

基于多核计算的实时数字图像稳定

周渝斌¹ 时永刚²

(上海工程技术大学机械工程学院 上海 201620)¹ (北京理工大学信息科学与技术学院 北京 100081)²

摘要 多核计算已成为计算领域新的发展趋势。针对电子数字稳像系统中待处理的数据量和计算量都非常大的特点,结合双核 PC 平台的计算特点,采用了一种联合多种加快算法运行速度的处理方案,以满足系统的实时性要求。该方法结合图像视频数据处理的高度并行化、重复性处理的特征,在程序编制上,采用多线程并行处理方法,利用单指令多数据流(Single Instruction Multiple Data, SIMD)、流水线 SIMD 技术,在双核 CPU 上完成并行多线程任务分配和共享内存管理。并在快速搜索基础上,采用自适应多局部运动矢量估计消除运动物体干扰,用 Kalman 滤波分离了两种不同的运动信号,最终完成有害抖动信号的运动补偿。通过这些措施,在普通商用的双核 PC 机上实现了稳像实验系统的高效实时处理。

关键词 多核计算, 图像稳定, 运动估计, 并行处理

中图分类号 TP311.11 **文献标识码** A

Real-time Electronic Digital Image Stabilization System Based on Multi-core Computation

ZHOU Yu-bin¹ SHI Yong-gang²

(School of Mechanics Engineer, Shanghai University of Technology and Science, Shanghai 201620, China)¹

(School of Information Science and Technology, Beijing Institute of Technology, Beijing 100081, China)²

Abstract The multi-core computing has become the new trend of computer industry. Electronic Digital Image Stabilization (EDIS) system involves both large amount of data and intensive computation. The real-time implementation of EDIS system entails rapid data processe with high parallelism and repeated computation, which are suitable for data parallel processe of image sequences. The algorithm solution was proposed to deal with high speed image stabilization to meet the requirements of real-time systems. According to the data processe characteristics of EDIS, efficient C++ with inline SIMD (Single Instruction Multiple Data) and multi-thread programming were written and compiled on dual-core PC for real parallel running threads to achieve real-time performance in conventional PC. After the motion vectors acquired from fast search strategy, adaptive multi-local motion vectors avoided the moving objects interference. And the Kalman filter removed high-frequency image jitter motion while smooth global movements were retained for scanning and vehicle moving. The test results show that it is possible to implement a high efficient and robust real-time stabilization system on conventional PC with Multi-core at last.

Keywords Multi-core computation, Image stabilization, Motion estimation, Parallel processing

数字图像稳定是图像序列处理中一项重要的前处理步骤。如果待处理的图像序列拍摄于运动载体,载体非平稳运动可导致获得的视频序列间存在严重的帧间抖动。例如,在手持摄像^[1,2]、机器人视觉^[3]、车载摄像^[4]以及飞行器的监视观察^[5]等系统中,都存在这一问题。图像稳定技术的目的就是通过对运动补偿去除或尽量减少有害的抖动^[6],为后续处理提供稳定的图像序列。电子稳像技术集电子、计算机、数字信号处理等技术为一体来实现图像序列稳定,便于和图像拼接、图像增强、信息融合、目标追踪、目标识别等各种图像处理技术综合运用。在现场观察系统中,电子数字稳像必须能进行实时处理。另外,图像被稳定后还要进行目标探测、识别、跟踪等后续任务,也要求图像稳定处理必须高度实时化。但是

待处理的图像序列其图像数据量很大,特别是帧间运动矢量估计具有极大的计算量。这些特点使得完成实时的稳像需要高性能的硬件投资,对算法运行效率优化和程序优化都提出了很高的要求。

本文中的稳像实验使用块匹配进行运动估计,以菱形搜索与三步快速搜索相结合的搜索策略减少了计算量,使用帧间运动矢量预测进一步加快搜索速度,同时扩大了有效的搜索范围。使用自适应多区域运动估计可有效消除场景内的运动物体干扰。实验中还结合电子稳像系统的实际要求和视频图像数据处理高度并行化的特征,采用多线程任务管理,利用单指令多数据(SIMD)和流水线 SIMD 技术,最后在多核上进行多线程任务并行化分配的处理方法,充分发挥现有多核

到稿日期:2009-05-22 返修日期:2009-08-05 本文受上海高校选拔培养优秀青年教师科研专项基金(GJD08017)资助。

周渝斌(1973-),男,博士,讲师,主要研究方向为图像处理等,E-mail:choybeen@yahoo.com;时永刚(1970-),男,博士,副教授,主要研究方向为图像处理等。

PC 机处理多媒体数据的潜力,在普通的双核 PC 机上实现了系统的高效实时处理。

本文第 2 节介绍了电子稳像系统的构成及原理;第 3 节是运动矢量估计数学模型的建立和快速搜索方法;第 4 节介绍稳像算法整体方案;第 5 节为系统的软硬件实现部分,给出了多核系统上的具体实验结果;最后为结论。

