

# 基于 MMX™ 技术的图象边沿和边沿交点的快速提取

吴雨 林学闯

(清华大学 计算机科学与技术系, 北京 100084)

**摘要** 介绍了 Intel 最新推出的 MMX™ 技术及其在图象边沿和边沿交点提取快速算法实现中的应用。边沿提取的速度可达 30 帧/秒。

**关键词** MMX™ 技术, 边沿提取, 边沿交点提取

## 1 引言

Intel 为多媒体应用推出的 MMX™ 技术的新一代奔腾芯片是 96/97 一个热点话题。MMX™ 的出现, 必将为图象、语音、多媒体、三维模式生成、虚拟现实等技术在微机上的应用创造一个新的天地。我们首先将它应用在三维视觉的前期图象处理当中, 并取得了良好的加速效果。

## 2 MMX™ 技术简介

所谓 MMX™ 技术, 实际是对奔腾指令集的一个多媒体扩展。其核心思想是适应多媒体时代低精度, 大数据量的数据流特点, 充分地利用奔腾 64 位宽度的数据通路, 在一个指令周期内并行处理多个数据。Intel 的 MMX™ 计算机包括 P5+MMX™ 的 P55C 和 P6+MMX™ 的 Klamath。MMX™ 技术的主要要点是:

- (1) 4 种新的 64 位数据类型: 紧缩字节、紧缩 16 位字、紧缩 32 位双字、4 字。可以在一定程度上实现有限精度定点数据的并行操作。
- (2) 8 个 64 位 MMX™ 寄存器 MM0~MM7;
- (3) 57 条 MMX™ 指令组成的 MMX™ 指令集:

① MMX™ 指令集由以下五组指令组成:

- 算术指令: 加, 减, 乘, 比较
- 逻辑指令: 逻辑操作, 移位
- 转换指令: 紧缩, 解紧缩
- 数据移动指令: 对内存或寄存器的 32/64 位访问
- EMMS: 清除 MMX 状态

MMX™ 指令集使用保留的操作码和奔腾的各种寻址方式, 指令的风格也与 86 系列相同。

② MMX™ 算术指令有饱和与环绕两种操作方式。例如:

$0xaf + 0xd7 = 0x186$  超出 Byte 表示的范围

环绕算法:  $0xaf + 0xd7 = 0x86$

饱和算法:  $0xaf + 0xd7 = 0xff$

其中饱和算法又分为按有符号数饱和及按无符号数饱和两种方式。

③ MMX™ 指令和寄存器只对数据进行操作, 不参与寻址和程序流控制。MMX™ 比较指令以全 0 或全 1 的方式表示结果, 并以屏蔽位的方式参与结果的生成。这样可以避免局部转移指令, 增强了流水线的性能。

(4) MMX™ 指令集具有 RISC 的特点, 可按类似奔腾的指令配对原则配对, 在一个指令周期之内同时执行两条 MMX™ 指令。

• 本中引用的原始图象由陈向荣采集, 特致谢意。

收稿日期: 1997-01-24

(5) MMX™ 寄存器是对浮点寄存器的复用, MMX 指令不能与浮点指令同时使用。

### 3 MMX™ 程序开发过程

MMX™ 技术为提高应用程序的性能提供了新的可能性,但如何使用 MMX™ 开发应用程序则是一个新的课题。就目前的开发手段和条件看,为了最大限度地享受 MMX™ 的高性能,就必须付出辛勤的汗水,这是目前 MMX™ 技术最大的不足。我们在应用 MMX™ 开发应用程序的过程中对此有深切的体会。为了开发 MMX™ 的应用程序,必须按照以下步骤进行:

(1)先要找到一个支持 MMX™ 的编程环境,例如 VC++4.2;

(2)分析所用算法的浮点运算是否可以降低精度,用定点运算代替;

(3)调整数据流,使数据的空间分布便于 MMX™ 的并行数据通路沟通。具体地说,并行计算的数据必须位于相邻的空间位置。这对于图象或语音数据来讲是比较容易实现的;

(4)设计 MMX™ 并行通路,合并程序控制流程,用 MMX™ 指令取代奔腾指令;

(5)调整 MMX™ 指令序列,使之发挥最大效能;

