

一种消除图像拼接痕迹的新方法

宋宝森 付永庆 宋海亮
(哈尔滨工程大学 哈尔滨 150001)

摘 要 图像融合是消除拼接痕迹的关键技术之一,已有的众多图像融合算法大都是针对完全重合图像的,忽略了针对图像拼接痕迹消除的图像融合方法的研究,从而影响了图像拼接技术的应用发展。为了高效、高质量地完成图像拼接,利用缝合线搜索技术结合小波多分辨率分析融合技术,给出了一种消除图像拼接痕迹的新方法。实验结果表明,该方法在保证较好实时性的同时,高质量地消除了图像拼接的痕迹。

关键词 图像拼接,图像融合,缝合线搜索,拼接痕迹消除

中图分类号 TP391.7 文献标识码 A

New Efficient Image Fusion Algorithm for Image Mosaic

SONG Bao-sen FU Yong-qing SONG Hai-liang
(Harbin Engineering University, Harbin 150001, China)

Abstract Image fusion is one of key technics of image stitch tracer eliminate, multitudinous existing algorithms of image fusion are aimed at complete overlap images, but ignore the research for algorithms of partially overlap image fusion for image stitching applications, thus affecting the application and development of image stitching technology. In order to efficiently, high-quality accomplish image stitching, we presented a new method of image stitch tracer eliminate using technology of searching suture silk combined with wavelet multi-resolution analysis. Experiment shows the method not only has a preferable real-time, but also has high-quality result of stitch tracer eliminate.

Keywords Image stitch, Image fusion, Suture silk searching, Stitch tracer eliminate

1 引言

图像拼接痕迹消除的关键技术是图像融合,目前图像融合技术主要分为像素级、特征级和决策级 3 个层次,多数应用场合都需要进行像素级融合,原因是像素级融合能尽可能多地保留场景的原始信息。其实现方法主要有基于颜色空间变换融合方法(HIS 空间融合^[2]、基于神经动力学融合等^[3])、基于数学/统计学融合方法(加权平均融合、PCA 融合^[4]、期望最大值融合等)、基于多分辨率分析融合方法(金字塔分解融合^[5]、小波变换融合^[6]、脊波变换融合^[7]、曲波变换融合^[8]、Contourlet 变换融合等^[9])。在这些方法中,基于颜色空间变换方法的局限性在于要求亮度分量与替换亮度分量之间有较大相关性,但实际应用中多数场合得不到满足;基于数学/统计学方法需要一些先验信息(实际应用中很难得到)或者近似计算(使图像失真过多);基于多分辨率分析融合性能最优越,其融合是在图像的不同尺度、不同空间分辨率和不同分解层分别进行,使融合效果获得明显改善,但是目前的几种多分辨率分析方法除小波变换外计算都相对复杂,很难满足实际应用中的实时性需求,本文利用小波多分辨率分析技术结合缝合线搜索提出一种用于图像拼接痕迹消除的新方法,既保证

了较好实时性,又使拼接后的融合效果很好。

目前图像融合的主要对象有以下几种:多传感器图像之间融合^[1]、多聚焦图像之间的融合^[9]、不同波段遥感图像之间的融合^[3]、多光谱图像与全色图像的融合^[8]、SAR 图像与可见光图像的融合^[6],它们有一个共同点,即它们融合的两幅图像是完全重合的,这与图像拼接的图像融合有很大的不同。图像拼接中,图像之间只有小部分重合,不但要找到缝合线,且过渡区域的融合规则要合理选择,否则将拼接失败或者拼接处有明显的“拼接缝”。本文针对此问题,利用小波变换,将图像分割成不同频带的子图像,搜索缝合线后,在各子图像对应的过渡区采用不同的融合规则,在保证融合效果的情况下,高效地消除了拼接的痕迹。

本文第 2 节回顾了经典小波变换图像融合过程;第 3 节详细描述了我们的用于图像拼接痕迹消除的方法;第 4 节给出了仿真验证结果与讨论;最后总结了本文的工作。

2 经典小波变换的图像融合过程

基于小波变换的图像融合主要有 3 个步骤:

1) 对参加融合的图像分别进行小波分解,建立图像的小波金字塔图像。

到稿日期:2010-03-17 返修日期:2010-06-18 本文受水下智能机器人技术国防科技重点实验室开放课题研究基金(200736)资助。

宋宝森(1982-),男,博士,主要研究方向为信号与信息处理,E-mail:hrbeusongbs@163.com;付永庆(1956-),男,硕士,教授,博士生导师,主要研究方向为微弱信号检测、混沌通信及信号处理、认知无线网络、图像编码、拼接及特征提取、电路与电子系统;宋海亮(1984-),男,博士,主要研究方向为信号与信息处理。

2)对各分解层的小波系数根据不同的融合规则分别进行融合处理,形成融合后的小波分解金字塔图像。

3)通过对融合后的小波分解金字塔图像进行小波逆变换来重构图像,形成融合后的图像。

图 1 中的 LL,HL,LH,HH 为小波变换后每个分解层的 4 幅子图像,LL 为原图像中的主要低频成分,HL 为垂直方向的高频信息,LH 为水平方向的高频信息,HH 为 45 度方向的高频信息。不同分解层之间的子图像为原图像的不同频率的带通子图像,在各带通子图像中使用的融合规则是决定融合后图像质量的关键,也是研究重点。

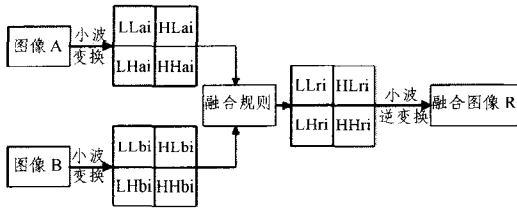


图 1 基于小波变换的图像融合过程

3 用于图像拼接痕迹消除的新方法

由于用于拼接的图像重合区域较小,使得融合的过渡区域相应也小,需要先确定缝合线,并在缝合线附近选择融合过渡区域 T ,然后在小波变换域进行融合,经过逆变换形成最后的融合图像。以上的过程即为提出的图像拼接痕迹消除新方法,简称 WT-Fusion。

3.1 图像小波变换

为了保证算法的实时性,本方法使用了经典的 Haar 小波基对配准后的两幅待融合图像分别进行小波变换,再经过数值规划后得到两幅原图的小波域系数。

这里使用的变换参数为:

- 1)支撑长度为 1。
- 2)尺度函数系数为: $\begin{cases} \sqrt{2}/2 \\ \sqrt{2}/2 \end{cases}$ 。
- 3)变换层数:3 层。

3.2 过渡区域确定

3.2.1 缝合线搜索

经过配准后的两幅图关系如图 2 所示,其中粗线包围的部分为重合区域,缝合线和过渡区域必须在重合区域选择。

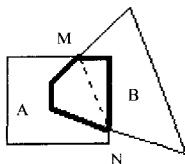


图 2 配准后两幅图的示意图

由于本文目的是在保证融合效果的情况下,尽可能提高融合的实时性,因此我们提出一种简单有效的方法来选择缝合线,具体地说就是通过两幅图的公共边界点(图 3 中 M 和 N 点)的直线作为缝合线,找到公共边界点 M 和 N 后,过 M 和 N 二点便可确定这条直线 MN 的位置。缝合线搜索可按如下两步完成:

1)搜索公共边界点

利用下式判断当前点是否为公共边界点:

$$F_{ij} \in \begin{cases} Y, & \forall A_{ij}^{\wedge} \in \phi_A \text{ 且 } \forall B_{ij}^{\wedge} \in \phi_B \\ E, & \text{其他} \end{cases} \quad (1)$$

式中, F_{ij} 代表两幅图像 A, B 经过配准后,拼接合成一幅图像对应的像素点, Y 代表公共边界点集合, E 代表非公共边界点集合, A_{ij}^{\wedge} 代表 F_{ij} 在图像 A 中对应的有效像素点, B_{ij}^{\wedge} 代表 F_{ij} 在图像 B 中对应的有效像素点, ϕ_A 代表图像 A 的边界点集合, ϕ_B 代表图像 B 的边界点集合。

2)计算缝合线

得到两个公共边界点后,很容易确定相应的直线方程:

$$y = ax + b \quad (2)$$

式中, $a = \frac{y_A - y_B}{x_A - x_B}$ 为斜率, $b = y_A - ax_A$ 为 y 轴截距,再由式(2)的直线方程得到缝合线的全部点的集合。在利用直线方程计算缝合线点集时须注意以下问题:

因为计算机内的坐标都是整数,在利用某一坐标集(如两点 x 坐标差的集合)循环计算另一坐标集(如 y 坐标差的集合)时,一定要用点数多的坐标集来计算点数少的坐标集,具体解释:设有两点 $p(x_1, y_1), p(x_2, y_2)$,当 $(x_2 - x_1) > (y_2 - y_1)$ 时,如果我们利用 y 来计算 x ,那么肯定得不到全部点集(y 连续但 x 不连续),所以我们要用 x 来计算 y ,同样,如果 $(y_2 - y_1) > (x_2 - x_1)$,我们就要用 y 来计算 x 才可以得到全部点集。

利用以上规则我们对图 6 中的两幅原图配准后进行搜索,得到的缝合线如图 3 所示,图中左边的图像稍暗,右边的图像稍亮,白色线条即为搜索到的缝合线。

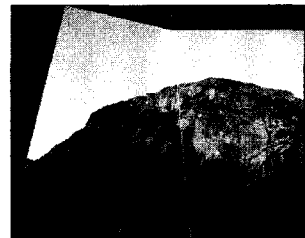


图 3 配准后找到缝合线的图像

3.2.2 缝合线多尺度处理

由于小波变换后,每层的子图像大小都是上层子图像大小的 $1/4$,因此我们要把缝合线也缩到相应大小。具体实现公式为:

$$p_{ij}^l(x, y) = p_{i/2^l, j/2^l}(x/2^l, y/2^l) \quad (3)$$

式中, $p_{ij}^l(x, y)$ 代表第 l 层子图像缝合线像素点, $p_{i/2^l, j/2^l}(x/2^l, y/2^l)$ 代表原始图像缝合线像素点坐标以 2 的 l 次幂为间隔,同时 x, y 都除以 2 的 l 次幂。经过多尺度处理后的缝合线如图 4 所示。



图 4 对应 3 层小波变换的缝合线多尺度化结果

3.2.3 过渡区域选取

用于拼接的融合可以由图 5 来表示^[5],为便于解释,这里

将问题简化为一维。

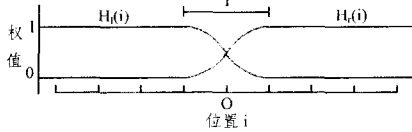


图5 融合示意图

过渡区长度为 T , 那么两幅图像的融合可以表示为:

$$F(i) = H_L(i - \hat{i})F_L(i) + H_R(i - \hat{i})F_R(i) \quad (4)$$

式中, $F_L(i)$ 和 $F_R(i)$ 是左右两幅图像的像素, \hat{i} 代表过渡区中心点(如图3中的 O 点), H_L 和 H_R 是加权函数, 且

$$H_R(i) = 1 - H_L(i) \quad (5)$$

可以看出 T 的选取非常重要, 如果想平滑过渡, 则 T 要与过渡区最大主要特征大小相当; 另一方面, 要想避免“鬼影”现象, T 不能比过渡区最小主要特征大很多, 这两方面在时域是很难同时满足的。我们可以在频域解释这个现象, 过渡区 T 在频域表现为一个窄的带通信号, 如果要对它平滑过渡, 则 T 不能比过渡区图像的最低频的波长小得太多, 而如果 T 比过渡区图像的最高频波长的 2 倍大很多, 则会出现鬼影现象^[5], 这也正是多分辨率分析图像融合被提出的原因。将时域图像变换到频域后, 在不同尺度和不同频带上选取不同的过渡区长度 T , 就既能平滑过渡, 又能避免“鬼影”的出现, 达到很好的融合效果。

文献^[5]建议, T 在 l 层时的长度取为 2^l 。考虑到使用小波变换进行多分辨率分析的融合效果, 我们在第 k 层融合时, 以第 k 层的缝合线为中心, 取 $T = 2^k + 1$ 作为过渡区长度, 故在第 3 层小波变换时 T 的长度为 9, 第 2 层长度为 5, 第 1 层长度为 3。

3.3 小波域融合

小波域融合是整个图像融合的重点, 其主要分两步:

- 1) 融合规则确定。
- 2) 根据融合规则进行逐个像素融合更新。

3.3.1 融合规则确定

融合规则对整个融合图像的质量起到关键作用, 也即式(4)中 H_L 和 H_R 的具体形式, 出于实时性考虑, 我们在选择融合规则的时候, 考虑了两个大方向:

1) 过渡区内, 选择两幅图中小波系数绝对值大的系数作为最后的融合系数, 公式如下:

$$f_{ij}^F = \max\{f_{ij}^A, f_{ij}^B\} \quad (6)$$

式中, f_{ij}^F 为融合后的小波系数, f_{ij}^A 为图像 A 的小波系数, f_{ij}^B 为图像 B 的小波系数。

2) 因为要平滑过渡缝合区域, 在众多函数中, 反比例函数在大于 1 的区间内函数值较小且变化比较平滑, 可以使过渡区系数尽量接近原值且平滑过渡, 所以我们采用一个反比例加权函数作为两幅图像小波系数的因子来进行融合, 会使效果最好, 具体公式如下:

$$f_{ij}^{F^l} = \begin{cases} f_{ij}^{A^l} H_L^l + f_{ij}^{B^l} H_R^l, & i \in T_L^l \\ f_{ij}^{B^l} H_L^l + f_{ij}^{A^l} H_R^l, & i \in T_R^l \end{cases} \quad (7)$$

式中, $f_{ij}^{F^l}$ 为 l 层的融合后小波系数, $f_{ij}^{A^l}$, $f_{ij}^{B^l}$ 分别为图像 A、图像 B 的 l 层小波变换系数, $H_L^l = \frac{1}{2} + \frac{1}{2(2^l \times |x_{ij} - \hat{x}|)}$, $H_R^l =$

$$\frac{1}{2} - \frac{1}{2(2^l \times |x_{ij} - \hat{x}|)}$$

分别为 l 层小波变换的左侧、右侧融合函数, $|x_{ij} - \hat{x}|$ 为当前点 x 方向距缝合线的距离 (\hat{x} 为当前层的缝合线上点的 x 坐标), T_L^l , T_R^l 分别为 l 层过渡区位于缝合线左侧和右侧的 x 坐标集合。融合函数中的 $\frac{1}{2}$ 是为了保证融合图像在缝合线上对应像素点的值为两幅原图的像素点的平均值^[5]。

实验发现, 系数线性加权的效果要远优于系数绝对值取大的效果, 因此, 最后选择了式(7)作为我们的融合规则。

3.3.2 像素融合

在融合的时候, 要逐个像素来计算, 过程如下:

- 1) 判断当前像素所在的层数。
- 2) 判断当前像素是否在过渡区内, 若不在过渡区内, 则直接赋值为两幅图中不为零的小波系数, 若在过渡区内, 则使用式(7)作为融合规则更新像素。

3.4 融合后的逆变换

融合完毕后, 要对融合后的小波系数做逆变换, 得到最后的融合图像, 结果见第 4 节内容。

3.5 拼接痕迹消除算法实现

总结上述讨论, 我们可以给出拼接痕迹消除算法如下:

- 1) 对配准后的两幅图做 3 层 Haar 小波变换。
- 2) 在配准后的图像中搜索缝合线: 首先根据边界条件找到两个公共边界点, 然后利用这两点计算出缝合线。
- 3) 确定过渡区: 首先把缝合线多尺度化, 每层过渡区在某 y 方向固定的情况下, 以缝合线为中心取 x 方向的长度为 $2^l + 1$ 。

4) 像素融合: 判断当前点是否在过渡区, 若在过渡区则根据式(7)来进行系数融合。

5) 将融合后的小波系数做逆变换, 得到融合后的图像。

可见本文提出的方法没有复杂的运算, 这保证了本文方法的实时性, 融合后的效果也很令人满意, 具体结果见第 4 节。

4 仿真和实验

为了验证本文提出的图像拼接痕迹消除方法的效果和实时性, 我们组织了大量实验, 由于篇幅的限制, 我们仅展示其中一个实例。我们用经典的时域图像加权平均法(简称 AVG-Fusion)与本文方法进行对比, 并组织以下实验验证。

实验用的计算机配置为: 主频 1.8GHz, 内存 1GByte; 仿真软件为 Microsoft visual C++ 6.0; 利用的函数库为 OpenCV 1.0。

4.1 评价方法

在众多方法中, 经典加权平均法是运算速度最快的一种方法, 为了验证本文方法的实时性, 用这两种方法分别对同一场景不同分辨率下的两幅配准后的图像进行拼接, 用微软级 Windows API 函数记录各自的运行时间, 并给出对比曲线。

