

基于 .NET 虚拟机的开放式图像处理平台设计

白雪丽

(洛阳光电设备研究所, 河南 洛阳 471000)

摘要: 设计了一个基于 .NET 虚拟机的开放性图像处理平台, 采用虚拟机技术解决了跨平台问题与内存模型兼容性问题, 突破了传统平台封闭、难于开发的问题; 平台完全对用户开放, 不设立绝对的格式规范, 使各个现存的图像处理功能模块得以大幅度的综合与重复利用; 最后针对阈值图像分割算法, 利用 Fisher 准则作为评判标准, 对算法进行了优化实验, 验证了平台的有效性与易用性。

关键词: 图像处理平台; 虚拟机; 算法优化

An Open Image Processing Platform Based on .NET Virtual Machine

Bai Xueli

(Luoyang Institute of Electro-optical Devices, Luoyang 471000, China)

Abstract: Design a open platform for image processing based on .net virtual machine. The virtual machine technology can solve the cross-platform problem and the compatibility problem of memory model. Breaking the traditional platform which is closed to user and difficult development issues. Platform is completely open to the user, set few absolute format specifications. So that all the existing image processing function modules can be most integrated and reuse. Finally through the optimization experiments of threshold image segmentation algorithm verifies the effectiveness of the platform and ease of use.

Keywords: image processing platform; virtual machine; optimized algorithm

0 引言

随着应用需求的不断扩大和计算机技术的飞速发展, 数字图像处理面临着复杂化和高速化的挑战。目前常用的图像处理软件, 例如 Photoshop、MATLAB Image Toolbox、PaintShop Pro、PhotoLine 等^[1], 这些软件具有比较出色的性能与处理能力, 但它们的着眼点多在于功能的扩展上, 并且拥有自己定义的标准内部格式, 相互兼容性差。长期以来, 并没有一种通用的平台可以兼容不同平台的格式及算法。本文设计了一种平台, 仅仅开放一组非常简单的接口, 所有的处理逻辑都围绕接口而展开, 使得整个平台都是抽象的, 避免了对内在规定的依赖, 从而具有了通用性。

1 开放图像处理平台的总体设计

.Net 平台以其跨语言, 跨平台, 安全, 以及支持开放互联网标准和协议等优点广泛应用于开发环境中。它具有一致的编程模型, 简化的编程方式, 轻便的部署, 无缝的语言集成, 丰富的调试支持, 强大的互操作能力, 统一的错误报告, 为开发出性能优异的测试平台提供了有力保障^[2]。

相比于其他与硬件紧密相关的操作系统而言, 虚拟机独立于底层硬件运行。因为虚拟机在各种平台上都具有一致的内存模型, 线程库的代码对于各个不同平台的虚拟机来说并没有任何区别, 使得实现多线程并行程序的方法变得可行。目前虚拟机发展的非常迅速, 例如 Java 拥有 ARM、SPARC 及其他处理器平台上的稳定实现^[3], .NET 也拥有在移动平台上的

Compact Framework 实现^[4]。可以想象, 由于虚拟机具有突出的通用性及部署优势, 其将在更多的硬件平台上得以应用, 具有良好的发展前景。于是, 本文选择 .NET 虚拟机作为线程库的运行环境。

整个平台分为底层简单类型接口, 运行逻辑与开发插件接口, 这种设计与一般的软件设计有所区别。在这个结构中, 用户能够扩展的不仅仅是提供各种处理功能的插件, 而且能够控制系统底层使用的图像类型或者色彩类型的具体实现。这样就不需要复杂的类型转换, 仅仅将现有的程序稍作包装就可以在新的平台上使用。开放的图像处理平台结构图如图 1 所示。

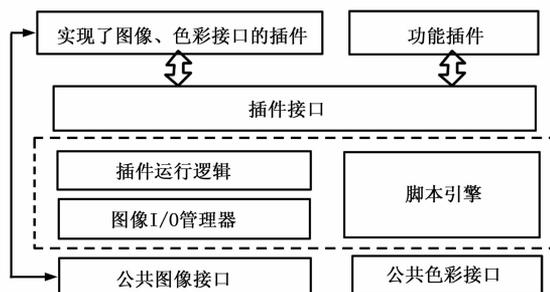


图 1 开放图像处理平台的总体结构图

整个平台中, 运行逻辑是联系用户实现的枢纽, 实现了整个平台的功能, 而其余两层仅仅提供了可供用户扩展的接口。

2 开放图像处理平台的关键模块设计

2.1 图像 IO 管理器设计

图像的 IO 管理器是图像处理平台的基础, 直接影响公共图像接口的设计。图像与其他类型的数据结构不同, 通常来说, 图像数据比一般的数据类型要大的多。为了最大限度的提