1 系统构成及工作原理

电子稳像系统一般主要由 3 部分组成:运动矢量估计模块(ME)、运动矢量补偿模块(MC)和图像序列合成模块(IC)。运动矢量估计模块使用匹配准则,并通过一定的搜索策略,估计出图像序列中相邻两帧间的全局运动矢量,并将运动参数传给运动补偿模块。运动补偿模块根据运动估计参数,从全局运动矢量的反方向,将当前帧校正到正确位置,同时将补偿后的图像序列传送到图像合成模块。在图像合成模块中,根据运动估计的搜索范围,对图像序列进行剪切等处理,去除只在某些单帧图像中出现的场景,通过合成,生成场景连续、画面大小一致的稳定图像序列。由于存在剪切等处理,稳定图像序列的画面尺寸比摄像机采集到的画面尺寸有所减小,减小量等于运动估计的搜索范围大小。3 个组成部分中,运动矢量估计是系统中最为关键和重要的环节,不仅待处理数据量巨大,而且包含了密集的计算量,也是整个稳像算法实现的基础。运动矢量估计的搜索范围、匹配策略和鲁棒性等直接决定了系统的整体性能。运动矢量补偿和图像序列合成,相对而言则涉及到较小的计算量,实现简单。

2 帧间运动矢量估计

2.1 摄像机在 3D 空间运动模型和成像面的仿射运动模型

电子稳像系统中,摄像机的运动是一种刚体运动模式,有 6 个自由度,即沿坐标轴(X,Y,Z)的 3 个平移量($\Delta x, \Delta y, \Delta z$)和 3 个绕轴旋转量(φ, ψ, θ)。摄像机和场景之间的运动是一种相对运动,为简化处理,以摄像机镜头光心为原点,水平方向为 X 轴,垂直方向为 Y 轴,光轴方向为 Z 轴,建立坐标系{C},如图 1 所示。

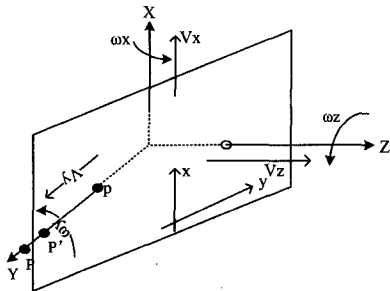


图 1 物体投影成像模型

在这一坐标系中,摄像机可看作是静止的,而运动全部来自场景。类似地在成像面建立图像坐标系{I}。在目标上任取一点 P 作为观察点,设采样周期为 T,则在 kT 时刻,点 P 在坐标系{C}中的坐标为 $(X_k, Y_k, Z_k)^T$,其在坐标系{I}中对应点 p 的坐标为 $(x_k, y_k)^T$;在 $(k+1)T$ 时刻,点 P 在{C}中坐标为 $P'(X_{k+1}, Y_{k+1}, Z_{k+1})^T$,在{I}中对应点 p 坐标为 $(x_{k+1}, y_{k+1})^T$ 。在笛卡尔坐标系,不同时刻 P 点中的仿射运动模型为:

$$\begin{pmatrix} X_{k+1} \\ Y_{k+1} \\ Z_{k+1} \end{pmatrix} = \begin{pmatrix} R_{xx} & R_{xy} & R_{xz} \\ R_{yx} & R_{yy} & R_{yz} \\ R_{zx} & R_{zy} & R_{zz} \end{pmatrix} \begin{pmatrix} X_k \\ Y_k \\ Z_k \end{pmatrix} + \begin{pmatrix} \Delta X \\ \Delta Y \\ \Delta Z \end{pmatrix} \quad (1)$$

其中, $R_{xx} = \cos\psi\cos\theta, R_{xy} = \sin\varphi\sin\psi\cos\theta - \cos\varphi\sin\theta, R_{xz} = \cos\varphi\sin\psi\cos\theta + \sin\varphi\sin\theta, R_{yx} = \cos\varphi\sin\theta, R_{yy} = \sin\varphi\sin\psi\sin\theta - \cos\varphi\cos\theta, R_{yz} = \cos\varphi\sin\psi\sin\theta - \sin\varphi\cos\theta, R_{zx} = -\sin\psi, R_{zy} = \sin\varphi\cos\psi, R_{zz} = \cos\varphi\cos\psi$ 。

而图像传感器 CCD 是平面的,经透镜成像后场景中的目标在摄像机的 CCD 成像面上,表现为 2D 运动。由于摄像机距离场景目标很远,相对 CCD 大小可以认为是无限远,绕(X,Y)轴的旋转在像面上表现不明显,其影响可忽略不计。此时,场景目标在成像面上的 2D 仿射运动模型为:

$$\begin{pmatrix} x_{k+1} \\ y_{k+1} \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x_k \\ y_k \end{pmatrix} + \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} \quad (2)$$

