

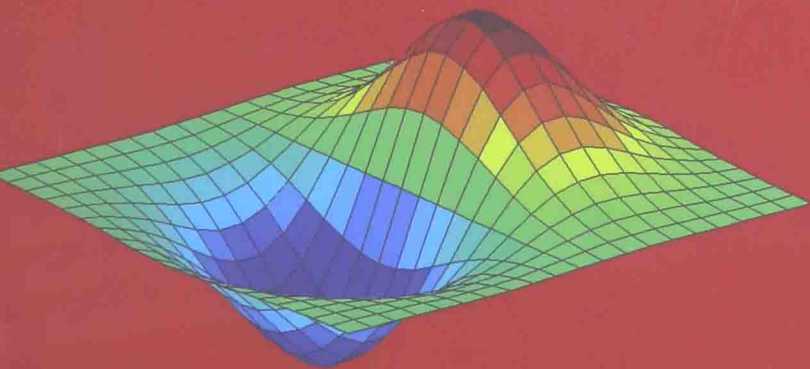


国内首本基于MATLAB与CUDA的数字图像处理高级应用的著作
本书瞄准前沿热点、理论讲解透彻、代码注解详细
MATLAB中文论坛与作者在线交流

ADVANCED DIGITAL IMAGE PROCESSING APPLICATION BASED ON MATLAB AND CUDA

数字图像处理 高级应用

基于MATLAB与CUDA的实现



- 基于模型的图像边缘检测
- 雾霾天气图像增强技术及其实现
- 基于运动估计的视频帧频插帧
- SIFT特征提取与描述
- SURF特征提取与匹配
- 基于余弦变换的多聚焦图像融合
- 基于并行计算的图像处理
- 图像并行处理环境构建
- 从图像处理算法到GPU代码实现
- 使用纹理存储器加速
- 基于CUFFT函数库的图像频域变换
- 自适应直方图均衡
- RGB图像的GPU并行处理

赵小川 何灏

Zhao Xiaochuan He Hao

吴军 李利华 马子领

Wu Jun Li Lihua Ma Ziling

◎ 编著

清华大学出版社



ADVANCED DIGITAL IMAGE PROCESSING APPLICATION BASED ON MATLAB AND CUDA

数字图像处理 高级应用

基于MATLAB与CUDA的实现

赵小川 何灏

Zhao Xiaochuan He Hao

吴军 李利华 马子领

Wu Jun

Li Lihua

Ma Ziling

◎ 编著

清华大学出版社

内 容 简 介

本书主要介绍现代数字图像处理的高级应用,内容包括热点图像处理方法原理及其 MATLAB 仿真、并行计算及其在图像处理中的应用、图像并行处理环境构建、典型图像处理方法的 CUDA 实现。本书的主要特色为瞄准前沿热点、理论讲解透彻、代码注解详细。

本书可作为高等院校电子信息、计算机相关专业本科生、研究生的教材,也可作为毕业设计、研究生课题、各类相关竞赛的参考资料,还可作为相关工程技术人员的参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

数字图像处理高级应用:基于 MATLAB 与 CUDA 的实现/赵小川等编著.--北京:清华大学出版社,2015

ISBN 978-7-302-38488-5

I. ①数… II. ①赵… III. ①数字图像处理—MATLAB 软件 IV. ①TN911.73

中国版本图书馆 CIP 数据核字(2014)第 260987 号



责任编辑:刘 星

封面设计:李召霞

责任校对:李建庄

责任印制:宋 林

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 刷 者:三河市君旺印务有限公司

装 订 者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:12.75 字 数:317千字

版 次:2015年4月第1版 印 次:2015年4月第1次印刷

印 数:1~2500

定 价:39.00元

产品编号:060383-01

当前,信息处理技术和计算机技术取得了长足的进步,数字图像处理技术已在工业检测、航空航天、星球探测、军事侦察、公安防暴、人机交互、文化艺术等领域受到了广泛的重视并取得了众多成果。在人工智能、思维科学、仿生学等新兴学科的推动下,现代数字图像处理技术正在向着更高、更深层次发展,实用性也日渐增强。

随着现代数字图像处理技术的飞速发展,新理论、新方法、新技术层出不穷,并呈现出智能化、高效化、实用化的特点。笔者在结合自身教学与科研的基础上,对这些新理论、新方法、新技术及其实现过程与方法进行了梳理和总结,撰写成书。

与其他同类的数字图像处理书籍相比,本书有以下特点:

(1) 瞄准前沿

本书所介绍的图像去雾技术、对图像去雾技术、尺度不变特征点的提取、图像融合、视频图像运动估计、基于 CUDA 的图像并行处理等内容均是现代数字图像处理技术的前沿和研究热点。通过对这些知识与技术的讲解,便于读者了解现代数字图像处理领域的研究热点和最新研究动向。

(2) 代码翔实

本书所讲解的相关理论、算法、技术配有代码实现,并且对这些代码进行了详细的注解。读者可通过阅读代码,对本书讲解的内容进行更加深入的了解。

本书共四章,前两章为提高篇,后两章为应用篇。内容包括:热点图像处理方法原理及其 MATLAB 仿真、并行计算及其在图像处理中的应用、图像并行处理环境构建、典型图像处理方法的 CUDA 实现。

本书的适应人群是:

- 对数字图像技术感兴趣的读者;
- 电子信息工程、计算机科学技术相关专业的本科生、研究生的教材;
- 本科毕业设计、研究生学术论文的资料;
- 相关工程技术人员的参考资料。

在本书的撰写过程中,刘祥对第 1 章进行了程序调试与文字校对,李阳对第 2 章进行了程序调试与文字校对,梁冠豪对第 3 章进行了程序调试与文字校对,刘瑞瑞对第 4 章进行了程序调试与文字校对。在此,对他们的辛勤劳动表示衷心的感谢。

书中疏漏或不足之处在所难免,敬请读者批评指正。有兴趣的朋友可以发邮件到 zhaoxch1983@sina.com,与作者进行交流。

赵小川

2015 年 3 月

第 1 章 热点图像处理方法原理及其 MATLAB 仿真	1
1.1 基于模型的图像边缘检测及其代码快速生成	1
1.1.1 模型驱动开发思想概述	1
1.1.2 模型驱动开发的优势	2
1.1.3 模型驱动开发在图像处理领域中的应用	3
1.1.4 基于 Simulink-Blocks 的模型驱动开发图像处理	3
1.1.5 基于 Sobel 算子的边缘检测的基本原理	3
1.1.6 基于模型的实现	5
1.1.7 代码的快速生成	7
1.2 从向量相关角度实现图像匹配	9
1.2.1 基于相关运算图像匹配的过程	9
1.2.2 在向量空间分析图像相关匹配	10
1.2.3 基于向量相关的图像匹配的 MATLAB 实现	10
1.3 雾霭天气图像增强技术及其实现	12
1.3.1 Retinex 理论	12
1.3.2 基于 Retinex 理论的图像增强的基本步骤	13
1.3.3 多尺度 Retinex 算法 (MSRCR 算法)	13
1.3.4 例程精讲	14
1.4 基于运动估计的视频倍频插帧	23
1.4.1 运动估计简介	23
1.4.2 运动估计的应用领域	24
1.4.3 运动估计方法分类	24
1.4.4 基于块匹配方法的运动估计	25
1.4.5 相关概念	26
1.4.6 匹配方法: 价值函数	27
1.4.7 搜索算法	28
1.4.8 实际应用举例	33
1.5 SIFT 特征提取与描述	42
1.5.1 SIFT 算法	43
1.5.2 SIFT 特征描述	47
1.5.3 实例精讲	49
1.6 SURF 特征提取与匹配	69

目录

1.6.1	积分图像	69
1.6.2	DoH 近似	70
1.6.3	尺度空间表示	72
1.6.4	SURF 特征描述算子	73
1.6.5	程序实现	77
1.7	基于余弦变换的多聚焦图像融合	82
1.7.1	什么是图像融合	82
1.7.2	图像融合的层次	84
1.7.3	多聚焦图像融合	85
1.7.4	程序实现	87
第 2 章	并行计算及其在图像处理中的应用	91
2.1	并行计算的理论知识	91
2.1.1	如何理解并行计算	91
2.1.2	并行计算的发展	92
2.1.3	并行计算的各类实现方式	98
2.1.4	并行计算机常用编程工具	99
2.1.5	什么是 GPU	100
2.2	基于并行计算的图像处理	102
2.2.1	基于并行计算的图像处理的优点	103
2.2.2	效能评价标准	104
2.2.3	图像处理并行设计方法	107
2.2.4	需要考虑的一些问题	108
第 3 章	图像并行处理环境构建	111
3.1	建立一个简单的对话框工程	112
3.2	用 OpenCV 显示 Hello World	114
3.3	安装配置 CUDA 环境	117
3.4	用 CUDA 进行并行图像处理	119
3.5	使用 OpenCV 读入、处理和显示图像	121
3.6	CUDA 编程简介及其在图像处理中应用	122
3.6.1	主机端和设备端	123
3.6.2	内核函数、CUDA 软件体系和 NVCC 编译器	125
3.6.3	CUDA 线程模型的层次结构	127

3.6.4 GPU 组成结构及其与线程模型的关系	130
3.6.5 SDK 和函数库	131
第 4 章 典型图像处理方法的 GPU 实现	133
4.1 基于 kernel 函数的并行图像处理的程序实现	134
4.2 从图像处理算法到 GPU 代码实现	145
4.2.1 模板匹配的 MATLAB 代码	145
4.2.2 模板匹配的 CPU 串行实现	146
4.2.3 模板匹配的 GPU 并行实现	149
4.3 使用纹理存储器加速	154
4.3.1 Sobel 算子边缘提取的 GPU 实现	154
4.3.2 双线性插值图像缩放的 GPU 实现	159
4.4 基于 CUFFT 函数库的图像频域变换	162
4.4.1 图像 FFT 变换和频域低通滤波	163
4.4.2 频域高通滤波	166
4.4.3 多尺度 Retinex 去雾算法	170
4.5 自适应直方图均衡	173
4.6 RGB 图像的 GPU 并行处理	178
4.6.1 彩色图像去雾算法	180
4.6.2 彩色图像去雾和直方图均衡	183
4.6.3 K 最近邻滤波器	185
4.6.4 非局部均值滤波器	190
参考文献	195

