

第 5 章 图像处理基础

机器视觉系统的输入是图像，而输出则是一些符号或者数值，这些符号或数值，有可能表达了物体的特性和位置信息，由图像处理成符号或数值的过程是将图像逐步简化。视觉系统在处理过程中的数据量是非常大的，加上机器视觉系统对处理时间的实时性要求，使得对机器视觉的计算需求是非常苛刻的。

5.1 图像处理分类

关于图像处理，图像分析和计算机视觉的划分并没有一个很统一的标准。一般的来说，图像处理的书籍总会或多或少的介绍一些图像分析和计算机视觉的知识，而计算机视觉的书籍基本上都会包括图像处理和图像分析，只是不会介绍的太详细。其实图像处理，图像分析和计算机视觉都可以纳入到计算机视觉的范畴。

5.1.1 图像处理

底层视觉 (low level vision)，对输入的图像做某种变换，输出仍然是图像，基本不涉及或者很少涉及图像内容的分析。比较典型的有图像变换，图像增强，图像去噪，图像压缩，图像恢复，二值图像处理等等。基于阈值的图像分割也属于图像处理的范畴。一般处理的是单幅图像。

5.1.2 图像分析

中间层视觉 (middle level vision)，对图像的内容进行分析，提取有意义的特征，以便于后续的处理。处理的仍然是单幅图像。

5.1.3 计算机视觉

高层视觉 (high level vision)，对图像分析得到的特征进行分析，提取场景的语义表示，让计算机具有人眼和人脑的能力。这时处理的是多幅图像或者序列图像，当然也包括部分单幅图像。

5.2 编程语言

目前在图像处理中有两种最重要的语言：C/C++和 C#

5.2.1 C/C++语言

C++ 是最重要的编程语言之一，用 C++ 编写的程序的目标代码以它的效率著称，因此通常被用于占用大量 CPU 资源的应用程序。比较适合大型的工程，效率较高，而且容易转成硬件语言，是工业界同时也是目前机器视觉行业最主流开发语言。

5.2.2 C#语言

C#是面向对象的编程语言，是一种安全的、稳定的、简单的、优雅的，由 C 和 C++衍生出来的面向对象的编程语言。它在继承 C 和 C++强大功能的同时去掉了一些它们的复杂特性（例如没有宏以及不允许多重继承）。C#综合了 VB 简单的可视化操作和 C++的高运行效率，以其强大的操作能力、优雅的语法风格、创新的语言特性和便捷的面向组件编程的支持成为.NET 开发的首选语言。

5.3 编程环境

5.3.1 VS (Microsoft Visual Studio)

VS 是 Microsoft Visual Studio 的简称。VS 是美国微软公司的开发工具包系列产品。VS 是一个基本完整的开发工具集，它包括了整个软件生命周期所需要的大部分工具，如 UML 工具、代码管控工具、集成开发环境(IDE)等等。所写的目标代码适用于微软支持的所有平台，包括 Microsoft Windows、Windows Mobile、Windows CE、.NET Framework、.NET Compact Framework 和 Microsoft Silverlight 及 Windows Phone，Visual Studio 是目前最流行的 Windows 平台应用程序的集成开发环境。

5.4 相关知识

5.4.1 .NET-Framework

.NET framework 是微软公司的现代软件开发平台。应用这一框架的关键原因之一是为了大幅度加快软件开发进程。

5.4.2 ActiveX

ActiveX 起初是对于运行在网页浏览器上的应用程序的最佳技术。现在，ActiveX 已经成为用 Visual

5.4.3 DirectX

DirectX 是基于 Windows 的被特定设计用于处理视频数据流的基础架构。如果一个软件组件和 DirectX 兼容，它就可以运行于所有与 DirectX 兼容的硬件上。

5.4.4 WDM

WDM 的全称是 “Windows Driver Model”。具有 “WDM Stream Class” 驱动程序硬件直接与 DirectX 兼容。因此 WDM Stream Class 驱动程序是硬件与软件组件之间的重要连接。

5.5 常用图像处理工具

5.5.1 OpenCV

OpenCV 的全称是：Open Source Computer Vision Library。OpenCV 是一个基于 BSD 许可（开源）发行的跨平台计算机视觉库，可以运行在 Linux、Windows、Android 和 Mac OS 操作系统上。它轻量级而且高效——由一系列 C 函数和少量 C++ 类构成，同时提供了 Python、Ruby、MATLAB 等语言的接口，实现了图像处理和计算机视觉方面的很多通用算法。