该模型中,只剩下图像平移和旋转运动,即帧间运动矢量只剩下(X,Y)方向平移运动和绕 Z 轴旋转。因此,稳像算法只要能估计出这 3 个参数即可。

2.2 基于块匹配的快速帧间运动矢量估计

帧间运动估计最常用的是块匹配法^[7-9]。块匹配法计算简单,实现方便,通过计算帧间最相似匹配位置得到两幅图像的位移,由帧间像素总绝对差值 SAD 作为匹配标准:

$$SAD = \sum_{i=1}^N \sum_{j=1}^N |I_{i,j} - I'_{i,j}| \quad (3)$$

其中, $I_{i,j}$ 和 $I'_{i,j}$ 分别为参考图像和当前图像(i,j)位置的像素强度。

块匹配的基本思想如图 2 所示。当某一位置处两个匹配块的 SAD 函数取得最小值时,该位置就是正确的匹配位置,而两帧图像匹配块之间的位置差就是帧间运动偏移矢量。

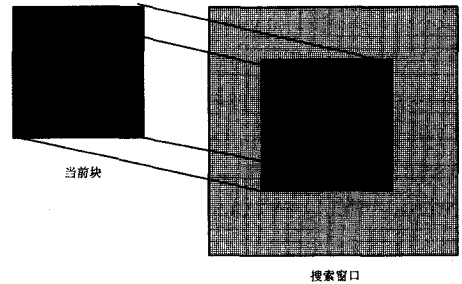


图 2 块匹配示意图

采用块匹配搜索,计算量必然很大,计算速度极慢。为了加快搜索速度,人们提出了很多快速搜索方案,如变步长方法、三步快速搜索法、对数搜索法、螺旋式搜索法、菱形搜索法、交叉搜索法、多分辨率策略、小波变换金字塔方法^[8]等。本次实验选用了改进的三步搜索法^[8],如图 3 所示。该方法并在三步法的基础上,结合了菱形搜索法和螺旋搜索法的特点,以菱形顶点作为初始搜索点。搜索时,从中心点向四周螺旋发散地进行搜索。一旦符合指定条件,停止搜索,记为初始匹配位置。如果菱形搜索找不到最佳位置,为避免进入局部最优,改菱形螺旋为矩形螺旋搜索。找到初始匹配位置后,进入搜索的第二步,以初始匹配位置点为中心进行三步搜索。这种算法比三步法计算量增加很少,通过螺旋式搜索,可以在很大程度上避免陷入局部极值。

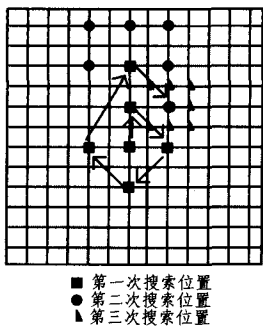


图3 改进的三步搜索法

3 图像序列稳定方案

获得帧间运动估计是进行图像配准的基础。但是在实际试验中,还要考虑多种因素才能获得稳定的视频序列输出。

3.1 自适应参考帧更新

参考帧更新要考虑到输出图像的的稳定效果和相机载体的运动范围。主要有两种参考帧更新方式:单一参考帧和实时参考帧。单一参考帧适合静态场景的稳像。如果图像序列的场景始终局限在一定的场景范围内,这种情况下,后续帧总能在参考帧中搜索到正确的运动矢量,采用图像序列的第一帧作为参考帧就可以了。但在实际应用中,相机成像场景很快就超出了运动搜索范围,单一参考帧显然不再有效。参考帧的实时更新虽然可以解决超出场景之外的运动矢量估计,却容易累积误差,稳像效果较差。本次试验采用了自适应参考帧更新,即局部单一参考帧。当运动估计超出该参考帧后,才更新参考帧,将当前帧与参考帧之间的中间帧作为新参考帧。

3.2 运动物体干扰处理

基于图像块进行运动估计时,图像块总取自图像中心。在实际的视频序列中,由于场景中存在各种运动物体,如随风摆动的树叶、行走的人、运动的车辆等,特别是当运动物体进入匹配画面中心时,稳像在在一定程度上就变成了目标追踪。而当运动物体进入或移出搜索范围时,在参考帧中又找不到相对的匹配位置而出现错误匹配。一种折中的方法是在画面的几个局部区域内先估计出运动矢量,然后从中挑出一个最可靠的值作为整个画面的全局运动矢量^[7-9]。文献[7,8]都是将图像分成4个区域,先在每个区域内进行局部运动矢量估计,然后用模糊判别法选出最终的全局运动矢量。文献[9]是将图像分为前景和背景区域,每次至少估计3个局部运动矢量。在得到局部运动矢量后,从中选择最优匹配者作为全局运动矢量。

本文在文献[7-9]的基础上,提出了一种自适应运动估计法,具体可分为以下几步:

1)首先将画面分成4个大小相等的较小块,并在画面中心进行运动估计。如果得到的SAD小于预先设定的阈值,认为该次估计结果正确,则按通常的估计方法,结束该次运动估计,继续进行运动补偿。如果得到的SAD大于预先设定的阈值,即为该次匹配不可靠,进入下一步。