收稿日期: 2014-03-05; 修回日期: 2014-04-05。

作者简介: 白雪丽(1978-), 女, 工程师, 主要从事音视频、图像数据处理方向的研究。

升图像的 IO 操作性能，在系统中采用了中间磁盘缓存的方式管理系统中所有图像变量，以节省内存的使用，同时，预留了图像的内存以提升读取一幅图像的性能。图像 IO 管理器的设计结构如图 2 所示。

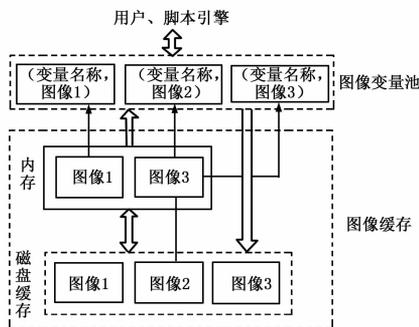


图 2 图像 IO 管理器结构简图

在传统的缓存设计中，需要处理原始数据更改之后缓存不一致的问题，这是一个比较复杂的操作。本文中认为每一个图像都是只读的，仅仅将更改后的结果保存为另一个变量，避免了对于缓存不一致性的处理。这样做同时允许用户操作失败后重新操作，使用户更加容易处理各种异常情况。

由于用户实际使用的图像的制式可能是各种各样的，每种制式都有其独特之处，因此我们期望图像缓存中的制式与用户的原始图像制式一致以避免信息的丢失。目前使用比较广泛的是 32 位 RGB 图像^[5]，所以本文将 32 位 RGB 位图作为可选的中转制式。

2.2 插件系统逻辑设计

插件系统为整个平台提供了扩展支持，可以通过单纯的将插件挂靠到系统中以扩展系统的性能，加之公共图像接口与颜色接口的设计，整个系统的插件就可以几乎无缝的衔接在一起。我们通过 .NET 的反射方式实现插件系统，只规定接口类型而不限名字空间和实现方式，使用户简单的实现单程序集中多个插件类型的聚合。

在程序启动的时候装入插件会造成资源的浪费，因此，采用配置文件的方式进行插件的注册，如图 3 所示。这样做有两点好处：第一，明确的定位了程序集中的每一个插件；第二，可以自由的更改插件对应的命令，方便用户对命令系统的定义，提高用户的使用体验。

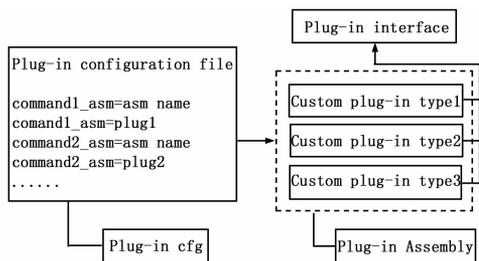


图 3 插件注册系统结构简图

由于图像处理的任务分类较多，不同类型任务之间的差异也比较大，采用一种单一的插件接口是不合适的。本文将整个插件接口按照任务的类型分为 5 类：图像处理插件、颜色空间

插件、图像统计插件、图像编码和解码插件，这样用户针对某一特定功能开发插件时，不必考虑由于系统通用性而引进额外的编码负担。

(1) 图像处理插件：包含图像处理的逻辑，图像处理插件的输入为一个或者多个图像变量与相关控制参数，输出为处理结束后的图像变量以及生成的文档。插件提供处理过程中的信息输出机制，方便用户观察插件的运作或者帮助进行插件开发的测试。

(2) 颜色空间插件：包含用户实现的颜色空间，以及与系统默认的颜色空间的转换逻辑。系统并没有规定某一种颜色空间作为系统的标准颜色空间，而仅仅提供两种默认颜色空间以方便颜色转换算法的实现。第一种默认的颜色空间为 RGB 颜色空间，第二种为 CIE 颜色空间，用户可以提供这两种空间的颜色转换接口以进行没有直接相关的颜色空间的转换操作。

(3) 图像统计插件：图像统计插件并不生成新的图像变量，而是从现有的图像中得到数据形式的或者可视化的信息。

(4) 图像编码插件：将图形变量转存为某种类型的文件。

(5) 图像解码插件：将某种类型的文件作为图形变量输入到系统中。

前面已经指出，为了避免图像信息的损失，采用用户自定义制式对每一个图像变量进行储存。但是如何让插件在不同的图像制式之间保持兼容性呢？这时候就需要用到系统默认的图像制式作为中间媒介了。插件的兼容性逻辑设计如图 4 所示。

