁凤刚,刘建成. 不同插值方法实现数字图像旋转研究[J]. 软件导刊,2010 第 9 卷,第 4 期.  
[5] 姚敏. 数字图像处理 [M]. 北京:机械工业出版社,2006.
- 230 [6] KENNETH R.CASTLEMAN. 数字图像处理 [M]. 朱志刚. 北京:电子工业出版社,2002.  
[7] TMS320C6000 Optimizing C / C++ Compiler User's Guide[Z]. SPRU187k. Texas Instruments Incorporated,2003.  
[8] TMS320 DSP / BIOS User's Guide[Z]. SPRU423. Texas Instruments Incorporated,2002.