2)将画面分成大小相等的4个小区域,依次在每个小区域内进行运动估计。

3)步骤2)中的每次估计完毕后,都按照步骤1)进行相似的比较,直至找到可靠的运动估计值。

该方法在画面中有运动物体时,能有效地避免运动物体

的干扰。如果画面中央没有运动物体,仍然只在画面中心进行一次运动估计。只有在最坏的情况下,才在所有的4个子画面中进行运动估计。因此,该方法的总体平均计算量比文献[7-9]中的方法都要小。

3.3 基于Kalman滤波原理的全局运动矢量估计

电子稳像系统中,摄像机运动是载体行进运动和随机抖动的叠加,运动估计矢量也相应地是这两种信号的叠加。通常,载体正常行进运动的频率低于1Hz,包括载体的前进运动和相机的扫描运动。而抖动信号由机械振动和路面颠簸造成,相对平滑运动来说是高频信号,一般在5Hz以上,具有随机性,可看作噪声。运动矢量是低频平稳信号和高频随机噪声信号的叠加,且相互独立。稳像的目的就是要找出并消除连续帧之间运动矢量包含的随机信号——抖动运动矢量。

低通滤波器可以过滤掉不期望的高频随机抖动噪声信号,但有时延,不适合实时计算,但Kalman滤波器的效果较好。Kalman滤波的迭代处理可以根据已有的观测值预测下一次的观测值,并通过新的观测值对估计值进行修正。因此,Kalman滤波具有预测功能,每次预测的当前运动矢量可用作快速搜索策略中的起始搜索中心位置,以进一步加快运动矢量估计过程。

在应用Kalman滤波估计理论时,必须首先建立关于所要解决问题对象的数学模型,即系统的状态方程和观测方程^[10,11]。本文结合相机载体的运动方式,设计了两种模型运动矢量的状态转移方程。

3.3.1 匀加速运动模型

图像序列中每帧图像的全局运动矢量反映了当前帧与参考帧之间的空间位置变化,根据式(2)可以算出当前帧的瞬时速度。对匀加速运动模型来说,下一帧的水平运动速度 v_x 为前一帧的水平速度与水平方向加速度和采样周期乘积 $a \times T$ 之和。垂直方向的速度也是如此。由此得出匀加速运动模型的状态转移方程^[11]:

$$\begin{bmatrix} v_x(k) \\ v_y(k) \\ a_x(k) \\ a_y(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x(k-1) \\ v_y(k-1) \\ a_x(k-1) \\ a_y(k-1) \end{bmatrix} + [w] \quad (4)$$

$w(k) = [0, 0, u_1(k), u_2(k)]$ 是 x, y 方向加速度变化的扰动, $u_1(k), u_2(k)$ 相互独立,序列的每一帧都可以根据式(4)计算。令 mv_x 和 mv_y 为 kT 时刻相邻两帧图像在水平和垂直方向运动矢量的观测值,得到观测方程:

$$\begin{bmatrix} mv_x(k) \\ mv_y(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_x(k) \\ v_y(k) \\ a_x(k) \\ a_y(k) \end{bmatrix} + [v] \quad (5)$$

$v(k) = [v_1(k), v_2(k)]$ 是具有零均值的高斯白噪声。

在电子稳像系统的匀加速运动模型里,每一帧的速度修正矢量 $V_{cor}(k)$ 为:

$$V_{cor}(k) = V_{klm}(k) - V_{act} + V_{cor}(k-1) \quad (6)$$

其中, $V_{klm}(k), V_{act}(k), V_{cor}(k-1)$ 分别为Kalman滤波估计值,根据式(2)经运动估计得到的实际运动矢量,即前一帧的修正矢量。

3.3.2 匀速运动模型

在匀速运动模型中,以每一帧相对于第1帧的绝对空间

位置作为差分运动矢量,其数值等于当前两帧的位置差与前面累积运动矢量之和。实际上,可以通过帧的空间位置确定摄像机的瞬时位置,也可以通过帧间运动矢量确定摄像机的瞬时速度。这一模型中,设参考帧位置为0点,以后每一帧的绝对位置都是前一帧的位置加上当前的帧间运动矢量。这样,可以得到匀速运动模型的状态方程:

$$\begin{bmatrix} p_x(k) \\ p_y(k) \\ v_x(k) \\ v_y(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & \tau & 0 \\ 0 & 1 & 0 & \tau \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x(k-1) \\ p_y(k-1) \\ v_x(k-1) \\ v_y(k-1) \end{bmatrix} + [w] \quad (7)$$