(6)对于某些具体的应用可以利用 Intel 提供的 MMX™ 函数库,可以减少编程的困难。

### 4 边沿提取问题及应用 MMX™ 加速的快速算法

边沿提取是图象处理中最常遇到的问题。一个实际应用的边沿提取算法的主要性能指标是速度、精度和抗噪能力。

M0	M1	M2	M3	M4	M5	M6	M7
1 2 1	2 1 0	1 0-1	0-1-2	-1-2-1	-2-1 0	-1 0 1	0 1 2
0 0 0	1 0-1	2 0-2	1 0-1	0 0 0	-1 0 1	-2 0 2	-1 0 1
-1-2-1	0-1-2	1 0-1	2 1 0	1 2 1	0 1 2	-1 0 1	-2-1 0

用上面 M0~M7 8 个模板分别与图象匹配,取其中的最大值为梯度的大小,最大值的标号为梯度的方向。

### 4.1 传统的一阶边沿提取算法的主要问题

传统的基于一阶梯度的边沿提取算法由梯度算子,阈值和细化三部分构成。这一算法有以下缺点:

(1)先二值化再提取边缘,损失了梯度图象中的相对大小信息,令边沿的精度下降;

(2)边沿的检出依赖于阈值的选择,使算法对光照,对比度等因素极为敏感,尤其在同一张图上对比度变化比较大时,还会形成丢边,断边等错误;

(3)细化算法不能保证边沿的形态不变,会出现分叉等现象;

(4)算法的速度被迭代的细化算法所制约,且与阈值选择的优劣有关。

### 4.2 对边沿提取算法的改进

传统算法症结的核心在如何从梯度图象中细化出边沿。仔细分析可以发现,梯度图象比一般的图象包含更丰富的信息,即梯度的方向信息。利用这一条件,完全可以在未经二值化的梯度图象中直接提取边沿信息。算法如下:

设图象中某一点  $i$  的梯度为  $T_i$ ,梯度的方向为  $b_i$ ,  $i$  在  $b_i$  方向的最近邻点为  $i'$ ,在  $b_i$  负方向的最近邻点为  $i''$ ,则  $i$  为边沿的条件是:

$T_i > T_{i'}$  且  $T_i > T_{i''}$ 。该算法不仅极其简单,速度较传统算法有很大的提高,而且在保留了一阶边沿提取算法的优点(如抗噪能力强,无虚边等)的同时,在精度方面达到了二阶边沿提取算法的水平。

### 4.3 Robinson 梯度算子及快速算法

为了能运用改进的细化算法,选用的梯度算子必须能同时提取梯度的方向。此类的算子很多,我们选用了 Robinson 模板算子:

在 Robinson 算子当中,存在着许多重复的计算。可以用如下的快速算法实现:

(1)将  $M \times N$  的二维图象用一维数组  $I_{M \times N}$  表

示;

- (2) 计算  $H_i = I_i + I_{i+1}, V_i = I_i + I_{i+N}$ ,  
 $i = 0, 1, \dots, M \times N$ ;
- (3) 分别计算  $M0 \sim M7$ ,  
 $M0_i = H_{i-N-1} + H_{i-N} - H_{i+N-1} - H_{i+N}$   
:  
 $M7_i = H_{i-N} + V_{i-N+1} - H_{i+N-1} - V_{i-1}$
- (4) 在  $M0 \sim M7$  中选取最大值。

### 4.4 运用 MMX™ 技术加速

原图象的灰度范围是 0~255, 运算过程中多次超过字节数的表示范围, 只能利用 MMX™ 进行 4-并行加速。为了最大限度地发挥 MMX™ 技术的潜力, 加速后又手工进行汇编语言/流水线级优化。处理一幅 75K (320×240) 的图象, 在 120M 的 P55C 上实测的结果是:

未应用 MMX™	MMX™ 加速	优化后
0.1741s	0.0538s	0.0346s

MMX™ 的加速比大约 3.2~5.0。其中, 手工优化前的 MMX™ 代码的优化程度基本上可以与 C 语言编写的程序相当。

## 5 边沿交点提取算法及 MMX 快速实现

对于所有边沿提取算法来讲, 边沿交点的提取都是一个难题。然而, 边沿的交点往往是图象当中最重要的特征点, 在我们的三维视觉系统当中更是占有特别重要的地位。为了能准确地提取边沿的交点, 我们设计了一个新的模板, 从原始的灰度图象中直接提取。该模板是:

$$\begin{matrix} I_0 & I_1 & I_2 \\ I_7 & T_i & I_3 \\ I_6 & I_5 & I_4 \end{matrix}$$

