

# Line Object Extraction and Image Matching Based on an Intelligent Image Card

Han Xiao, Kui Yuan, Wen-hao He, Wei Zou

Institute of Automation Chinese Academy of Sciences, Beijing, China

(han.xiao@ia.ac.cn, kui.yuan@mail.ia.ac.cn, whhe@hitic.ia.ac.cn, wei.zou@mail.ia.ac.cn)

**Abstract**—A robot vision system is developed based on an intelligent image gathering card that contains an FPGA (Field Programmable Gate Array) and a DSP (Digital Signal Processor) as main calculators. Two real-time visual modules are developed through the cooperation of hardware logics on FPGA and software on DSP. First, with edge images extracted by the FPGA, a highly efficient algorithm is designed to extract a line object on the ground with DSP on real-time. Second, an image matching algorithm based on gray variance forms is designed, which is robust to illumination changing. Real-time computation is achieved by implementing this algorithm on FPGA, and object recognition is realized further.

**Keywords**—embedded systems; FPGA; DSP; image processing; robot vision; object tracking

## 基于智能图像卡的直线目标提取和图像匹配

肖晗, 原魁, 何文浩, 邹伟

中国科学院自动化研究所, 北京, 中国

**摘要** 本文在一块以一片现场可编程门阵列 (FPGA) 和一片数字信号处理器 (DSP) 为主要计算元件的智能图像采集卡上, 进行了机器人视觉系统的开发。通过 FPGA 内硬件逻辑和 DSP 上软件程序的结合, 开发了两个实时的视觉模块。第一, 在 FPGA 完成边缘图像提取的基础上, 设计了一种高效的算法, 用 DSP 实现了地面直线目标的实时提取。第二, 设计了一种基于灰度变化形态的图像匹配方法, 对光照具有鲁棒性, 利用 FPGA 的并行计算实时地实现了该匹配算法, 并在此基础上进行了目标识别。

**关键词** 嵌入式系统; FPGA; DSP; 图像处理; 机器人视觉; 目标跟踪

### 1. 引言

FPGA (Field Programmable Gate Array) 是一种可以通过硬件描述语言设计其逻辑功能的数字集成电路。由于开发方便, 并且能够实现高速的硬件计算和并行计算, 近年来, 人们一直在探索如何用 FPGA 来进行图像和视觉信息处理。

人们在 FPGA 上利用卷积网络 (Convolutional Network) 来进行人脸检测<sup>[1]</sup>, 采用块匹配算法实现了运动估计<sup>[2]</sup>, 并且实现了 SIFT 特征点检测算法的实时计算<sup>[12]</sup>。

FPGA 高速计算的能力和小巧的体积, 对实现实时的嵌入式图像信息处理系统具有极大帮助。然而, 单纯依靠 FPGA 的硬件计算, 有时还缺乏灵活性。为此, 人们将 FPGA 与微处理器结合起来, 以此来弥补硬件计算寻址能力的不

足。DSP (Digital Signal Processor) 作为一种常见的嵌入式微处理器, 可以担当这种角色。例如, 文献[13]用 FPGA 和 DSP 相结合, 实现了基于旋转不变相位相关算法的图像匹配的实时计算。

本文所介绍的工作, 是在一块针对移动机器人的视觉系统而设计的智能图像采集卡上进行的。该图像卡以 FPGA 和 DSP 为主要计算元件, 在采集图像的同时, 能够对图像进行处理, 实时地从中提取视觉信息。由于面向低成本应用, 该图像卡采用了低成本的 FPGA 和 DSP, 同时具有体积小、功耗低的特点。

本文介绍了在该图像卡上开发的两个视觉模块。在第一个模块中, FPGA 通过一种改进的 Canny 算子进行边缘检测。Canny 算子是一种性能优良的边缘检测算子, 何文

国家 863 项目 (资助号: 2008AA040204, 2008AA040209)

国家自然科学基金项目 (资助号: 60875051, 60705026)

浩<sup>[5]</sup>采用自适应阈值的方法对其进行改进，进一步提高了其性能，并在 FPGA 中实现了实时计算。在此基础上，我们将其应用在地面直线目标的识别上，设计了一种高效的算法，通过 DSP 的计算，从边缘图像中提取出目标信息，并通过视觉反馈实现了机器人对地面直线目标的跟踪。