$p_x(k), p_y(k)$ 分别表示第 k 帧在水平和垂直方向的空间位置。建立观测系统的方法有两种,既可以通过观测帧间运动矢量来建立,也可以通过累积前面帧的运动矢量估计当前帧的绝对位置来建立。为了统一这两种模型,用帧的绝对位置建立观测方程:

$$\begin{bmatrix} mp_x(k) \\ mp_y(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} p_x(k) \\ p_y(k) \\ v_x(k) \\ v_y(k) \end{bmatrix} + [v] \quad (8)$$

建立上面的状态转换方程和观测方程后,也就确定了估计值和观测噪声变量,加入 Kalman 滤波进行实时滤波,可以估计出状态的变化。匀速全局运动模型直接平滑了每帧的坐标后得到稳像后的输出坐标。Kalman 滤波位置直接给出了每一帧相对于前一帧的坐标位置,因此,任一帧的速度修正矢量 $V_{cor}(k)$ 为:

$$V_{cor}(k) = V_{kln}(k) - V_{act}(k) \quad (9)$$

其中, $V_{kln}(k), V_{act}(k)$ 意义与式(6)中相同。

图4是 Kalman 滤波的运动矢量效果。滤波后的运动矢量曲线平滑,不仅没有了剧烈的抖动,同时保留了载体的运动特性。在颠簸处,有效地消除了颠簸运动,而人为的快速扫描运动却被保留了下来。从监视器上观察,图像序列保持了比较平滑的运动状态过渡,没有明显的跳跃。

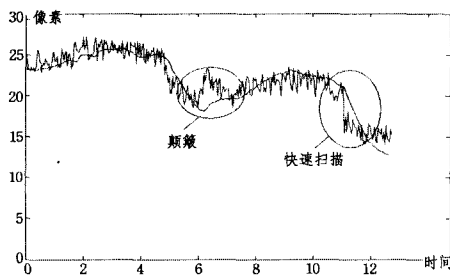


图4 Kalman 滤波的效果

4 运动估计程序的优化与实现

本次试验在基于多核 CPU 的 PC 上进行,采用 VC2005 集成环境下开发的应用软件系统。整套试验装备包括高速 CCD 摄像机、图像采集卡、仿真振动平台、监视显示器。带有高速快门的 CCD 摄像机安装在振荡平台上,对景物进行拍摄。为了保证相机在高速运动时图像的清晰度,快门设定为 $1/5000s$ 。

4.1 系统任务的多线程并行处理

使用 Windows 多线程程序设计方法,可以在一个进程中同时处理多个任务。电子稳像系统主要的处理任务有:(1)

图像序列数据的采集和传输;(2) 帧间运动矢量估计;(3) 运动补偿;(4) 图像序列合成视频数据的输出。

采用单线程编程方法,这些处理任务只能按顺序依次实现,不仅可能会造成处理任务之间发生冲突,而且无法满足实时运行。采用多线程,就可以将不同的处理任务分配到多个线程里去并行执行,从而大大提高程序的运行效率,满足实时处理要求。在数据采集线程中,为避免数据处理线程对数据进行访问,在计算机内存里设计两个反转数据缓冲区域,进行乒乓数据读写。当图像采集线程向一个内存区中写图像数据时,用互斥锁对其进行锁定。存储结束后,写线程解除锁定标志,并传递给另一个缓冲区。这时读线程可以对第1个区域的数据进行读处理,而第2个区域则正在接受写线程的数据。如此循环反复。

4.2 系统数据的 SIMD 并行优化处理

SIMD 是一种单指令多数据技术,在基于 DSP 的程序设计中有着广泛的应用^[9],能在一个指令周期内同时处理多个数据,通过并行处理,提高数据的吞吐量。Intel 公司先后在 Pentium 系列中引入的 MMX, SSE, SSE2 和 SSE3 等 SIMD 多媒体处理指令集,在许多方面得到了应用^[12-14],也得到了 AMD 公司的全力支持。MMX 可以同时处理 8 个 8 位的整数进行操作, SSE 可以同时处理 4 个 32 位的浮点数进行操作,而 SSE2 可以同时处理 16 个 8 位的整数进行操作,而且对数据存取进行了优化。

在实时的电子稳像系统中,运动估计部分虽然只占不到程序代码的 10%,却要占用多达 80%~90% 的执行时间。根据运动估计的匹配过程中数据处理高度并行、重复计算等特点,通过在程序中嵌入并行化 MMX 和 SSE 汇编指令进行混合编程,加快了程序的执行速度。CPU 内部 MMX 有 8 个 64 位整数寄存器 mm0-7,而每两个 8 字节数的差分绝对值需要用 4 个 mm 寄存器,故每次循环内部可做两次两个 8 字节对齐加减。用 C, MMX 技术进行运动估计的运算速度结果如表 1 所列。本次实验采用全搜索法,搜索范围是 128×128 ,匹配块大小是 32×32 。测试平台为单核 PC,主芯片为 Intel Celeron 2.6G CPU, 2GDDR400 内存,图像帧频为 30 帧/s。表 1 至表 4 中的测试结果都是基于该平台。