OpenCV 用 C++ 语言编写，它的主要接口也是 C++ 语言，但是依然保留了大量的 C 语言接口。该库也有大量的 Python, Java and MATLAB/OCTAVE (版本 2.5) 的接口。这些语言的 API 接口函数可以通过在线文档获得。如今也提供对于 C#, Ch, Ruby 的支持。

OpenCV 拥有包括 500 多个 C 函数的跨平台的中、高层 API，提供的视觉处理算法非常丰富，并且它部分以 C 语言编写，加上其开源的特性，处理得当，不需要添加新的外部支持也可以完整的编译链接生成执行程序，所以很多人用它来做算法的移植，OpenCV 的代码经过适当改写可以正常的运行在 DSP 系统和 ARM 嵌入式系统中，这种移植在大学中经常作为相关专业本科生毕业设计或者研究生课题的选题。

5.5.2 Halcon

HALCON 是德国 MVtec 公司开发的一套完善的标准的机器视觉算法包，拥有应用广泛的机器视觉集成开发环境。这套 image processing library，由一千多个各自独立的函数，以及底层的数据管理核心构成。其中包含了各类滤波，色彩以及几何，数学转换，形态学计算分析，校正，分类辨识，形状搜寻等等基本的几何以及影像计算功能，由于这些功能大多并非针对特定工作设计的，因此只要用得到图像处理的地方，就可以用 HALCON 强大的计算、分析能力来完成工作。应用范围几乎没有限制，涵盖医学，遥感探测，监控，到工业上的各类自动化检测。整个函数库可以用 C，C++，C#，Visual basic 和 Delphi 等多种普通编程语言访问。HALCON 为大量的图像获取设备提供接口，保证了硬件的独立性。它为百余种工业相机和图像采集卡提供接口，包括 GenICam 和 GigE

5.5.3 Labview

由美国国家仪器 (NI) 公司研制开发，类似于 C 和 BASIC 开发环境，但是 LabVIEW 与其他计算机语言的显著区别是：其他计算机语言都是采用基于文本的语言产生代码，而 LabVIEW 使用的是图形化编辑语言 G 编写程序，产生的程序是框图的形式。LabVIEW 软件是 NI 设计平台的核心，也是开发测量或控制系统的理想选择。LabVIEW 开发环境集成了工程师和科学家快速构建各种应用所需的所有工具，旨在帮助工程师和科学家解决问题、提高生产力和不断创新。

Labview 具有模块化的特点，其中 NI Vision Development Module 模块，针对从事开发机器视觉和科学图像应用的科学家、工程师和技术人员设计。

NI 视觉开发模块包括 NI Vision Assistant--供需要不通过编程就实现将 LabVIEW

应用快速成型的直观环境；以及 IMAQ 视觉--拥有强大视觉处理函数的库。与其它视觉产品不同，NI Vision Assistant 和 IMAQ 视觉的紧密协同工作简化了视觉软件的开发。NI Vision Assistant 可自动生成 LabVIEW 程序框图，该程序框图中包含 NI Vision Assistant 建模时一系列操作的相同功能。您可以将程序框图集成到自动化或生产测试应用中，用于运动控制、仪器控制和数据采集等。

- 高级机器视觉、图像处理功能以及显示工具
- 高速模式匹配可以定位大小方向各异的多种对象，甚至在光线不佳时也可实现
- 用于计算 82 个参数的颗粒分析(Blob analysis)，包括对象的面积、周长和位置
- 包括用于 1 维和 2 维代码的可培训 OCR 和读取工具
- 用于纠正透镜变形和相机视角的图像校准功能
- 灰度、彩色和二进制图像处理及分析

NI Vision Builder 与 IMAQ Vision 软件配合工作，能大大简化视觉系统的开发工作。NI Vision Builder 能够自动产生一个在 LabVIEW 环境下运行的 IMAQ Vision 框图，框图包括与您在 NI Vision Builder 中所进行一系列操作相同的功能。

5.5.4 Matlab

MATLAB 是 matrix&laboratory 两个词的组合，意为矩阵工厂（矩阵实验室）。是由美国 mathworks 公司发布的主要面对科学计算、可视化以及交互式程序设计的高科技计算环境。它将数值分析、矩阵计算、科学数据可视化以及非线性动态系统的建模和仿真等诸多强大功能集成在一个易于使用的视窗环境中，为科学研究、工程设计以及必须进行有效数值计算的众多科学领域提供了一种全面的解决方案，并在很大程度上摆脱了传统非交互式程序设计语言（如 C、Fortran）的编辑模式。

MATLAB 包括拥有数百个内部函数的主包和三十几种工具包。其中 computer vision system toolbox (计算机视觉工具箱), Image Processing Toolbox (图象处理工具箱), DSP system toolbox (DSP 处理工具箱) 等工具箱均可适用于图像处理。