1.1 基于模型的图像边缘检测及其代码快速生成

1.1.1 模型驱动开发思想概述

基于模型设计方法是一种设计复杂系统的可视化数学方法。基于模型的软件开发方法是一种以模型为中心的软件设计方法,用模型来形式化表示用户所构建系统的功能、行为和结构。形式化的意思是描述模型的语言具有定义良好的语法和语义。基于模型设计思想是通过模型抽象和描述系统的关键问题,用模型表示出系统解决方案,并通过形式化的工具对系统模型进行验证,以保证系统设计正确性。基于模型设计思想从根本上解决了软件设计的正确性保证问题。

什么是模型?模型可以是具体的物理实体,也可以是抽象的描述,如图像描述、数学表达式。模型特征是对对应的物理系统或者非物理系统的一种简化表示,是人们的一种有意识的创造物。有了模型,系统开发人员就可以依赖模型来构建系统,而忽略掉无关的细节。

系统建模和模型仿真验证是基于模型设计方法的两个重点。采用特定的建模语言和模型设计工具,来抽象出被描述的目标系统,从而建立准确的模型,这样可以消除其他系统描述方式(比如文本描述)的模糊性和二义性。仿真验证毫无疑问需要基于建立好的准确的目标模型,而仿真验证带来的好处是通过相应的仿真验证工具对模型进行仿真验证之后可以保证模型的正确性。

模型驱动架构(Model Driven Architecture,MDA)是2002年被提出的,包含了模型的精确形式化表示、模型存储以及模型交换等方面的一系列规约,如MOF(Meta Object Facility)、OCL(Object Constraint Language)、XMI(XML Metadata Interchange)等,旨在提高程序设计的抽象层次,提高软件开发效率,增强软件的可移植性、协同工作能力和可维护性。要求设计时进行严格的模型定义,力求最终能全部或部分自动生成工程代码。

MDA的软件开发过程是由软件系统建模驱动的。模型根据抽象

层次不同可分为：平台无关模型 PIM 和平台相关模型 PSM。平台无关模型描述系统的行为需求不涉及平台实现细节，平台无关模型的设计者专注于表示系统的业务逻辑；平台相关模型不仅包含系统功能描述，而且包含实现平台的设计思想。平台无关模型可以被转换为一个或多个平台相关模型，平台相关模型最终会转换成具体平台的实现代码。MDA 软件开发周期如图 1.1.1 所示。



图 1.1.1 MDA 软件开发周期

MDA 引入了模型驱动开发 (Model-Driven Development, MDD) 的软件开发和模型集成运算 (Model-Integrated Computing, MIC) 的开发模式。模型驱动开发是当今嵌入式系统开发方法的热点，模型集成运算开发方法是面向领域的模型驱动开发方法，在面向特定领域的建模语言方面扩展了模型驱动开发方法，能通过元模型针对特定领域进行定制和裁剪，并能基于领域模型自动生成代码。

代码生成的概念最初见于编译器设计和开发中，编译器先将输入的源程序翻译成某种中间表示，然后产生等价的某种目标程序语言表示的目标程序。这种将中间表示生成目标程序的过程也是代码生成，但是在基于模型的软件设计方法中的代码生成是指利用计算机程序(代码生成器)将模型自动地生成可用的代码。代码生成的过程大致包括：模型建立和生成代码。建立一个好的能准确描述应用系统并充分考虑系统应用环境的模型并没有解决所有问题，因为必须将这个模型转化成有效的可运行的模型代码。

基于模型的开发方法和传统的软件开发方法有着显著的区别。传统的软件开发方法如图 1.1.2 所示，其核心是代码，代码的开发、测试和维护占用项目开发的绝大多数时间和精力。而基于模型的开发方法，如图 1.1.3 所示，其核心是模型，项目开发的主要工作是模型的设计，测试和维护也是围绕着模型来开展的，其代码可以方便地依据模型而自动生成。代码生成可以很好地提升软件开发效率、缩短软件开发周期、提高软件开发效率。

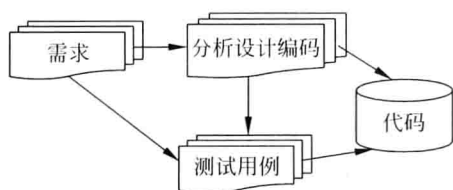


图 1.1.2 传统的以代码为中心的软件开发方法

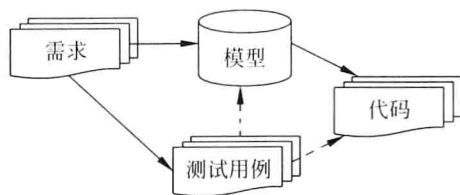


图 1.1.3 以模型为中心的软件开发方法

1.1.2 模型驱动开发的优势

基于模型驱动的设计开发在各个过程之间没有阻碍，它的优势在于：

(1) 在统一的开发、测试平台上，允许产品从需求分析阶段就开始验证，并做到持续不断的验证与测试；

- (2) 产品的缺陷暴露在产品开发的初级阶段,开发者把主要精力放在算法和测试的研究上,代码生成和验证过程则留给计算机自动完成;
- (3) 大大缩短开发周期,降低开发成本。

1.1.3 模型驱动开发在图像处理领域中的应用

众所周知,数字图像处理的实现过程中代码量巨大,将模型驱动开发引入到图像处理领域,则可大大提高其规范性和高效性。

基于模型驱动的数字图像处理的主要步骤包括:

- (1) 需求分析;
- (2) 将实现过程按照逻辑先后顺序分为各个阶段,再将各个阶段按照功能划分为各个模块;
- (3) 建立基于功能模块的模型,并进行仿真;
- (4) 代码自动生成。

1.1.4 基于 Simulink-Blocks 的模型驱动开发图像处理

MATLAB 中的计算机视觉系统工具箱(Computer Vision System)如图 1.1.4 所示,提供了视频和图像处理的各种模型,共计 11 个大类库,每个模型库提供了数种模块。用户可以通过拖动、组合,构建视频和图像处理模型,进行视频和图像的仿真和分析。

启动 MATLAB R2012a,选择界面左下角的 Start,按图 1.1.4 所示选择 Toolboxes → Computer Vision System → Block Library,系统就会载入视频和图像处理模块工具箱,如图 1.1.5 所示。视频和图像处理模块包含 Sources(输入模块)、Sinks(输出模块)、Analysis & Enhancement(分析和增强)、Conversions(转换)、Filtering(滤波)、Geometric Transformations(几何变换)、Morphological Operations(形态学运算)、Statistics(统计)、Text & Graphics(文本和图像)、Transforms(变换)和 Utilities(自定义)11 个大类库。这几乎包含了图像处理中的所有操作和算法,并附带了文字标注子模块,为图像处理的模型建立和仿真提供了充足的模块。

1.1.5 基于 Sobel 算子的边缘检测的基本原理

边缘是图像最基本的特征。边缘检测在计算机视觉、图像分析等应用中起着重要的作用,是图像分析与识别的重要环节。因为图像的边缘包含了用于识别的有用信息,所以边缘检测是图像分析和模式识别的主要特征提取手段。

所谓边缘,是指其周围像素灰度阶跃变化或屋顶状变化的那些像素的集合,它存在于目标与背景、目标与目标、区域与区域、基元与基元之间。因此,它是图像分割所依赖的重要特征,也是纹理特征的重要信息源和形状特征的基础;而图像的纹理形状特征的提取又常常依赖于图像分割。图像的边缘提取也是图像匹配的基础,因为它是位置的标志,对灰度的变化不敏感,所以可作为匹配的特征点。