$$T_i = |I_0 - I_1| + |I_1 - I_2| + |I_2 - I_3| + |I_3 - I_4| + |I_4 - I_5| + |I_5 - I_6| + |I_6 - I_7| + |I_7 - I_0|$$

即模板上相邻象素差值的绝对和。它实际上反映了图象中穿过象素邻域的边沿的条数和边沿的强度。为了有效地在模糊的图象中工作, 模板还可以扩大成 5×5 或 7×7。

### 5.1 快速算法及 MMX™ 快速实现

该算法加速的原理与 Robinson 算子加速算法

是完全一样的。

- (1) 计算  $H_i = |I_i - I_{i+1}|, V_i = |I_i - I_{i+N}|$   
 $i = 0, 1, \dots, M \times N$
- (2) 计算  $H'_i = H_i + H_{i+1}, V'_i = V_i + V_{i+N}$   
 $i = 0, 1, \dots, M \times N$
- (3) 计算  $T_i = H'_{i-N-1} + V'_{i-N+1} + H'_{i+N-1} + V'_{i+N+1}$   $i = 0, 1, \dots, M \times N$

兼顾到处理的精度和速度, 算法的 MMX™ 并行加速分别采用了 4-并行和 8-并行。加速的结果如下: (同一幅 75k 图象, 7×7 的模板)

未用 MMX™	MMX™ 4-并行	MMX™ 8-并行
0.1494s	0.0412s (手工优化后 0.0329s)	0.023s

MMX™ 的加速比大约 3.63~4.54 (4-并行) 及 6.518 (8-并行)。

### 5.2 局部极大值算法

经边沿交点提取算子处理过的图象中, 边沿的交点用局部极大值来表示。为了实现实时快速处理, 图象是未经预处理的, 包含了很大的噪声, 因此不能简单地提取局部极大值作为边沿的交点。为了达到抑制噪声影响的目的, 采用了一个较大模板的算法。

- (1) 提取所有的  $i$ , 满足  $T_i$  大于  $T_{i+k}$ ,  
 $k = 0, 1, \dots, 23$

0	1	2	3	4	5	6
23	.	.	.	.	.	7
22	.	.	.	.	.	8
21	.	.	$T_i$	.	.	9
20	.	.	.	.	.	10
19	.	.	.	.	.	11
18	17	16	15	14	13	12

- (2) 滤去面积小于 4 的点 (认为是噪声)
- (3) 以每一点块的重心作为提取的结果。

### 5.3 局部极大值算法的 MMX 快速实现

从表面上看来, 该算法对每一点要进行 24 次比较, 但实际上有半数左右的点只须比较一次就放弃了。当然也有的点当真要比较 24 次。因此, 这是一个严重不平衡的算法。从并行计算的角度, 有些点已经结束比较, 而另一些点却还要比较下去, 众口难调, 难于 MMX™ 并行处理。为了应用 MMX 技术, 必须将其写成一个平衡的算法。

(1) 计算  $H_i = \text{Max}(T_i, T_{i+1}), V_i = \text{Max}(T_i, T_{i+N}) \quad i=0, 1, \dots, M \times N$

(2) 计算  $H'_i = \text{Max}(H_i, H_{i+2}, H_{i+4}), V'_i = \text{Max}(V_i, V_{i+N}, V_{i+2N}) \quad i=0, 1, \dots, M \times N$

(3) 对点  $i$ , 提取的条件变成:

$$T_i > \text{Max}(H'_{i-3N-3}, V'_{i-3N+3}, H'_{i+3N-3}, V'_{i+3N+3})$$

这个算法的平均比较次数是 9 次, 比前面的不平衡算法慢很多。经 MMX 加速后的速度:

不平衡算法	平衡算法的 MMX™ 实现
0.0351s	0.0198s

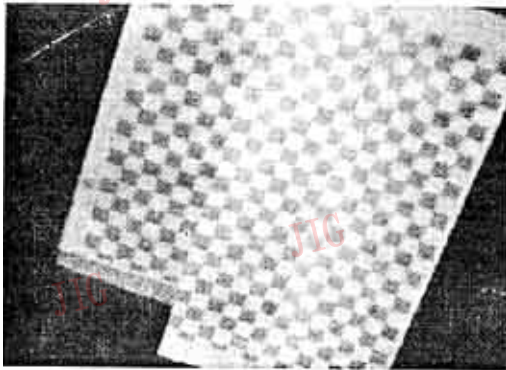


图 1 原图象

Fig.1 Original image

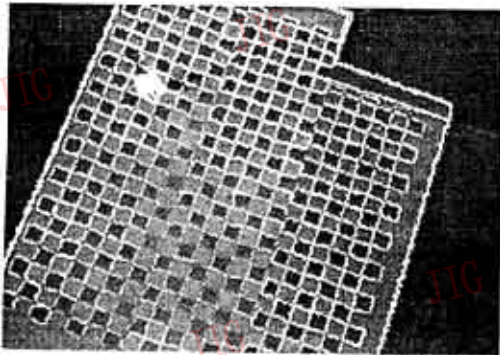


图 2 边缘和交点的提取

Fig.2 The result of edge and edge crossing point extraction

## 6 小 结

对于边沿提取这样的典型的图象处理问题应用了 MMX™ 技术, 可以在一定程度上体会 MMX™ 的加速性能。MMX™ 技术的引入, 使以往许多必须利用特殊的图象处理硬件才能完成的图象处理任务在微机上也可以完成。MMX™ 使微机进入多媒体时代的同时, 也将令图象处理进入微机时代。除此以外, 只要经过精心地编排, MMX™ 同样也可以应用到更加广阔的领域, 全面地提高微机系统的性能。然而, 目前 MMX™ 技术尚有许多不完善的地方。例如用 MMX™ 技术加速应用程序的过程基本上还处于手工编制阶段, 在 MMX™ 令应用程序的性能达到一个新的高度的同时, 也令编程的难度达到了一个新的高度。MMX™ 技术仍然是专家俱乐部的成员, 短时间内, 离大多数程序员仍有一定的距离。可以说, 现在 MMX™ 最需要的就是一种自动化的或至少是大众化的编程手段。如果我们能躺在长椅上, 听着音乐享受 MMX™ 编程的乐趣, 那么 MMX™ 带给我们的就不仅是一个个奇迹, 而是一个崭新的时代。

## 参 考 文 献

- 1 Intel Architecture MMX™ Technology Programmer's Reference Manual.
- 2 钟玉琢, 乔秉新, 李树青. 机器人视觉技术. 北京: 国防工业出版社, 1994. 11.
- 3 王积分, 张新荣. 计算机图象识别. 北京: 铁道科学出版社, 1987, 9.
- 4 吴立德. 计算机视觉. 上海: 复旦大学出版社, 1987, 9.



吴雨,清华大学计算机科学与技术系92级本科、97级硕士研究生。本科生期间参加过多项计算机视觉与多媒体领域的重大科研项目的工作。现在的主要科研方向是计算机视觉底层处理算法的设计与应用研究。

## Real-time Extraction of Edge and Edge Crossing by Using the MMX™ Technology

Wu Yu, Lin Xuying

(Dept. of Computer Science and Technology, Tsinghua University, Beijing 100084)

**Abstract** In this paper the MMX™ technology recently developed by Intel is briefly introduced and an algorithm of edge extraction and crossing point extraction by using the MMX™ technology is presented. By taking advantage of technology supported by MMX™ the speed of edge extraction can be performed at the rate of 30 frames/s.

**Keywords** MMX™ technique, Edge extraction, Intersection point extraction

### “千校用奥思”活动计划

方正奥思多媒体创作工具是中国科协副主席、两院院士王选教授领导的北大方正技术研究院历时五年潜心研究开发的成果,使用方正奥思,用户无须编程就可开发多媒体应用系统。多媒体教学是方正奥思的重要领域,它将用力带动我国多媒体教学的发展。

北大方正作为北京大学的校办产业,多年来蒙受党和国家各级领导的大力支持,在企业发展的同时,不断回报国家、回报教育事业是北大方正义不容辞的责任和义务。在方正奥思2.0版推出之际,我们与国家教委联合推出“千校用奥思”活动,这一活动得到国家教委各级领导和北京大学领导的大力支持

和帮助。

“千校用奥思”活动计划如下:

(1)赠送1000套方正奥思普及版给1000所学校免费使用。

(2)1000所学校使用方正奥思进行多媒体教学课程的开发。

(3)北大方正遍布全国的34家分公司为各地受赠学校提供技术支持与服务。

此外还将于今年秋季组织“全国多媒体教学软件大奖赛”,对于在多媒体教学课程开发方面取得成就的单位和个人给予奖励,这一活动也将得到国家教委的大力支持。