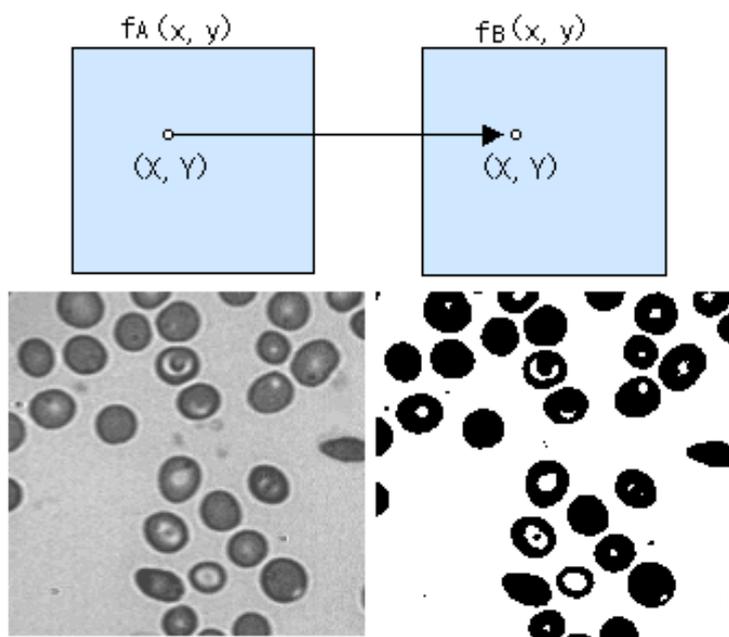
5.6 图像处理相关技术

5.6.1 点运算

有一些运算只对图像的每一个像元点操作, 产生一幅新图像, 二值化就是一个典型的例子。在预先设定了阈值的情况下, 二值化的输出仅仅取决于该点的值, 因此

$$f_B[i, j] = O_{\text{point}}\{f_A[i, j]\},$$

式中 f_A 和 f_B 分别为输入和输出图像。这种运算可使用查找表, LUT, 在图像数据顺序通过时一次完成。各种灰度校正也是这种运算, 所不同的是二值化产生了二值图像, 而校正产生的仍为灰度图像。

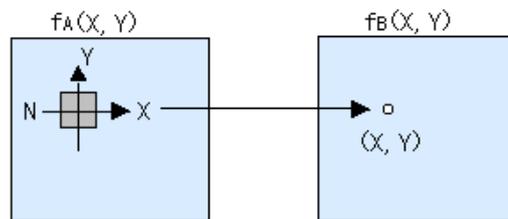
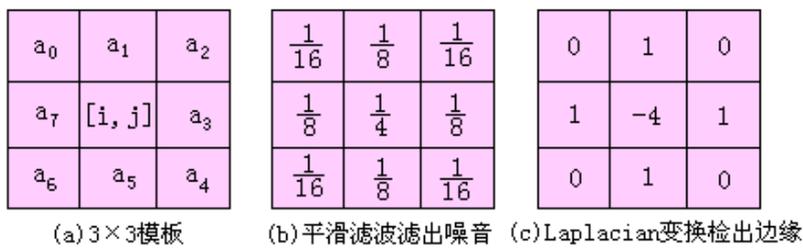


5.6.2 局部运算

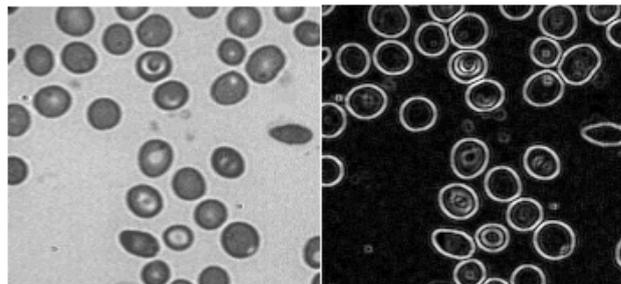
局部运算的输出仍为一幅图像，输出图像的每一个像元值取决于输入图像相对应的像元以及它周围的邻近像元。因此

$$f_B[i, j] = O_{1,ca1} \{f_A[i_k, j_l]; [i_k, j_l] \in N[i, j]\}$$

式中的 N 代表以 $[i, j]$ 为中心的局部子图像。这种运算的一个例子见下图，这种运算几乎会出现在每一个机器视觉系统中，它的输入图像可以是灰度图像，也可以是二值图像；它可以作平滑、锐化、去噪、细化、边缘检出等等运算，局部像元的取法可以有多种方法，十字形、方形、蜂窝形等等，但使用得最多的是方形，例如使用 3×3 或 5×5 或 7×7 的方形模板，模板中每一位置具有一个核系数，用此模板对每一像元及其相邻像元作卷积，是用得最多的一种运算，其它如中值滤波也是去除噪音的较好算法。