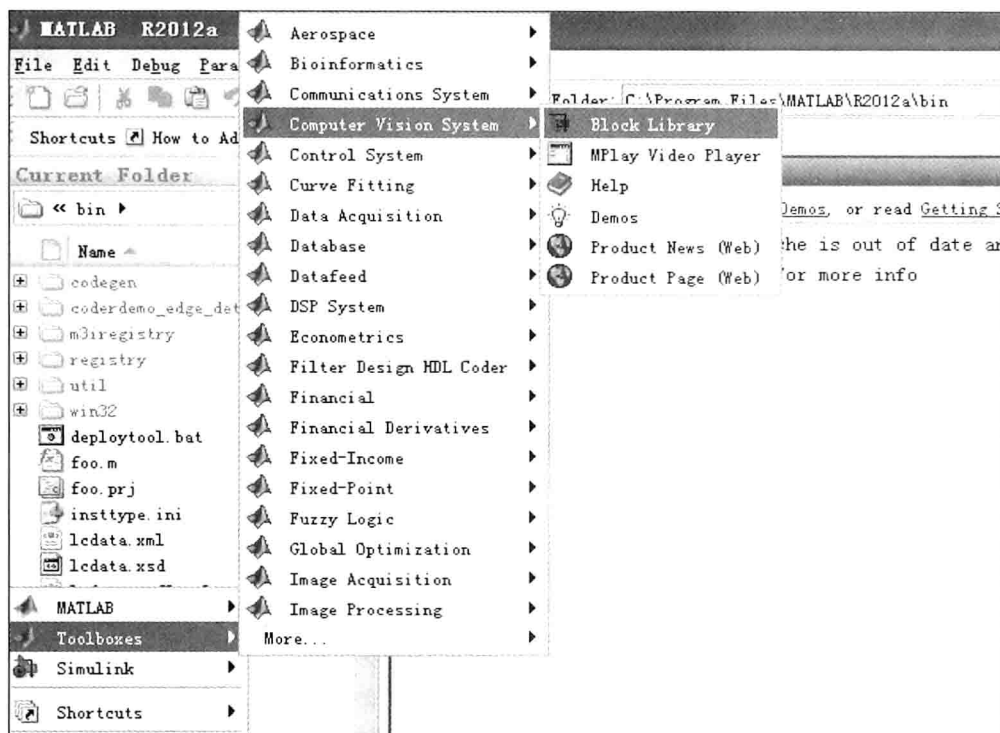


图 1.1.4 计算机视觉系统工具箱的路径

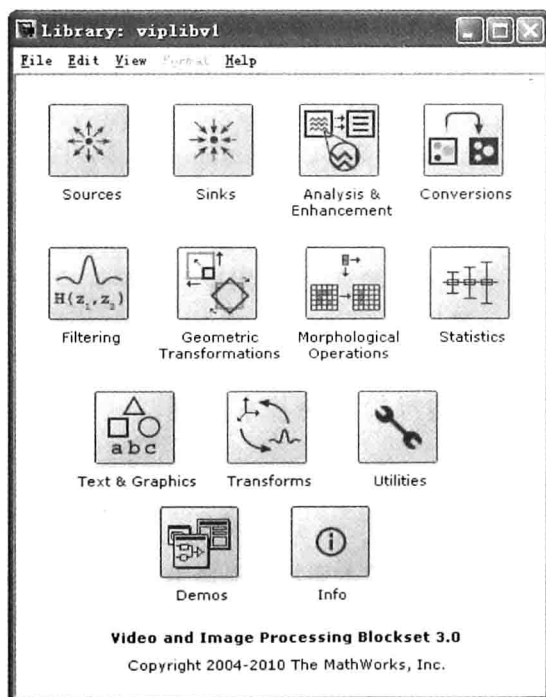


图 1.1.5 视频和图像处理模块

图像的其他特征都是由边缘和区域这些基本特征推导出来的。边缘具有方向和幅度两个特征。沿边缘走向,像素变化比较平缓;而垂直于边缘走向,像素变化比较剧烈,而这种剧烈变化可能呈现阶跃状,也可能呈现斜坡状。边缘上的这种变化可以用微分子检测出来,通常用一阶或二阶导数来检测边缘。不同的是,一阶导数认为最大值对应边缘位置,而二阶导数则以过零点对应边缘位置。

边缘检测的步骤如下:

(1) 图像滤波。边缘检测算法主要是基于图像亮度的一阶和二阶导数,但是导数的计算对噪声很敏感,因此必须使用滤波器来改善与噪声有关的边缘检测器的性能。

(2) 图像增强。增强边缘的基础是确定图像各点邻域强度的变化值。增强算法可以将邻域强度值有显著变化的点突出显示。

(3) 图像检测。在图像中有许多点的梯度幅值比较大,而这些点在特定的应用领域中并不都是边缘,应该用某些方法来确定哪些是边缘点。最简单的边缘检测判据是梯度幅值阈值判据。

(4) 图像定位。如果某一应用场合要求确定边缘位置,则边缘的位置可以在子像素分辨率上来估计,边缘的方位也可以被估计出来。

对于图像的边缘检测来说,一般在识别过程中有如下的要求:

- (1) 首先能够正确地检测出有效的边缘;
- (2) 边缘定位的精度要高;
- (3) 检测的响应最好是单像素的;
- (4) 对于不同尺度的边缘都能有较好的响应并尽量减少漏检;
- (5) 对噪声应该不敏感;
- (6) 检测的灵敏度受边缘方向影响应该小。

边缘检测的实质是采用某种算法来提取出图像中的对象与背景间的交界线。图像灰度的变化情况可以用图像灰度分布的梯度来反映,因此可以利用局部图像微分技术获得边缘检测算子。经典的边缘检测方法是原始图像中像素的某个小邻域来构造边缘检测算子。

Sobel 算子的基本原理是将图像中每个像素的上、下、左、右四邻域的灰度值加权差,与之接近的邻域的权值最大。因此,Sobel 算子定义如下:

$$s_x = \{f(x+1, y-1) + 2f(x+1, y) + f(x+1, y+1)\} \\ - \{f(x-1, y-1) + 2f(x-1, y) + f(x-1, y+1)\} \\ s_y = \{f(x-1, y+1) + 2f(x, y+1) + f(x+1, y+1)\} \\ - \{f(x-1, y-1) + 2f(x, y) + f(x+1, y-1)\}$$

该方法不但能产生较好的检测效果,而且对噪声具有平滑作用,可以提供较为精确的边缘方向信息。但是,在抗噪声好的同时增加了计算量,而且也会检测出伪边缘,定位精度不高。如果检测中对精度的要求不高,则该方法较为常用。

1.1.6 基于模型的实现

下面就通过一个例子来看一下,如何通过 MATLAB Simulink-Blocks 来实现基于模

型驱动的数字图像处理。

- (1) 需求分析。对输入的图像进行 Sobel 边缘检测。
- (2) 实现步骤。RGB 图像输入→转换成灰度图像→边缘提取。
- (3) 建立功能模块,并通过功能模块进行仿真。

其中,各功能模块及其路径如表 1.1.1 所列,连接方式如图 1.1.6 所示。

表 1.1.1 各功能模块及其路径

功 能	名 称	路 径
读入图像	Image From File	Computer Vision System/Block Library/Sources
色彩空间转换	Color Space Conversion	Computer Vision System/Block Library/Conversions
边缘检测	Edge Detection	Computer Vision System/Block Library/ Analysis& Enhancement
观察输出结果	Video Viewer	Computer Vision System/Block Library/Sinks

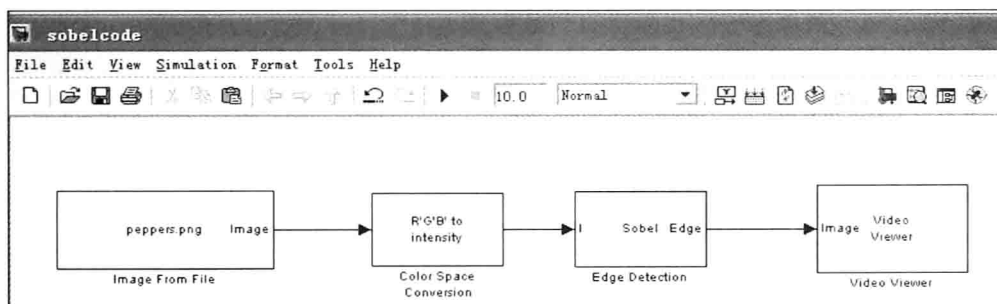


图 1.1.6 基于模型驱动的 Sobel 边缘检测

对各模块的属性进行设置如下:

- (1) 双击 Image From File 模块,将其参数设置为如图 1.1.7 所示的内容。

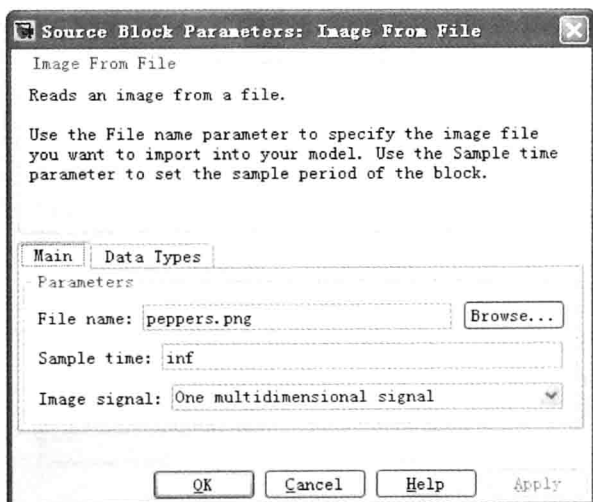


图 1.1.7 Image From File 模块参数设置

(2) 双击 Color Space Conversion 模块,将其参数设置为如图 1.1.8 所示的参数。

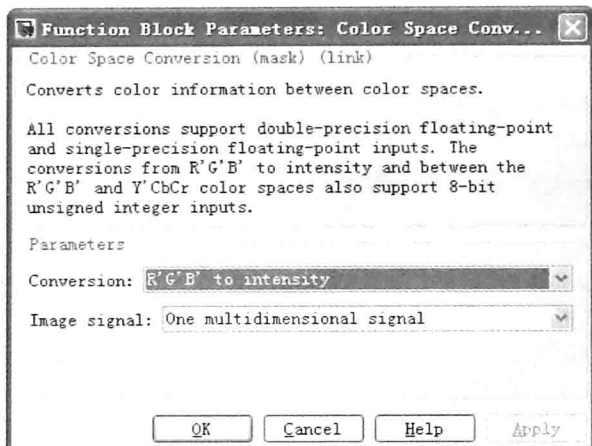


图 1.1.8 Color Space Conversion 模块参数设置

(3) 双击 Edge Detection 模块,将其参数设置为如图 1.1.9 所示的参数。

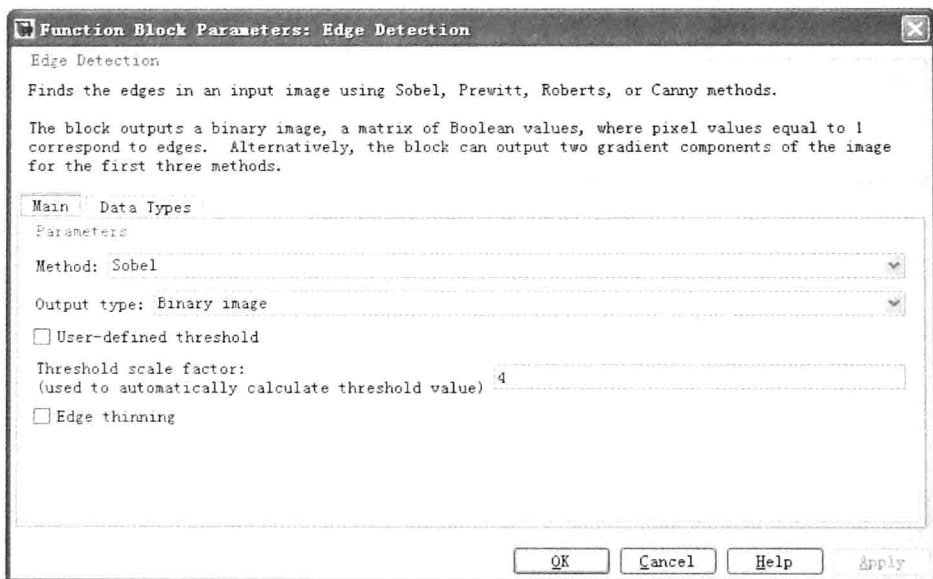


图 1.1.9 Edge Detection 模块参数设置

在完成各功能模块的设置之后,对整个模型进行仿真,证明该模型是正确的并且是可行的,其运行效果如图 1.1.10 所示。

1.1.7 代码的快速生成

接下来,进行代码的自动生成。在 Simulation 下拉菜单下,选择 Configuration Parameters,如图 1.1.11 所示。

将 Solver options 的属性设置为 Fixed-step,如图 1.1.12 所示。

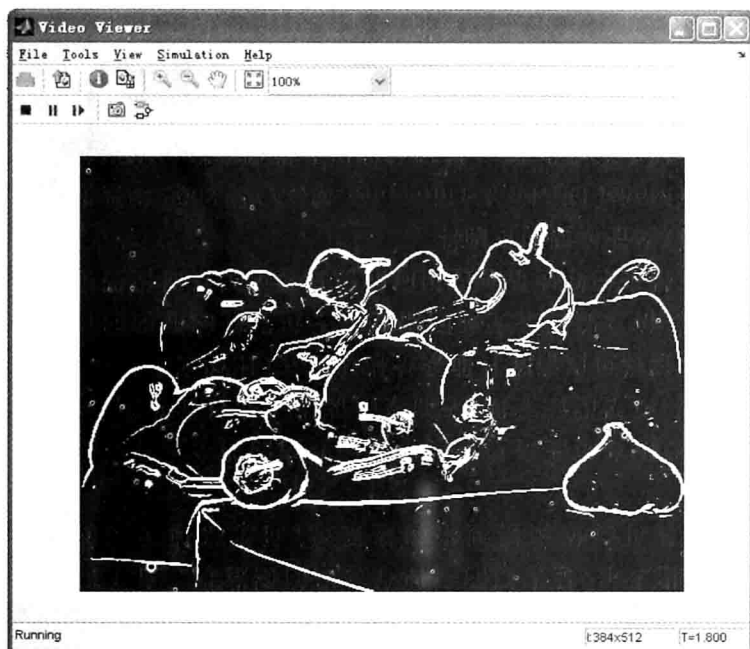


图 1.1.10 模型仿真结果

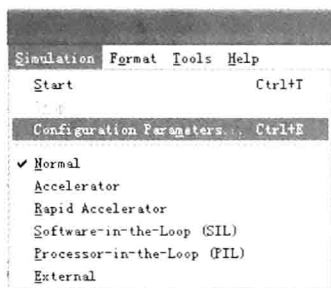


图 1.1.11 选择 Configuration Parameters 操作

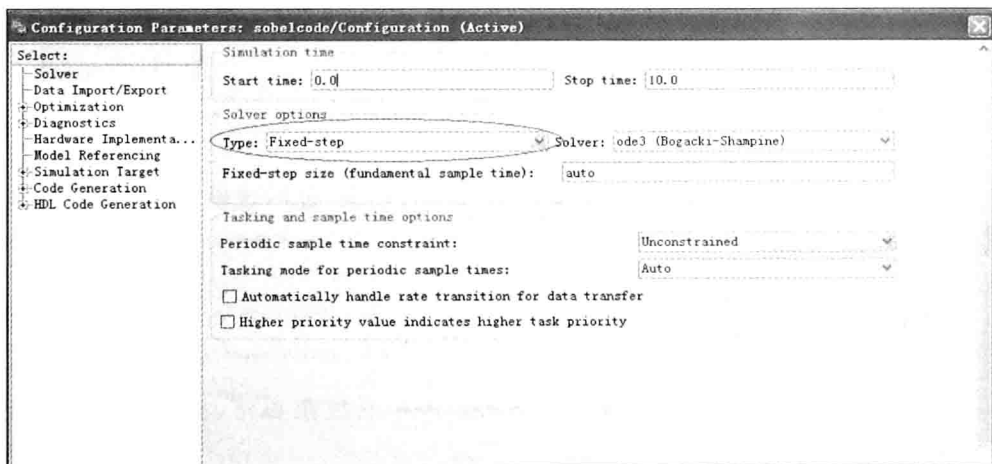


图 1.1.12 将 Solver options 的属性设置为 Fixed-step

单击 Code Generation 下的 Build 按钮(如图 1.1.13 所示),便可生成可执行的 C 代码,如图 1.1.14 所示。