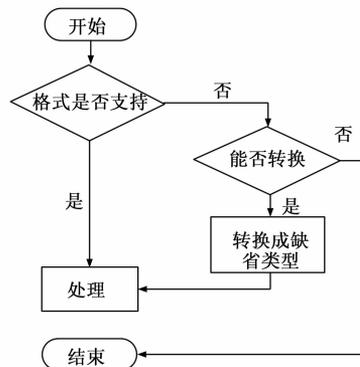


图 4 插件的兼容性逻辑设计简图

这样对于用户自定义的图像制式，则兼容该制式的插件并直接处理图像数据，而不会有任何额外的操作。否则如果插件可以接受系统默认制式的图像，将该数据转换为系统默认制式，再进行处理。

2.3 脚本引擎系统与插件接口的交互设计

本次平台的设计引入了脚本引擎系统，使用户不用进行额外插件编写工作就可以实现自己的图像处理流程。为保证脚本引擎功能调用的一致性，首先引入了命令系统来驱动每个插件，而后在脚本中通过命令的调用来实现图像的处理逻辑。系统由前端的控制台接受用户输入命令，通过命令类别与名称在配置文件中找到相关的程序集与其中具体的插件类型，并创建其实例。将创建的插件实例与相应的插件接口绑定之后，送入插件的执行逻辑执行插件的功能，最后根据插件的类型选择将

结果通过 IO 管理器输出到文件或者输出为图像变量。图 5 为命令系统驱动插件工作的流程。

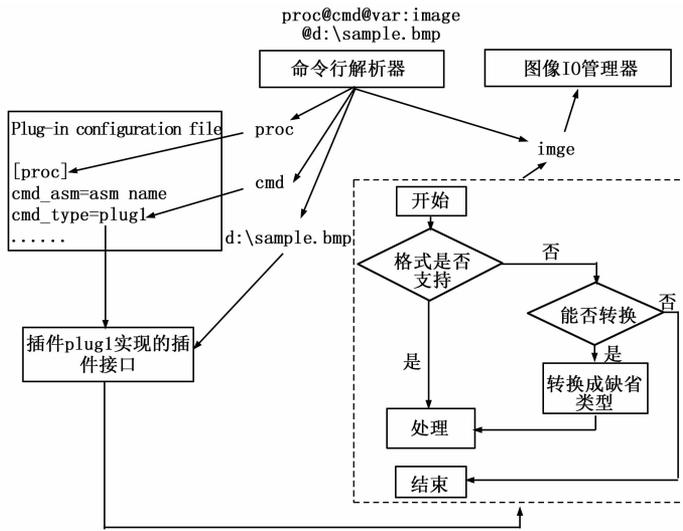


图 5 命令系统驱动插件工作流程图

命令系统采用了非常简单的接口形式并将接口暴露给用户，以使用户在脚本引擎中以一种一致的方式进行命令的调用。命令系统实际上涵盖了 .NET 基本的程序集使得用户可以再脚本系统中使用 .NET CLI 的几乎所有功能。同时，系统提供了控制台输出使得用户方便的从命令行输出脚本的执行状态或调试自己的脚本。

3 基于平台的图像阈值分割算法的并行优化验证

作为计算机视觉和图像处理中的难点和热点之一，图像分割的研究受到了研究工作者的的高度重视，对图像分割进行了深入、广泛的研究^[6]。但现提出的分割算法大多是针对具体问题的，并没有一种合适所有图像的通用算法。图像分割算法中最常用的就是阈值分割法^[7]，因此本文将图像分割法为例，介绍基于 .NET 虚拟机的线程库与开放图像处理平台的图像阈值分割算法的并行优化测试。通过实验数据展现图像处理平台的易用性，同时说明虚拟机线程库对算法性能的提升。

以 Fisher 准则作为评判标准，算法描述如下^[8]：对彩色图像的每个通道来说，其最佳阈值定义为：

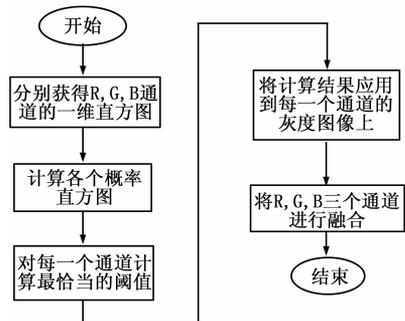
$$t^* = \operatorname{argmax}_{t \in G} \{J(t)\}$$