(a) 算法示例



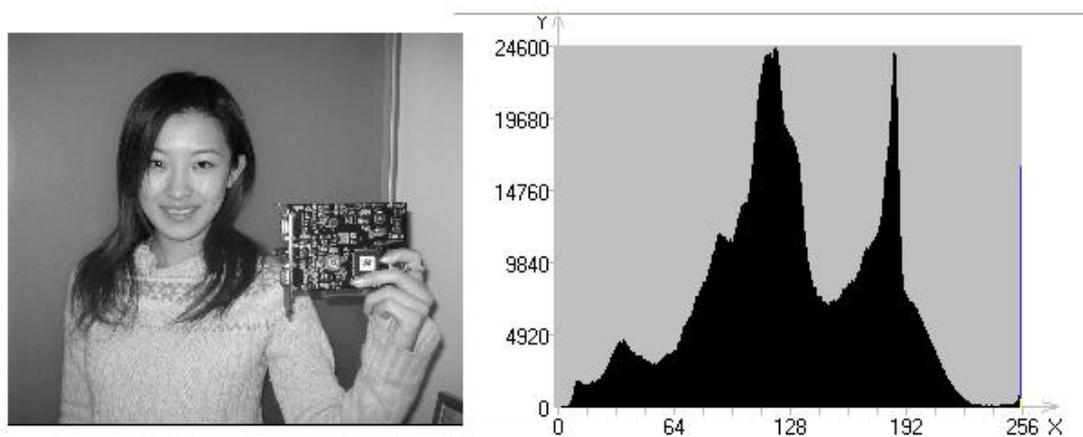
(b) 示例

5.6.3 全局性算法

有一些算法是基于整个输入图像而获得的，被称作全局算子

$$P=O_{\text{global}}\{f[i, j]\}$$

这类算法的一个例子在下图中显示。



它的输出可以是一幅图像，也可以是符号输出。直方图、富氏变换、广义 Hough 变换都是全局算法。

5.6.4 物体级算法

在大多数机器视觉的应用领域，都要求计算出图像中物体的特性。为了识别这些物体，大小、平均灰度、周长、重心、形状和其他一些特性都是常用到的，是直接对目标物体计算而获得的；而在缺陷检测中，也是通过对目标物体所作的检测而获得的。这就引出一个非常困难，而在机器视觉系统中又至关重要的问题：目标物体是什么？在哪里？

在机器视觉中的很多算法都是围绕图像中的物体在那里这一目的而展开的。而图像中的物体还会带来令人左右为难的决策。例如，我们必需使用属于目标物体的所有点来计算该物体的特性；但我们又必需使用这些特性来区别这些点是否属于该物体。

所以从背景像元成功分割出前景物体是视觉系统工作成败的关键。为了彻底理解图像的内容，机器视觉必需对目标物体作多种运算操作，以便做出正确的分割。

下图演示了对一个物体的边缘进行跟踪以后获取的链码，通过对链码可以直接作形状分类；还可以通过链码做富式算子变换，从而突出反映形状的分量。从这个例子可以看出物体级算法不象前面所述的三种类型的算法那么有规律，可以按照像元扫描的顺序来进行，便于使用专门芯片来实时完成。物体目标级算法往往是较复杂的，只适用于处理器来执行，例如 PC 机的 CPU，专用信号处理器 DSP 等；同时这些算法的运算时间不是往往固定的，而是随着图像内容的复杂程度而变化的。

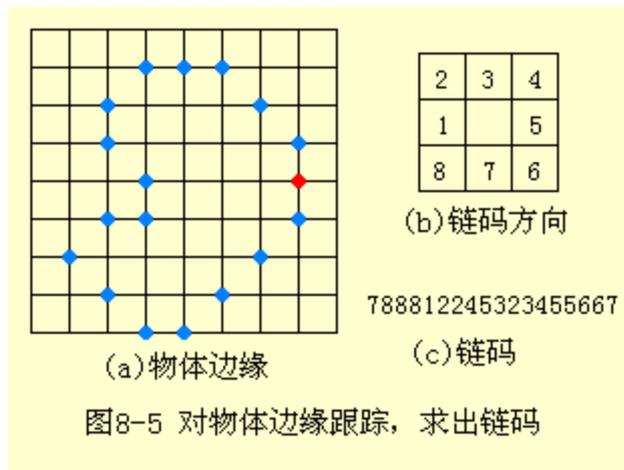


图8-5 对物体边缘跟踪，求出链码