在第二个视觉模块中，我们用 FPGA 实现了一种基于灰度变化形态的图像匹配。我们将灰度变化分成 4 种方向和 4 种类型，共 16 种形态。按照灰度变化的剧烈程度，对每一种形态都分为若干等级。在固定大小的窗口内，对所有像素处的灰度变化形态按照等级进行加权统计，得到 16 维的特征描述向量，以此为依据来进行图像块的匹配。由于统计的是灰度的相对变化，而不是灰度的绝对值，因此该方法对光照变化具有很好的鲁棒性。在 FPGA 内，通过窗口内对像素的并行操作和多级流水线模块的并行工作，在每一帧图像采集的同时能够实现实时的图像匹配。DSP 在此基础上，通过图像块之间相对位置的组合，进一步实现了目标识别的功能。

第 2 节介绍了该智能图像卡的工作原理。第 3 节介绍了机器人通过视觉反馈跟踪地面直线的实现。第 4 节介绍了在 FPGA 内完成的基于灰度形态统计的图像匹配及其在目标识别上的应用。第 5 节进行了总结。

## 2. 智能图像卡的工作原理

该智能图像卡的印刷电路板（PCB）设计者为路鹏<sup>[4]</sup>。文献[4]中对其性能特点进行了较详细的介绍。

计算元件采用了 Altera 公司 Cyclone 系列的 EP1C12Q240C8N 型号 FPGA，以及 TI 公司的 TMS320C5416 型号 DSP。两者均为低成本的器件。

图像卡的工作原理如图 1 所示。上电后，DSP 从 Flash（可断电保存数据的存储器）中加载程序开始运行，对整个流程起控制作用，并通过参数寄存器为 FPGA 的图像处理模块中的算法指定参数。

CCD 摄像头提供的模拟视频输入，经 AD 转换后，变成数字信号，送入 FPGA 内部。FPGA 内的图像处理模块

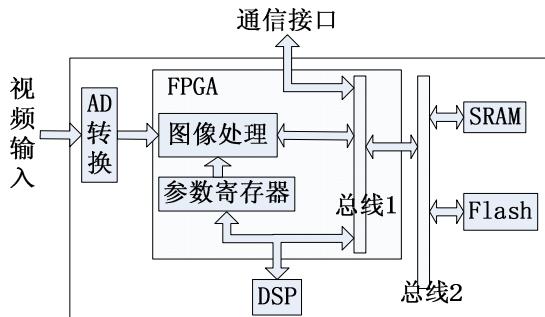


图 1 智能图像采集卡的原理图

通过窗口操作和流水线结构来实现并行操作，在图像信号采集的同时进行处理，并将处理结果写入 SRAM（静态随机存储器）。在一帧图像采集结束后，DSP 从 SRAM 中将 FPGA 的处理结果读入自身内部的存储器，进一步提取高层视觉信息。与此同时，FPGA 继续采集和处理下一帧图像。从而，FPGA 和 DSP 两者实现了并行工作。

本系统处理的视频图像为 PAL 制式，可以实现 25 帧/秒的实时处理。标准 PAL 制式的数字图像为  $720 \times 576$  大小，隔行扫描，分为奇偶两场，先传送奇场，后传送偶场。我们的系统只对奇场进行处理，再加上 FPGA 内流水线的需要，所以实际处理的图像大小为  $360 \times 284$ 。

由 ARM9 控制器构成的机器人上位机，可以通过通信接口访问 DSP 片内的存储器。DSP 在处理完每一帧图像后，向上位机发出一个中断，通知上位机来读取处理结果。

## 3. 机器人跟踪地面直线

作为一个应用，我们以该嵌入式图像处理卡为基础，构造了一个视觉反馈系统，用来指导机器人跟踪地面上一条固定的直线。如图 2 所示，四轮移动机器人上装备的摄像头朝向前下方，用于提取地面上具有一定宽度的直线目标，后文称之为“带状目标”。

FPGA 对每一帧图像都通过改进的 Canny 算子提取边缘，并能够在采集图像的同时，实时地完成。具体细节见文献[5]。

经过 FPGA 的处理，得到的是一幅边缘图像。FPGA 将处理结果以紧缩的格式存储在图像卡的 SRAM 中，即：边缘像素记为 1，非边缘像素记为 0，这样 SRAM 中每一个 16 位的字存储了 16 个像素的值。

进一步的高层信息提取，由 DSP 完成。图 3(a)为机器人视野中的一幅边缘图像，在其中定义了带状目标的偏距  $x$  和偏角  $\theta$ 。DSP 所要求取的就是这两个参数。