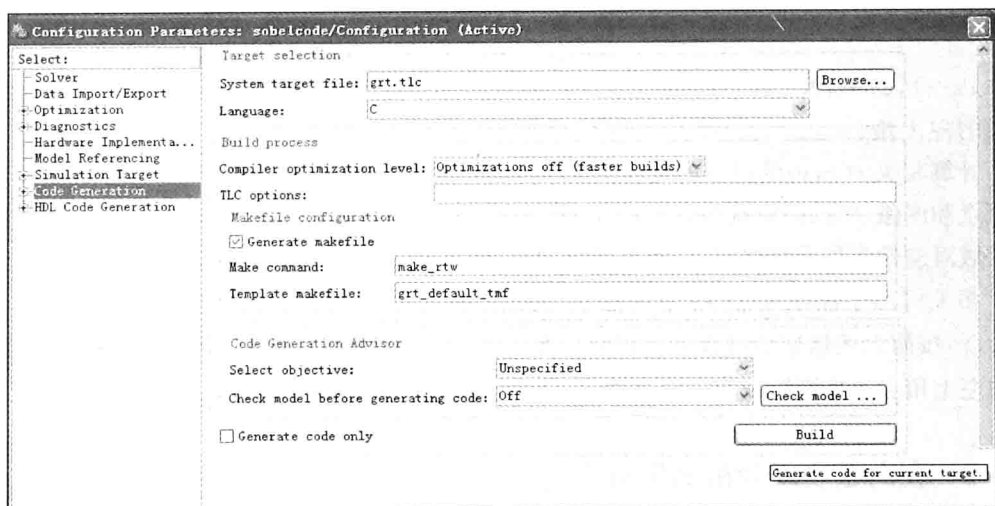


图 1.1.13 单击 Code Generation 下的 Build 按钮

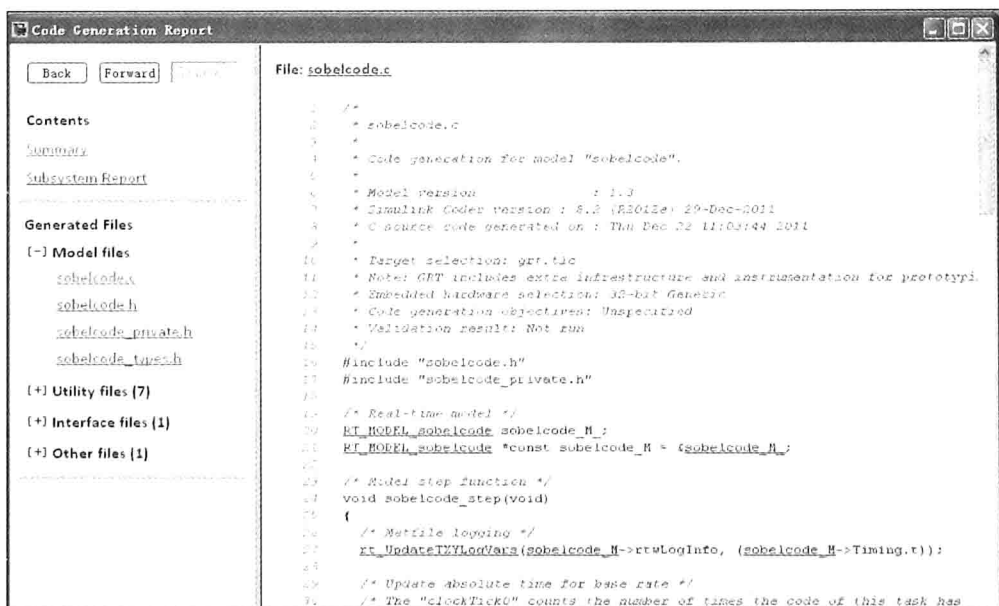


图 1.1.14 生成的可执行的 C 代码

1.2 从向量相关角度实现图像匹配

1.2.1 基于相关运算图像匹配的过程

基于相关运算的图像匹配技术可以直接用于在一幅图像中寻找某个子图,并确定子

图的位置。对于大小为 $M \times N$ 的图像 $f(x, y)$ 和大小为 $J \times K$ 的子图模板 $w(x, y)$, $f(x, y)$ 与 $w(x, y)$ 的相关运算可以表示为:

$$c(x, y) = \sum_{s=1}^{K+1} \sum_{t=1}^{J+1} w(s, t) f(x+s, y+t)$$

其中, $x=1, 2, \dots, N-K+1, y=1, 2, \dots, M-J+1$ 。子图模板 $w(x, y)$ 的原点设置在子图像的左上角。

计算相关 $c(x, y)$ 的过程就是在图像 $f(x, y)$ 中逐点地移动子图像 $w(x, y)$, 使 $w(x, y)$ 的原点和图像 $f(x, y)$ 中点 (x, y) 重合, 然后计算 $w(x, y)$ 与 $f(x, y)$ 中被 $w(x, y)$ 覆盖的图像区域对应像素的乘积之和, 以此计算结果作为相关图像 $c(x, y)$ 在点 (x, y) 处的响应。

相关可用于在图像 $f(x, y)$ 中找到与子图像 $w(x, y)$ 匹配的所有位置。实际上, 当 $w(x, y)$ 按照上述描述的过程移过整幅图像 $f(x, y)$ 之后, 最大的响应点 (x_0, y_0) 即为最佳匹配的左上角点。也可以设定一个阈值 T , 认为响应值大于该阈值点的均是可能匹配的位置。