为对比本文提出的方法和经典加权平均法的效果, 我们采用主观评价结合客观评价的形式。主观采用评分制, 客观采用融合后图像的标准差和平均梯度参数来进行对比衡量。

由于篇幅限制, 我们仅以多组实验中的一组来进行说明。

4.2 实验结果

4.2.1 运行时间对比

我们采用本文的方法和经典加权平均方法分别对同一场景原图分辨率为 $192 \times 256, 200 \times 272, 224 \times 304, 256 \times 336, 288 \times 384, 336 \times 448, 384 \times 512$ 的图像进行融合, 其中 288×384 的两幅原图如图 6 所示。

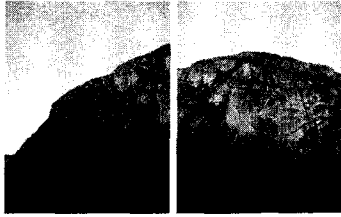


图 6 288×384 的两幅原图

两种方法的时间对比曲线如图 7 所示。

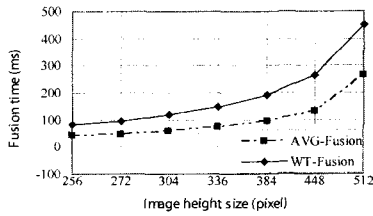


图 7 各分辨率下 AVG-Fusion 法和 WT-Fusion 法时间曲线

4.2.2 融合效果对比

分别利用两种方法对两幅 288×384 的原图进行融合, 融合结果如图 8, 图 9 所示。

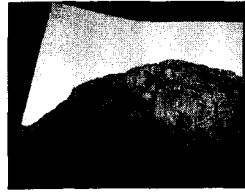


图 8 AVG-Fusion 法在缝合线左右 8 像素进行融合后的图像
图 9 WT-Fusion 法融合后的图像

对于主观评价, 我们采用表 1 所列的尺度评分表。

表 1 主观评价尺度评分表

分数	质量尺度	妨碍尺度
9	非常好	丝毫看不出图像质量变坏
8	很好	很难看出图像质量变坏
7	较好	基本看不出图像质量变坏
6	一般好	不妨碍观看和信息获取
5	稍好	基本不妨碍观看和信息获取
4	稍差	稍许妨碍观看和信息获取
3	一般差	妨碍观看且妨碍信息获取
2	较差	严重妨碍观看较大程度妨碍信息获取
1	很差	非常严重地妨碍观看及信息获取

根据表 1, 我们经过向相似领域实验室的同学及老师征求意见, 最后统计本文提出的方法打 8 分, 给经典加权平均法打 6 分。

客观评价, 针对两种融合方法, 我们考查融合后图像的空间分辨率和清晰度, 分别采用融合后图像的标准差和平均梯度两个参数来衡量融合后的效果, 两个参数计算公式如下:

$$SD = \sqrt{\frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (I(i, j) - \bar{I})^2} \quad (8)$$

$$AG = \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \sqrt{(\Delta I_x^2 + \Delta I_y^2) / 2} \quad (9)$$

式(8)中 \bar{I} 为像素的均值, 式(9)中 $\Delta I_x, \Delta I_y$ 分别为 x 与 y 方向的差分。

由于采用的是彩色图像, 因此我们分 BGR 3 个通道计算融合后图像的标准差和平均梯度, 结果如表 2 所列。

表 2 两种参数对比表

图像	标准差 SD(BGR)			平均梯度(BGR)		
原图 1	99.8	80.9	67.6	9.25	8.55	6.32
原图 2	88.8	72.7	63.3	9.11	8.05	6.44
AVG-Fusion (过渡区)	24.7	21.1	26.9	4.15	3.76	3.06
WT-Fusion (过渡区)	34.4	27.7	34.7	6.24	5.99	5.12

4.3 实验结果分析

由图 7 可见, WT-Fusion 方法大约是 AVG-Fusion 方法在同一分辨率图像上所用时间的 2 倍, WT-Fusion 方法耗时主要是两点: 一是小波变换, 二是要针对不同层、不同带通子图像要做不同尺度、不同融合规则处理。总体来看, AVG-Fusion 方法是最为快捷方法, 而我们提出的 WT-Fusion 的所用时间虽然是其 2 倍, 但是时间相差不大, 如一幅 288×384 的图像使用 AVG-Fusion 方法需要 93.52ms, 使用 WT-Fusion 方法需要 192.42ms, 所以本文提出的 WT-Fusion 方法的实时性基本满足实时性要求。