由表 1 可见, SIMD 技术对运算速度的提高非常明显,由于每次循环内部都做了 16 个字节数的减法,其速度虽然没有提高 16 倍,但也提高了 11 倍多。

表 1 SAD 计算速度对比(ms)

序号	C 执行时间	SSE 执行时间
第 1 次	1633	140
第 2 次	1592	141
第 3 次	1612	150
第 4 次	1622	143
第 5 次	1603	149
平均值	1612	145
速度比	1:11.1	

在图像进行旋转时,需 SSE 指令进行浮点运算。根据计算量和插值效果,本次试验采用了双线性插值^[15]。实验中,采用 SSE 技术每次计算 4 个点来提高运算速度。旋转角范围为 $\pm 7^\circ$,步长为 0.2° 。由于图像旋转的步长和角度范围具有规律,预先算好这些点旋转后的新位置值,作为文件保存,程序启动时读入内存。计算的时候采用直接查表的方式。表

2 列出了用 C、Intel 汇编、SSE 技术对 400×300 大小的 24 位色图像进行旋转时的耗时对比。表 2 数据显示,汇编编程(Assemble)比 C 大约提高了 15%,SSE 比 C 大约提高了 30%,运用查表技术计算速度进一步提高了 30%左右。

表 2 图像旋转计算速度对比(ms)

序号	C 程序执行时间		汇编执行时间		SSE 执行时间	
	查表前	查表后	查表前	查表后	查表前	查表后
1	731	551	610	490	531	401
2	722	550	621	491	541	390
3	721	573	620	501	525	392
4	732	561	631	491	533	401
5	713	551	614	493	540	410
平均值	724	557	619	493	534	399
速度比	1.3 : 1		1.256 : 1		1.34 : 1	
查表后 3 种程序执行速度比:1 : 1.23 : 1.45						

为提高运动估计的精度,进行亚像素插值,用二维插值法进行图像放大。实验中选用两维多项式插值法进行图像缩放,算法中主要运用了浮点乘除法和平方运算。表 3 中列出了分别用 C 和 SSE 对图像进行缩放的速度对比。

表 3 图像缩放计算速度对比(ms)

序号	C 执行时间	SSE 执行时间
第 1 次	636	163
第 2 次	642	175
第 3 次	633	160
第 4 次	612	152
第 5 次	598	147
平均值	624	159
速度比	1 : 3.9	

表 4 给出了二维情况下的运动估计速度,使用 C,MMX,SSE2 等不同实现方法所得到的运动矢量估计所需时间的对比。搜索范围是 64×64 ,匹配块大小是 16×16 ,角度旋转范围是 $\pm 3^\circ$,步长为 0.2° 。表 4 中实验结果为 50 次计算的平均值,结果表示为 $\mu \pm \sigma$, μ 为均值, σ 为均方差。从表 4 中的实验结果可以看出,采用 SSE2 后的速度提升了将近 7 倍。

表 4 不同实现方法的运动估计时间(ms)

C 程序执行时间	MMX 程序执行时间	SSE2 程序执行时间
21.5 ± 1.18	4.73 ± 1.04	3.37 ± 1.71
速度比 1 : 1.4 : 6.4		

4.3 多核下的并行处理

多核处理器是单个半导体处理器上拥有两个或多个具有同样功能的处理器核心。多核的出现,将多线程程序变成真正的“并行运行”。在单核芯片上,用时间片分割的方式切换多个线程运行,从而在表面看起来是同时运行。而多核的出现完全改变了这种局面。以双核为例,如果程序中只有两个处理任务,芯片中的每个 CPU 核运行一个任务,不需要进行时间分片,两个任务做到真正的同时并行运行,从而大大提高了运算效率,特别适合图像处理等大容量并行数据处理的应用^[16-18]。

稳像过程的 4 个步骤对应了 4 个线程,而其中的运动估计线程计算量最大,占总时间的 80%左右。如果将任务量平均分配到两个核上,即需要将运动估计线程中的部分计算量分配到另外一个核运行,尽量让两核计算任务平衡。运动估计过程可分为图像块旋转、像素级运动估计、亚像素估计 3 步,而图像块旋转和亚像素估计占运动估计总计算量的 1/5

左右,可将该部分任务和其他 3 个线程分配到内核 1 上运行,像素级运动估计分配到内核 2 上运行。为了加快数据吞吐,采用了 16×16 的图像块^[16]。内核 1 将图像块放大后,旋转所有角度。完成计算后,将数据放入共享内存,供内核 2 随后直接读取,以进行运动估计。