$J(t)$ 为所有可能的阈值的评价函数结果：

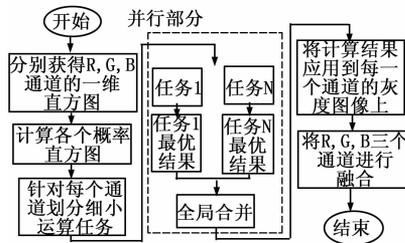
$$J(t_0, t_1, \dots, t_{N-1}) = \frac{|w_0 u_0 - w_1 u_1|^2 + |w_1 u_1 - w_2 u_2|^2 + \dots + |w_{N-1} u_{N-1} - w_N u_N|^2}{w_0 \sigma_0^2 + w_1 \sigma_1^2 + \dots + w_N \sigma_N^2} \quad (1)$$

式中， W_j 为第 i 类的先验概率， u_i 为第 i 类的数学期望， σ_i 为第 i 类的类内方差。

为了计算某一个通道上的最佳阈值 t^* ，我们需要分别计算 W_j 、 u_i 、 σ_i 。由于 Fisher 算法的 HitCount 非常巨大，可以考虑将其看作一个整体再高层次进行优化，而其内部方法则不必过度并行化以避免并行的过度暴露。我们引入线程内存储，采用并行库提供的分而治之的算法接口对整体过程进行优化^[9]。整体算法改进前和改进后的流程图如图 6 所示。



(a) 改进前



(b) 改进后

图 6 图像阈值分割算法流程图

由图 6 可知，整个算法流程的变化比较显著，而且这种变化仅仅通过非常少的工作就可以完成。只需告知程序库如何初始化局部结果以及如何将程序计算得到的局部结果进行合并，具体的细节全部由程序库自动完成，真正执行计算的部分集合没有任何改动就可以直接使用了。

将分割程序以插件的形式植入图像处理平台，并对其效果进行测试。测试结果如图 7 所示。

图像阈值分割法的并行优化结果

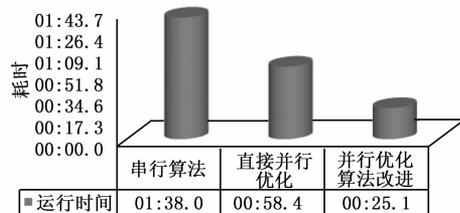


图 7 图像阈值分割算法的基准测试结果

4 结果分析

从以上验证实验可以看出，平台可以有效对图像阈值分割算法进行整合，在对算法的内部代码没有作任何改动的情况下，算法的执行时间也从原本的 1 分 38 秒降低到 58 秒，是原本执行时间的 59%。而如果针对小范围的迭代进行有针对性的算法改进，算法的运行时间更是缩短到了 25 秒，是原本执行时间的 25.5%。可见平台的易用性与有效性，同时在平台上采用并行算法可以大大提升算法的执行效率。

5 结束语

针对传统平台封闭、开发困难的特点，设计了插件系统完全透明的图像处理平台，完全对用户敞开，不设绝对的格式

规范，使各个现存的图像处理功能模块得以最大限度的综合与重复利用。整个平台从逻辑需求出发，严格基于接口进行设计，符合面向对象设计的开闭原则和依赖反转原则，即使更改细节也不会影响到既有插件的运行。整个平台改变了以往系统提供底层格式的惯例，使得现存的图像处理库得到充分再利用。

参考文献：

[1] 孙浩, 陈安, 胡跃明. 基于 DSP 和 FPGA 的通用图像处理平台设计 [J]. 电子设计工程, 2009, 6 (17): 41-43.
 [2] 黎松奇. 基于 .Net 平台的通用自动测试系统设计 [J]. 自动化与仪器仪表, 2011, (5): 45-47.
 [3] 李允, 罗蕾, 雷昊峰, 等. 嵌入式 Java 虚拟机的性能优化技



(上接第 2150 页)

集 3 种类别的运动定位结果 (跑步, 俯冲和骑马)。本文方法对俯冲类别的性能优于文献 [15-16] 中的方法, 但是对其余两种低于文献 [15-16] 中的方法。然而, 与本文方法相比较的其他各种算法均使用成本很高的人体边界框注释, 且它们的学习过程的复杂度远高于本文方法。

对 HighFive 数据集, 对注释为包含交集的帧计算 IOU。我们的 IOU 为 30.8%, 虽然仍然非常优异, 但是低于 UCF Sports 数据集。为验证这一观点, 我们也计算了检索率 (检索率定义为交集区域与被注释的运动区域之比)。表 4 中检索率较高, 证明被注释的运动区域主要由本文方法检测出。文献 [15] 和 [16] 分别给出了接吻类别的运行定位结果, 分别为 18.5% 和 39.5%。这些结果仍然没有直接可比性, 因为文献 [15] 和 [16] 均需要人体边界框, 而本文方法需要的监管量很小 (只需标签)。