由图 8、图 9 可见, 我们提出的 WT-Fusion 方法的效果要比 AVG-Fusion 方法在缝合线左右 8 像素的效果好很多, 我们的方法看不出“拼接缝”, 而 AVG-Fusion 方法可明显看出“拼接缝”, 若要 AVG-Fusion 方法实现平滑过渡, 则要增大其过渡区, 但是随着过渡区增大, 过渡区的图像将出现模糊现象(“鬼影”), 而我们的方法则完全避免了此问题。

由表 2 可见, 在图像过渡区内, 我们提出的 WT-Fusion 方法在标准差和平均梯度两个参数的结果上都比 AVG-Fusion 要好。标准差体现了图像空间分辨率大小, 越大表示图像空间分辨率越好; 平均梯度体现了图像的清晰度, 越大表示图像越清晰^[4]。综合表 2, 可以看出我们的方法效果要远优于 AVG-Fusion。

结束语 本文提出的图像拼接痕迹消除算法主要有以下优点:

- 1) 用简单、有效的方法搜索缝合线, 减少了运算时间, 从而保证了整个算法具有较好的实时性。
- 2) 利用成熟的小波多分辨分析技术, 在不同层上利用不同的融合规则, 可有效地清除缝合线痕迹。

后续研究将针对多幅图像, 结合小波变换、高效缝合线搜索以及捆绑调整技术, 实现高质量全景自动拼接。

参考文献

- [1] Liu Gang, Lu Xue-qin, Huang Guo-hong. Multisensor image fusion based on optimal filter bank[C]//Proceedings of the 2008 International Conference on Wavelet Analysis and Pattern Recognition. 2008, 1: 177-182
- [2] Wu Jin, Liu Yang, Liu Jian, et al. Wavelet Based Remote Sensing Image Fusion with Color Compensation Rule and HIS Transform[C]//IEEE International Conference on Mechatronics and Automation. 2006: 2079-2083

(下转第 292 页)

以对 viewee 线程的统计数据进行了归并,然后及时释放非活动的 pview stat 结构。

6.2 进程上下文切换时 PMU 操作的优化

在大多数集成了 PMU 的处理器(例如 IA、MIPS 和 POEWEER 系列)中,PMU 作为特殊的寄存器不属于执行上下文寄存器组的范围;无论从概念上还是从其访问特点来看,把 PMU 寄存器划分为硬件执行上下文都是不合适的,因此不能通过修改内核里执行上下文的数据结构来扩展 PMU 上下文。我们把 PMU 上下文视为进程的一个特殊属性。

Pview 对上下文切换时 PMU 的 save/restore 操作做了两点重要优化。

第一,在进程的硬件上下文描述数据结构 thread_struct 中添加一个长整型变量 pmu_bitmap,以位图形式指示当前 PMU 寄存器对的使用情况;pview_stat 结构中保存 PMU 控制寄存器值和计数寄存器值。在上下文切换时这两个数据结构就可以保证 PMU 的保存与恢复;由于这两个结构都是线程私有的,因此监测在多处处理器平台上运行多线程 viewee 程序时,可以不需要任何锁相关的操作。Pview context 结构只在 viewee 中线程对监测环境初始化、统计数据处理等操作时才去读。

第二,对 PMU 寄存器的恢复采取 lazy update 的策略。PMU 寄存器的操作一般都非常耗时;以 Intel 的 CPU 为例,对 PMU MSR(Model Specific Register)的访问指令 throughput 高达 100 个时钟周期以上^[12],这对于上下文切换这种频繁的调用是难以接受的性能损失。但是,基于两点事实,我们有理由对其 lazy update。首先,程序性能监测过程中,绝大多数情况下,同一个处理器核在一段时间内只会使用一个 pview context。多处处理器系统中,同一进程的线程调度倾向于分配到一组核上,由于这些线程共享 pview context,它们的 PMU 配置数据结构也是相同的,因此不需要经常更新 PMU 的控制寄存器。其次,由于进程调度的 CPU affinity 考虑,一个进程总是会优先被调度到上次执行所在的处理器核上执行,这样如果在两次执行期间,该处理器核上执行的其他程序没有使用 PMU,那么上次执行的 PMU 现场就没有被破坏,进而 PMU 的控制和计数寄存器都不需要进行更新,只需开始继续计数。