多核编程需要 OpenMP(Open Multi-Processing)支持,主要难点在于多个线程在不同核上的并行深度和负载均衡的分配^[17]。OpenMP 是一个支持共享存储并行设计的库,特别适合多核 CPU 上的并行程序设计。简单的方法是直接用开发工具 VC2005 提供的配置完成。在项目属性对话框中,左边框里的“配置属性”下有个“C/C++”项,在其“语言”项里,将 OpenMP 支持改为“是/(OpenMP)”就可以支持 OpenMP 了。这种方法虽然简单,但只支持标准的 fork/join 式并行模式。即程序中的串行部分都由主线程执行,并行的部分是通过派生其他线程来执行。当并行部分没有结束时,是不会执行串行部分的。因此,用这种模式,虽然计算密集的运动估计部分自动分配到双核上并行计算,但运动估计和其他线程间的逻辑顺序不能颠倒,仍然是串行方式运行。为了让程序运行更加有效率,必须调用 OpenMP 指令,分配各线程的并行运行方式。采用两帧并行运行的解决方式是,当内核 1 计算第 n 帧的运动矢量补偿时,内核 2 正在进行第 $n+1$ 帧的运动估计。计算完后,交内核 1 进行运动补偿。这时,如果内核 1 已经完成第 $n+2$ 帧的图像读取与块旋转,立即交内核 2 进行运动估计,然后开始第 $n+1$ 帧的运动补偿,如图 5 所示。

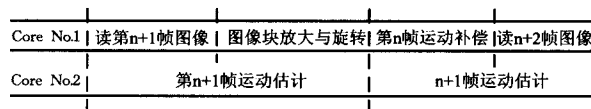


图 5 双核并行处理流程

表 5 是本次综合实验结果。本次实验的计算机其 CPU 为 Intel Pentium Dual CPU,型号为 E2180@2.0GHz,图像帧频为 60 帧/s。实验中,图像为大小 640×480 像素、256 灰度级、相邻两帧在水平和垂直方向上偏移量达到 $0 \sim \pm 64$ 像素,角度旋转范围是 $\pm 7^\circ$,步长为 0.2° 。从表中的数据可以看出,启动 OpenMP 后,速度有所提高,但是有限。而使用 OpenMP 命令将多线程分配到多核上并行处理后,执行速度提高高达 56%左右。

表 5 使用 OpenMP 前后的执行速度(fps)

不使用 OpenMP	打开 OpenMP	使用 OpenMP 命令
33.21 ± 2.75	37.42 ± 5.23	51.81 ± 3.05

结束语 电子数字稳像系统的数据量和计算量都非常之大,其实时性的实现对算法和数据的处理速度都提出了很高要求。本文采用运动矢量预测估计相结合的改进的快速搜索策略,加快了搜索计算速度。使用 Kalman 低通滤波方法去除最后输出的帧间运动矢量中的高频抖动成分,保持了图像序列的平滑输出。同时,结合系统图像数据处理高度并行化、重复处理以及数据对齐的特征,在程序编制上,采用多线程处理方法,利用单指令多数据流、流水线 SIMD 技术和多核下的多线程并行处理方法,大大提高了系统的运行效率。通过这些措施,在普通双核 PC 机上实现了系统的高效实时处理。

(下转第 277 页)

增大,PSNR 值有减小的趋势。低噪声水平的 PSNR 值下降趋势要快于高噪声水平。在相同的 k 值下,噪声标准差水平高的降噪增强图像的 PSNR 值要低于噪声标准差水平低的降噪增强图像。当 k 值大于 1.5 时,4 种噪声水平下的 PSNR 值下降幅度都比较明显,表现在图像视觉效果上就是噪声分布非常明显,这说明 k 值不能过大,否则会使按照 MAP 准则降噪遗留下来的噪声系数的增强幅度过大。

在噪声标准差水平 50 下,针对图像 House 绘制增强系数 $k=0.5,1,1.5,2,2.5,3$ 时增强 MAP 降噪图像与 MAP 降噪图像在第 160 行从第 150 列至 250 列的灰度值比较曲线,如图 4 所示。图中实线代表 MAP 图像灰度曲线,空心方框轨迹表示增强 MAP 图像曲线。从中可以看出,在 $k=0.5,1$ 时,增强 MAP 与 MAP 灰度值差异不大;从 $k=1.5$ 开始两者间的灰度值差异开始加大,到 $k=3$ 时,差异达到最大。灰度值差异越大,说明增强 MAP 降噪图像的 PSNR 值下降的幅度越大。

总之, k 值越小,图像增强的视觉效果越不明显; k 值越大,噪声会过度放大,同样使得视觉效果下降。在实际应用中,应该根据图像的特点合理选择增强系数 k ,以便在峰值信噪比和图像细节增强之间找到平衡。

结束语 本文针对红外成像易受噪声干扰造成图像模糊、分辨率低、对比度弱等特点,提出了带增强效果的基于最大后验概率准则的小波域自适应降噪算法。该方法通过考虑不同尺度和不同方向能量因素,对系数萎缩因子进行修正,使得在保证较高 PSNR 值的基础上,图像细节和边缘得到不同程度的增强。试验结果表明,本文提出的方法可以得到信噪