如图 3(b)所示，我们用 7 条横线来对图像进行等分，DSP 仅沿着这 7 条横线搜索边缘图像中的像素。7 条横线与边缘像素的交点，在图中用字母标注出来。DSP 搜索完 7 条横线后，依据这些交点的位置，即可完成对带状目标的提取。



图 2 机器人跟踪地面直线

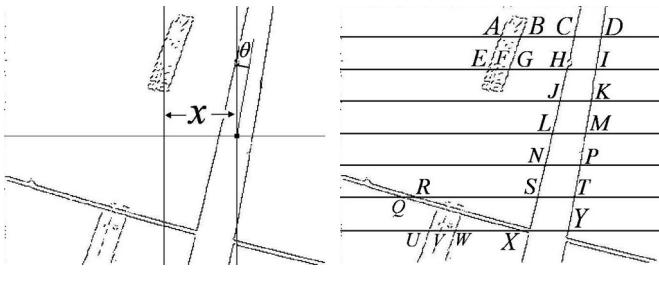


图 3 边缘图像中的带状目标

### 3.1 带状目标的判别依据

依据以下三个条件来判断带状目标：

- (1) 带状目标由两条相邻的直线构成。
- (2) 带状目标的宽度位于合适的范围内。
- (3) 构成带状目标的两条直线上的点应当尽可能地多。

首先通过点的共线检测来初步提取备选的带状目标，进而通过宽度条件进行筛选，最后选取点数最多的一个带状目标作为最终的检测结果。在图 3(b)中，显然希望检测到的带状目标由 13 个点构成，其中 C,H,J,L,N,S 共线，D,I,K,M,P,T,Y 共线。

### 3.2 计算方法

依据判别条件 (1)，在图 3(b)中，前三条横线上的相邻点共有 12 种可能的组合，见表 1。

对每一种组合，进行两个共线检测。例如，对 {AB,GH, JK} 这个组合，需要检测 A, G, J 三点是否共线，B, H, K 三点是否共线。

通过向量叉积的计算，可以很方便地进行共线检测。例如，用所在的行和列来表示点的坐标，则 A, G, J 三个点的坐标分别为 A(35, 182), G(71, 191), J(107, 249)。

$$\text{向量 } \vec{AG} = \{71 - 35, 191 - 182\} = \{36, 9\}$$

$$\text{向量 } \vec{GJ} = \{107 - 71, 249 - 191\} = \{36, 58\}$$

表 1 前三条横线上相邻点所有可能的组合

{AB, EF, JK}	{BC, EF, JK}	{CD, EF, JK}
{AB, FG, JK}	{BC, FG, JK}	{CD, FG, JK}
{AB, GH, JK}	{BC, GH, JK}	{CD, GH, JK}
{AB, HI, JK}	{BC, HI, JK}	{CD, HI, JK}

$$\text{叉积 } \vec{AG} \times \vec{GJ} = \begin{vmatrix} 36 & 9 \\ 36 & 58 \end{vmatrix} \vec{n} = 1764 \vec{n}$$

其中  $\vec{n}$  代表垂直于纸面的单位向量。

而对于 D, I, K 三个点，D(35, 294), I(71, 288), K(107, 283)，向量叉积

$$\vec{DI} \times \vec{IK} = \begin{vmatrix} 36 & -6 \\ 36 & -5 \end{vmatrix} \vec{n} = 36 \vec{n}$$

事实上，叉积的模为

$$|\vec{DI} \times \vec{IK}| = \|\vec{DI}\| \cdot \|\vec{IK}\| \sin \beta$$

其中  $\beta$  代表  $\vec{DI}$  和  $\vec{IK}$  的夹角。不难看出，当 D, I, K 严格共线时， $\beta = 0$ ，从而叉积为零。当存在误差时， $\beta$  是一个接近于零的数，由于  $|\vec{DI}|$  和  $|\vec{IK}|$  有界，因此叉积的模仍然是一个很小的数。

经过实验，我们选取 150 来作为共线检测的阈值。只要叉积的模小于 150，就认为三点共线。该阈值既能够容许误差，又不会出现误检测。

对表 1 所列出的 12 种组合逐一进行共线检测后，可以将最后一种组合 {CD, HI, JK} 所对应的带状目标提取出来。再经过 3.1 节中判别条件 (2) 和 (3) 的进一步确认，可以唯一确定由直线  $\overline{CHJLNS}$  与直线  $\overline{DIKMPTY}$  所构成的带状目标。进一步，可以提取出图 3(a) 中所定义的偏距  $x$  和偏角  $\theta$  两个参数。其中，提取偏角  $\theta$  时，利用像素坐标差与  $\theta$  值的对应关系，通过查表可实现快速计算。