我们为每个处理器核定义了两个 per-cpu 变量,即 last_pv_ctx 和 last_pv_stat,分别指示该处理器(核)上次使用的 pview context 和 pview stat 的 ID;pview context 的 ID 为一个整数,每次创建 pview context 或修改其中的 PMU 配置相关域,该 pview context 的 ID 值都要自增;pview stat 的 ID 使用拥有它的进程 pid;每次上下文切换时,通过对比所在处理器

与被调度进程的这两个变量是否一致来决定是否更新 PMU 寄存器值。

结束语 本文介绍了一种 Linux 下基于硬件性能计数单元的程序性能分析工具——Pview。它支持全系统范围和针对特定进程的性能事件直接计数和抽样的数据收集方法,以系统调用的方式为用户提供了一种利用 PMU 进行性能监测的统一接口,同时在内核层次对多线程程序性能事件数据统计给予了特别的支持;抽样过程是由独立的数据收集引擎以内核模块的形式实现的。

对 Pview 系统调用进行封装后的 Pview 库,可以为用户提供使用 PMU 来获取应用程序性能事件数据的 API;在此基础上,可以进一步构造用户层的工具链。

参考文献

- [1] Luk C K, Cohn R, et al. Pin: Building Customized Program Analysis Tools with Dynamic Instrumentation[C]// Proc. of the ACM SIGPLAN Conference on Programming Language Design and Implementation(PLDI). Chicago, IL, US, June 2005:190-200
- [2] Cantrill B M, Shapiro M W, Leventhal A H. Dynamic Instrumentation of Production System[C]//Proc. of USENIX ATEC. Boston, MA, US, June 2004:2-2
- [3] Srivastava A, Eustace A. Atom: A system for building customized program analysis tools[C]//Proc. of PLDI. Orlando, Florida, US, 1994:196-205
- [4] Sprunt B. The basics of performance monitoring hardware[J]. IEEE Micro, July 2002:64-71
- [5] Intel Corp. VTune [EB/OL]. <http://www.intel.com/software/products/vtune/>
- [6] Levon J. Oprofile[EB/OL]. <http://oprofile.sourceforge.net/about/>
- [7] Eranian S. Perfmon2[EB/OL]. <http://perfmon2.sf.net/>
- [8] Molnar I. Perf[EB/DK]. Linux Kernel Document, 2. 6. 33
- [9] Ahn D H, Vetter J S. Scalable analysis techniques for microprocessor performance counter metrics[C]//Proc. of Conference on Supercomputing. Baltimore, Maryland, US, Nov. 2002
- [10] Sweeney P F, Hauswirth M, Cahoon B, et al. Using hardware performance monitors to understand the behavior of Java applications[C]//Proc. of 3rd Virtual Machine Research and Technology Symposium. San Jose, CA, US, May 2004:5-5
- [11] Duesterwald E, Cascaval C, Dwarkadas S. Characterizing and predicting program behavior and its variability[C]// Proc. of 12th Intl. Conference on Parallel Architecture and Compilation Techniques(PACT). Dec. 2003
- [12] Intel Corp. Intel 64 and IA-32 Architectures Optimization Reference Manual [M]. Nov. 2009: C. 3
- [3] 倪国强. 基于视觉神经动力学的图像融合与处理技术若干新进展[J]. 激光与红外, 2005, 35(11):817-821
- [4] Luo Yan, Liu Rong, Zhu Yu-feng. Fusion of Remote Sensing Image Base on the PCA + ATROUS Wavelet Transform[J]. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2008, 37 (B7): 1155-1158
- [5] Burt P J, Adelson E H. A multiresolution spline with application to image mosaics[J]. ACM Trans. Graphics, 1983, 2(4): 217-236
- [6] Garzelli A. Wavelet-based Fusion of Optical and SAR Image Data Over Urban Area[C]// Photogrammetric Computer Vision. ISPRS Commission III, symposium2002. Graz, Austria. 2002:59-62
- [7] 李晖晖, 郭雷, 李国新. 基于脊波变换的 SAR 与可见光图像融合研究[J]. 西北工业大学学报, 2006, 24(4): 418-422
- [8] Choi M, Kim R Y, Kim M G. The curvelet transform for image fusion[C]// International Society for Photogrammetry and Remote Sensing, ISPRS. 2004, 35:59-64
- [9] Zhang Qiang, Guo Bao-long. Multifocus image fusion using the nonsubsampled contourlet transform [J]. Signal Processing, 2009, 89:1334-1346