比明显提高、细节突出、对比度提高的图像,缓解了图像降噪和边缘细节模糊的矛盾,获得了较好的视觉效果。如果利用小波系数根据尺度间的相关性进一步滤除基于最大后验概率准则降噪遗留下来的噪声后再进行增强,便会获得更好的处理效果。本文提出的小波系数增强方法可以应用于基于其它降噪准则的比例萎缩算法中。

参考文献

- [1] 谢杰成,张大力,徐文立.小波图像去噪综述[J].中国图象图形学报,2002,7(3):209-217
- [2] 李旭超,朱善安.小波域图像降噪概述[J].中国图象图形学报,2006,11(9):1201-1209
- [3] 潘泉,孟晋丽,张磊,等.小波滤波方法及应用[J].电子与信息学报,2007,29(1):236-242
- [4] 熊江.小波域图像降噪概述[J].计算机科学,2007,34(7):232-234
- [5] Donoho D L,Johnstone I M. Ideal spatial adaptation via wavelet shrinkage[J]. Biometrika,1994,81(3):425-455
- [6] Donoho D L. Denoising by soft-thresholding[J]. IEEE Transactions on Information Theory,1995,41(3):613-627
- [7] Shark L K, Yu C. Denoising by optimal fuzzy thresholding in wavelet domain[J]. Electronics Letters,2000,36(6):581-582
- [8] Malfait M, Roose D. Wavelet-based image denoising using a Markov random field a priori model[J]. IEEE Transactions on Image Processing,1997,6(4):549-565
- [9] 王文,芮国胜,王晓东,等.图像多尺度统计模型综述[J].中国图象图形学报,2007,12(6):961-969

(上接第 273 页)

参考文献

- [1] Haruhisa O, Manabu H, Kazuhiko S, et al. Optimum selection algorithm of motion estimation blocks for fast and robust digital image stabilization[C]//Proc. IEEE International Conference on Consumer Electronics. Los Angeles, CA, USA, June 2003:396-397
- [2] Kim P. Image sequence stabilization scheme using FIR filtering [J]. International Journal of Control, Automation and Systems, 2003,1(4):515-519
- [3] Kurazume R, Hirose S. Development of image stabilization system for remote operation of walking robots[C]//Proc. 2000 IEEE International Conference on Robotics & Automation. San Francisco, CA, USA, April 2000:1856-1861
- [4] Jin J S, Zhu Z, Xu G. A stable vision system for moving vehicles [J]. IEEE Trans. on Intelligent Transportation Systems,2000,1(1):32-39
- [5] Carmona C. Vegetation L-band image processing system, geometric performances and spatio-temporal stability[J]. International Journal of Remote Sensing,2004,25(9):1769-1777
- [6] Duric Z, Rosenfeld A. Image sequence stabilization in real time [J]. Real-Time Imaging,1996,2(5):271-284
- [7] Marc C, Purnendu S. Survey of motion estimation techniques for video compression[C]//Proc. of SPIE. vol. 4796, Seattle, WA, United States,2002:218-226
- [8] 周渝斌.数字图像稳定算法研究和系统设计[D].北京:北京理工大学,2004
- [9] Derby J H, Moreno J H. A high-performance embedded DSP

- core with novel SIMD features[C]//Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (IC-ASSP). Hong Kong, April 2003:301-304
- [10] Brown R G, Hwang P Y C. Introduction to Random Signals and Applied Kalman Filtering (2nd edition) [M]. New York: John Wiley & Sons,1992:190-288
- [11] Ertürk S. Real-time digital image stabilization using Kalman filters[J]. Real-Time Imaging,2002,8(5):317-328
- [12] Cheresiz D, Juurlink B, Vassiliadis S, et al. Architectural support for 3D graphics in the complex streamed instruction set[J]. International Journal of Parallel and Distributed Systems and Networks,2002,5(4):185-193
- [13] Ahmad I, Yeung D K, Zheng W, et al. Software based MPEG-2 encoding system with scalable and multithreaded architecture [C]//Proc. of SPIE: Commercial Applications for High-Performance Computing. Denver, Colorado USA, August 2001, 4528:44-49
- [14] Servetti A, Rinotti A, De Martin J C. Fast implementation of the MPEG-4 AAC main and low complexity decoder[C]//Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing. Montreal, Canada, May 2004,5:249-252
- [15] Press W H, Teukolsky S A, Vetterling W T, et al. Numerical Recipes in C: The Art of Scientific Computing[M]. Cambridge, U. K: Cambridge University Press,1999
- [16] 张鹏,杜建国,解晓东,等.一种基于多核流水的多标准视频编解码器体系结构[J].计算机研究与发展,2008,45(ii):1985-1993
- [17] Rasmussen M S, Stuart M B, Karlsson S. Parallelism and Scalability in an Image Processing Application[J]. International Journal of Parallel Programming (JournalSeek), Apr. 2009
- [18] Terboven C, Dieter M, Schmid D. First Experiences with Intel Cluster OpenMP [C]//International Workshop on OpenMP. LNCS 5004. 2008:48-59