### 3.3 控制率

将图 3(a) 中定义的偏距  $x$  和偏角  $\theta$  作为反馈变量，对机器人运动的线速度  $v$  和角速度  $\omega$  进行控制。只要使偏距  $x$  和偏角  $\theta$  都收敛到零，即可实现机器人对带状目标的跟踪。

采用简单比例负反馈，即可实现该控制目标：

$$\begin{cases} \omega = k(-x - b\theta) \\ v = v_0 \end{cases}$$

其中  $k, b, v_0$  均为大于零的常数。 $k, b$  为比例系数， $v_0$  为一个恒定的线速度。三者的数值均通过实验来确定。

图 4(1)~(6) 显示了在室外进行实验时，机器人向目标收敛的过程。

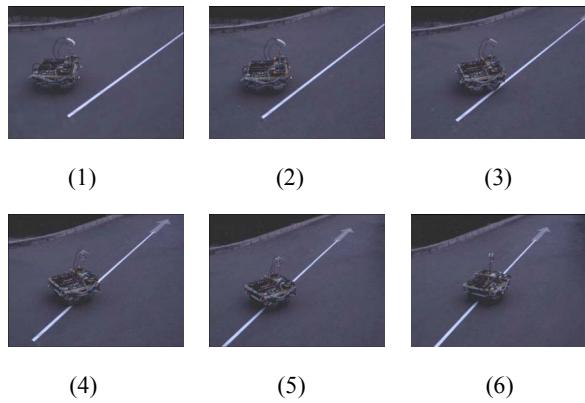


图 4 机器人跟踪直线目标的室外实验

#### 4. 图像匹配和目标识别

图像匹配的简单定义就是：给定一个小图像块，从整幅图像中寻找与该小图像块足够相似的局部区域。给定的小图像块通常称作“匹配模板”。

人们对图像匹配的研究大多是基于灰度图像的，每一个像素具有 0~255 之间的一个整数亮度值。匹配时，一个很关键的问题就是，如何对“相似”进行度量。最常用的是 SAD 方法（Sum of Absolute Difference），就是将对应位置像素的亮度值求差，求绝对值，最后再总起来求和。

文献[1]用 FPGA 来实现基于 SAD 的图像匹配。文献[2]在 FPGA 上通过基于 SAD 的图像匹配来进行光流场和立体视觉的计算，并进而应用于运动目标的实时检测。文献[3]在 FPGA 上通过 ZSSD（zero mean sum of squared differences，零均值像素亮度差的平方和）方法进行图像匹配，获得了比 SAD 更好的效果。

为了获得对光照的鲁棒性，我们设计了一种基于灰度形态统计的图像匹配方法。以这种方法为基础，在 FPGA 上实现的基于图像匹配的目标识别功能，能够在光照条件变化很大的情况下实现对同一个目标的识别。

##### 4.1 算法原理和系统结构

对于摄像头采集的 PAL 制式的彩色视频信号，FPGA 只提取 YUV 颜色空间中的 Y 分量（亮度分量），从而得到灰度图像。在此基础上，通过 Gauss 滤波对图像进行平滑。输出平滑后的结果时，进行了一个从亮度值到灰度值的转化。我们定义，灰度值 G 为亮度值 Y 的取反，即

$$G = 255 - Y$$

在紧接着的流水线模块中，开辟一个  $3 \times 3$  的窗口，进行灰度变化形态的计算。如图 5 所示， $3 \times 3$  窗口的中心像

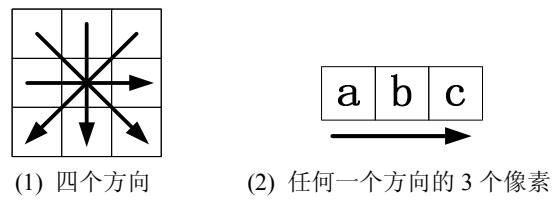


图 5 每个像素具有四个可能的灰度变化方向

素具有四个灰度变化方向，每个方向都可以提取出三个像素来。如图 6 所示，三个像素共可以构成四种灰度变化类型，将其形象地称为上坡、下坡、山峰、山谷。通过对三个像素灰度值大小的比较来判断其所属的灰度变化类型。例如，中间像素比两边像素的灰度值都大，则构成山峰。在此基础上，进一步判断灰度变化的等级。方法是，将三个像素中最大的灰度值与最小的灰度值相减，将结果按照大小分成 16 个等级。