1.2.2 在向量空间分析图像相关匹配

相关的计算是通过将图像元素和子模式图像元素联系起来获得的, 将相关元素相乘后再累加。完全可以将子图像 $w(x, y)$ 视为一个按行或按列存储的向量 \vec{b} , 将计算过程中被 $w(x, y)$ 覆盖的图像区域视为另一个按照同样方式存储的向量 \vec{a} 。这样一来, 相关运算就成了向量之间的点积运算。

两个向量的点积为:

$$\vec{a} \cdot \vec{b} = |\vec{a}| |\vec{b}| \cos\theta \quad (1.2.1)$$

其中, θ 为向量 \vec{a} 、 \vec{b} 之间的夹角。显然, 当 \vec{a} 和 \vec{b} 具有完全相同的方向(平行)时, $\cos\theta = 1$, 从而式(1.2.1)取得其最大值 $|\vec{a}| |\vec{b}|$, 这就意味着当图像的局部区域类似于子图像模式时, 相关运算产生最大的响应。然而, 式(1.2.1)最终的取值还与 \vec{a} 、 \vec{b} 自身的模有关, 这将导致式(1.2.1)计算的相关响应存在着对 $f(x, y)$ 和 $w(x, y)$ 的灰度幅值比较敏感的缺陷。这样一来, 在 $f(x, y)$ 的高灰度区, 可能其内容与子图像 $w(x, y)$ 的内容并不相近, 但由于 $|\vec{a}|$ 自身较大而产生一个很高的响应。可通过对向量以其模值来归一化解决这个问题, 即通过式(1.2.2)来计算。

$$r = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| |\vec{b}|} \quad (1.2.2)$$

1.2.3 基于向量相关的图像匹配的 MATLAB 实现

【例 1.2.1】 实现基于向量相关的图像匹配。

```
*****
function Icorr = imcorr(I,w)
```