表 4 High Five 数据集衡量为 IOU (%) 和检索率 (%) 均值的运动定位性能

类别	握手	致意	拥抱	接吻	平均
IOU	26.9	32.9	34.2	29.3	30.8
检索率	79.4	88.8	82.6	80.8	82.3

4 结论

本文提出了一种改进的人类动作识别与定位方法, 可用于有效检测真实视频中事件发生的可靠性和地点。仿真实验结果表明, 本文方法是有效的。我们的下一步工作是基于压缩感知理论, 进行视频的异常检测与定位。

参考文献：

[1] 徐光祐, 曹媛媛. 动作识别与行为理解综述 [J]. 中国图象图形学报, 2009, 14 (2): 189-195.
 [2] 吴伟和, 郝爱民, 赵永涛, 等. 一种人体运动骨骼提取和动画自动生成方法 [J]. 计算机研究与发展, 2012, 49 (7): 1408-1419.
 [3] 谭论正, 夏利民, 黄金霞, 等. 基于 pLSA 模型的人体动作识别 [J]. 国防科技大学学报, 2013, 35 (5): 102-108.
 [4] Laptev I, Marszalek M, Schmid C, et al. Learning realistic human actions from movies [A]. Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on. IEEE [C], 2008: 1

术 [J]. 计算机工程, 2004, (18): 46-49.
 [4] 严永斌, 吴健平. 基于 .NET Compact Framework 的移动 GIS 软件开发 [J]. 测绘与空间地理信息, 2008, 31 (4): 37-41.
 [5] 王鸣秋. 浅谈数字图形图像存储格式 [J]. 湖南广播电视大学学报, 2004, (1): 31-36.
 [6] 梁艳. 基于模糊理论的图像分割算法研究 [D]. 西安: 西安电子科技大学, 2012.
 [7] 张伟. 指纹图像分割算法的研究 [J]. 计算机测量与控制, 2011 (7): 1746-1748.
 [8] 陈果. 图像阈值分割的 Fisher 准则函数法 [J]. 仪器仪表学报, 2003, 24 (6): 564-567.
 [9] 夏红霞, 钟璐. 分而治之算法的并行性 [J]. 计算机应用, 1989, 4: 29-31.
 -8.
 [5] Wang H, Klaser A, Schmid C, et al. Action recognition by dense trajectories [A]. Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. IEEE [C], 2011: 3169-3176.
 [6] Poppe R. A survey on vision-based human action recognition [J]. Image and vision computing, 2010, 28 (6): 976-990.
 [7] Wang Y, Tran D, Liao Z, et al. Discriminative hierarchical part-based models for human parsing and action recognition [J]. The Journal of Machine Learning Research, 2012, 13 (1): 3075-3102.
 [8] Rodriguez M D, Ahmed J, Shah M. Action mach a spatio-temporal maximum average correlation height filter for action recognition [A]. In CVPR [C], 2008: 1123-1132.
 [9] Patron-Perez A, Marszalek M, Zisserman A, et al. High five: Recognising human interactions in tv shows [J]. 2010.
 [10] Leordeanu M, Sukthankar R, Sminchisescu C. Efficient closed-form solution to generalized boundary detection [M]. Computer Vision - ECCV 2012. Springer Berlin Heidelberg, 2012: 516-529.
 [11] Arbelaez P, Maire M, Fowlkes C, et al. From contours to regions: An empirical evaluation [A]. Computer Vision and Pattern Recognition, CVPR 2009. IEEE Conference on. IEEE [C], 2009: 2294-2301.
 [12] Raptis M, Kokkinos I, Soatto S. Discovering discriminative action parts from mid-level video representations [A]. Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE [C], 2012: 1242-1249.
 [13] Lan T, Wang Y, Mori G. Discriminative figure-centric models for joint action localization and recognition [A]. Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE [C], 2011: 2003-2010.
 [14] Gaidon A, Harchaoui Z, Schmid C. Recognizing activities with cluster-trees of tracklets [A]. BMVC [C], 2012.
 [15] Tran D, Yuan J. Optimal spatio-temporal path discovery for video event detection [A]. Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. IEEE [C], 2011: 3321-3328.
 [16] Tran D, Yuan J. Max-margin structured output regression for spatio-temporal action localization [J]. 2012, 21 (13): 2013-2020.