$3 \times 3$  窗口中心像素的四个灰度变化方向及相应的类型和等级，在 FPGA 内通过并行计算，同时计算出来。下一级流水线子模块从中选出等级最高的一个方向，用此方向上的灰度类型来描述该像素处的灰度变化。

接下来的模块开辟了一个  $7 \times 7$  的窗口，对灰度变化形态进行统计。 $7 \times 7$  的窗口中共有 49 个像素位置，每一个位置都具有其灰度变化的方向、类型和等级。4 种方向、4 种类型，共有 16 种组合，即 16 种灰度变化形态。按照等级进行加权统计，得到 16 维的特征描述向量。

最后一级流水线模块依据此 16 维的向量来进行图像匹配。匹配模板由两个 16 维向量构成，分别指定了上界和下界，记为 M1 和 M2。当 FPGA 内的统计窗口在图像中某个位置提取出来的 16 维向量位于 M1 和 M2 之间时，就实现一个匹配。FPGA 将该位置的行、列坐标写入 SRAM 中，同时继续检测新的位置。

以上各级流水线模块在 FPGA 中同时工作，实现了时间上的并行性。图 7 显示了各级流水线的顺序。

匹配模板作为 FPGA 的工作参数，由 DSP 写入 FPGA 内部的参数寄存器，在 FPGA 采集处理每一帧图像之前都可以进行修改。初始的匹配模板可以通过离线计算得到。

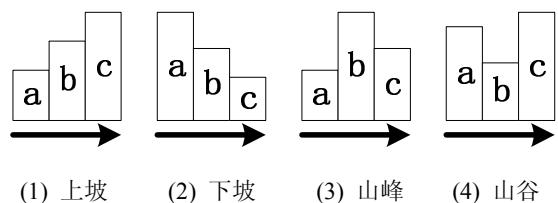


图 6 四种灰度变化类型

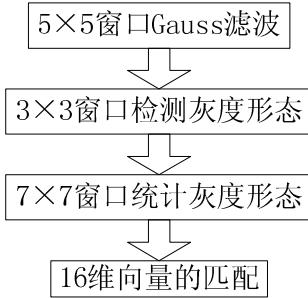


图 7 FPGA 内部的流水线

如图 8 所示, 选定图像中的目标区域后, 将其分解为 9 个图像块, 对每个图像块都可以提取出一副模板。

一副匹配模板能检测到的位置从几个到几十个不等。也就是说, 图 8(c)中每一个图像块在图像中都能检测到若干个匹配位置, 其中只有一个是在目标区域内的位置。通过图像块之间相对位置的组合, 可以唯一确定目标区域。

DSP 所要完成的就是这样的任务。在每一帧图像采集处理前, DSP 向 FPGA 内写入一副模板。FPGA 处理完该帧图像后, 将所有匹配位置的行、列坐标值写入 SRAM 中。这样, FPGA 在不同的帧中检测不同的图像块。在检测完多个图像块后, DSP 通过图像块之间的相对位置来最终锁定目标区域。

我们采用相同的模板, 在不同的光照条件下进行了实验(见图 9)。目标设定为图 8(b)所标注的靶形目标。实验结果表明, 在图 9 的六种不同光照条件下, 系统均实现了对该目标的稳定识别。



图 8 图像匹配的目标区域及其分解



图 9 不同的光照条件

在我们所开发的这个系统中, FPGA 在输出匹配位置坐标值的同时, 将实现匹配的 16 维向量也输出到 SRAM 中。该功能最初是用作调试的。亦可以用它来对模板进行在线修正。DSP 通过读取 FPGA “反馈”回来的 16 维向量, 调整模板的上下界, 以实现更鲁棒的目标检测和跟踪。

#### 4.2 资源消耗和实时性

在 FPGA 内, 用  $7 \times 7$  的窗口进行灰度变化统计时, 如果对窗口内 49 个像素位置全部进行并行操作的话, 将消耗大量的逻辑资源。考虑到窗口滑动时, 内部具有稳定的部分, 所以设计时采用了累积的方法, 只对左右两侧 14 个像素位置进行统计, 即只对  $7 \times 7$  窗口内变化的部分进行增减, 如图 10 所示。采用该方法, 大大节约了硬件资源。

整个系统已通过硬件验证。FPGA 的资源消耗情况如表 2 所示。

FPGA 内的处理模块在 27MHz 的 PAL 视频信号的时钟驱动下工作。DSP 工作在 80MHz 的时钟频率下。

目前只采用了一副匹配模板, 每一帧只检测一个图像块。从表 2 可以看出, 资源还有较大余量, 还可以增加几副模板, 来进行多个图像块的并行检测。只需对图 7 中最后一级流水线模块进行扩展即可。

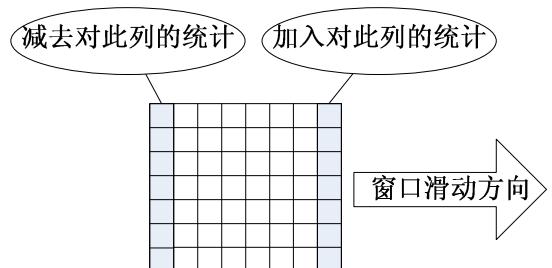


图 10 对  $7 \times 7$  窗口的统计技巧

表 2 图像匹配算法在 Altera 公司 Cyclone 系列 EP1C12Q240C8N 型号 FPGA 上的资源消耗

	LEs	Pins	Memory bits	PLLs
总数	12060	173	239616	2
消耗量	6945	79	34234	1
百分比	58%	46%	14%	50%

实时性与此密切相关。实验表明，用两个图像块即可锁定一个目标，但不够稳定。用三到四个图像块即可稳定地跟踪一个目标。DSP 上的计算所花时间甚少(不到 1ms)，系统的实时性完全取决于 FPGA 从每一帧图像中能并行检测几个图像块以及用几个图像块来跟踪目标。

## 5. 总结

我们在一个以 FPGA 和 DSP 为主要计算元件的体积小、成本低、功耗低的智能图像卡上开发了两个机器人视觉系统应用模块。实现了机器人对地面直线的跟踪，以及通过图像匹配来识别目标的功能。

算法设计、硬件设计、软件设计三者相结合，是本文的研究特点。算法本身要面向应用的需求，同时又要考虑硬件和软件的特点，通过合理的软硬件分工，来实现实时的计算。在直线目标跟踪模块中，边缘检测的任务由 FPGA 的并行计算来完成。通过设计一种高效的算法，使得 DSP 能够在此基础上实时地完成对直线目标的提取。在图像匹配模块中，设计了一种描述灰度变化的方法来获得对光照的鲁棒性。在此基础上，进行了 FPGA 和 DSP 的分工，实时的图像匹配任务由 FPGA 的并行计算来完成，DSP 负责匹配模板的指定、调节以及目标的识别。

## 参考文献

- [1] K. Nakano and E. Takamichi, "An image retrieval system using FPGAs," in *Design Automation Conference, 2003. Proceedings of the ASP-DAC 2003. Asia and South Pacific*, 2003, pp. 370-373.
- [2] H. Niituma and T. Maruyama, "Real-Time Detection of Moving Objects," in *Field Programmable Logic and Application*. Berlin/Heidelberg: Springer, 2004, pp. 1155-1157.
- [3] 周文晖, 等, "基于 FPGA 的双目立体视觉系统," *中国图象图形学报*, vol. 10, no. 9, 2005.
- [4] 路鹏, 原魁, 邹伟, "一种高性能低功耗的移动机器人视觉处理系统," *高技术通讯*, vol. 17, no. 8, pp. 803-808, 2007.
- [5] 何文浩, 原魁, 邹伟, "自适应阈值的边缘检测算法及其硬件实现," *系统工程与电子技术*, vol. 31, no. 1, pp. 233-237, 2009.
- [6] *TMS320VC5416 Fixed-Point Digital Signal Processor*. Texas Instruments, 2002.
- [7] *Cyclone FPGA Family Data Sheet*. Altera, 2003.
- [8] *SAA7113H Data Sheet*. Philips Semiconductors, 1999.
- [9] *IS61LV51216 Data Sheet*. Integrated Silicon Solution, Inc., 2005.
- [10] *SST39VF400A Data Sheet*. Silicon Storage Technology, Inc., 2001.
- [11] C. Farabet, et al., "CNP: An FPGA-based Processor for Convolutional Networks," in *NIPS08 Workshop*, 2008.
- [12] V. Bonato, E. Marques, and G.A. Constantinides, "A Parallel Hardware Architecture for Scale and Rotation Invariant Feature Detection," *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY*, vol. 18, no. 12, 2008.
- [13] 吴四夫, 付宇卓, 邓红梅, "基于相位相关算法的实时图像匹配系统," *计算机仿真*, vol. 22, no. 11, 2005.