目录

1. 绪论
   1. 课题的研究背景和意义
   2. 基于内容的图像检索技术（谈谈发展历史）
   3. 本文的主要内容和结构安排
2. 服饰图像检索技术

2.1服饰图像检索技术简介

2.2 服饰图像特征描述

2.3 GrabCut算法

2.4 视觉词技术

第三章 基于Android平台的服饰图像搜索系统实现

3.1系统架构

3.2 Android客户端的实现

3.3 服饰图像数据库及特征池的建立

3.3 服饰图像搜索服务器的实现

3.4 总结

第四章 系统测试与实验

4.1 测试环境

4.2 效果展示（颜色检索、纹理、SIFT搜索效果展示）

4.3 搜索性能评估（查准率、检索效率）

4.4 总结

SIFT算法

图像的高斯尺度空间

尺度空间理论的出现是为了模拟图像数据的多尺度特征。Koenderink与Lindeberg证明了高斯卷积核是唯一的线性尺度核。一副图像的尺度空间被定义为：



其中为尺度可变高斯函数：



式中：为尺度坐标；为尺度因子，决定图像的平滑程度，大尺度对应图像概貌特征（低分辨率），小尺度对应图像细节特征（高分辨率）。

SIFT特征

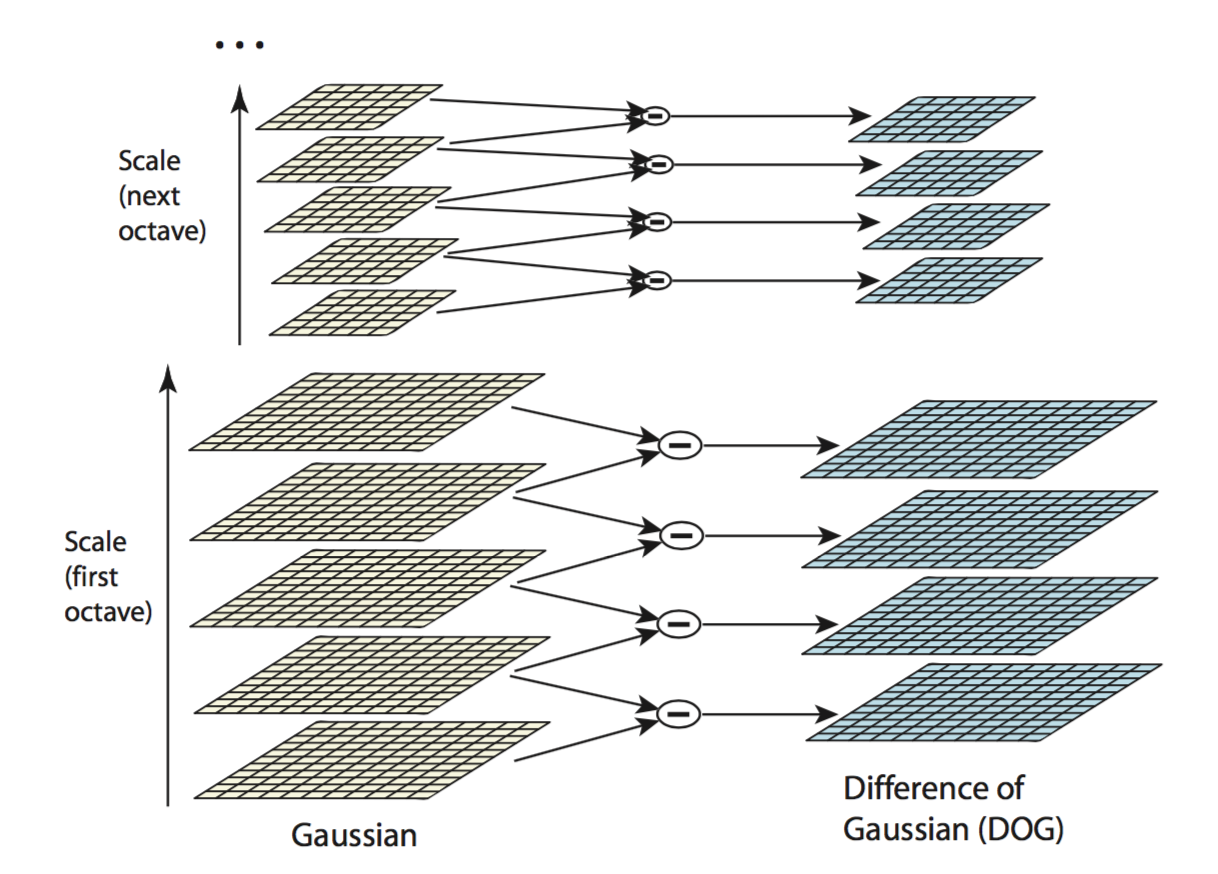
SIFT(Scale-Invariant Feature Tansform)描述子由Lowe于1999年提出，2004年总结完善。SIFT对旋转、尺度缩放、亮度变化具有不变性，对视角变化、仿射变换、噪声的容忍度较高。应用范围包含物体辨识、机器人地图感知与导航、影像缝合、3D模型建立、手势辨识、影像追踪和动作比对。SIFT特征提取主要分为4个步骤：

1. 构建高斯差分尺度空间（DOG scale-space）

高斯差分尺度空间利用不同尺度高斯差分核与图像卷积生成。是两个相邻尺度图像的差。



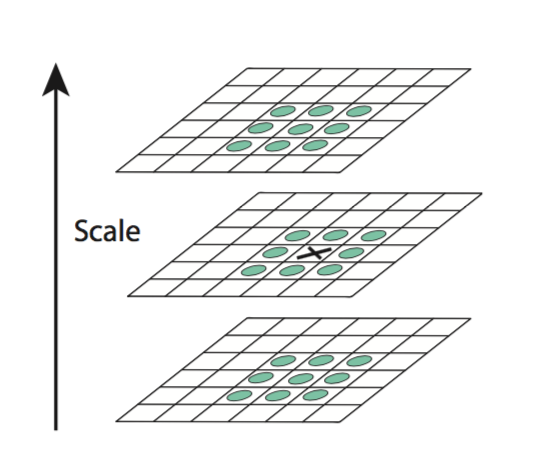
如图x所示。



图ｘ构建高斯差分尺度空间（DOG scale-space）

1. 关键点定位

如果一个像素在DOG尺度空间本层以及上下两层的领域中具有最大值或最小值时，则认为该像素是图像在该尺度下的一个关键点。如图x所示。



图x DoG尺度空间关键点定位

中间的检测点和它同尺度的8个相邻点和上下相邻尺度对应的9×2个相邻点共26个相邻点进行比较，以确保在尺度空间和二维图像空间都检测到极值点。

1. 关键点方向确定

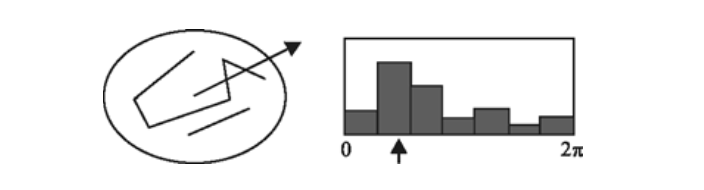
基于关键点邻域像素的梯度分布特性，分配给每个关键点一个或多个方向参数。后续的对图像的操作均相对于关键点的方向、尺度和位置进行变换，从而提供对于这些变换的不变性。



公式x、x分别为处梯度的模值和方向公式。其中所用的尺度为每个关键点各自所在的尺度。

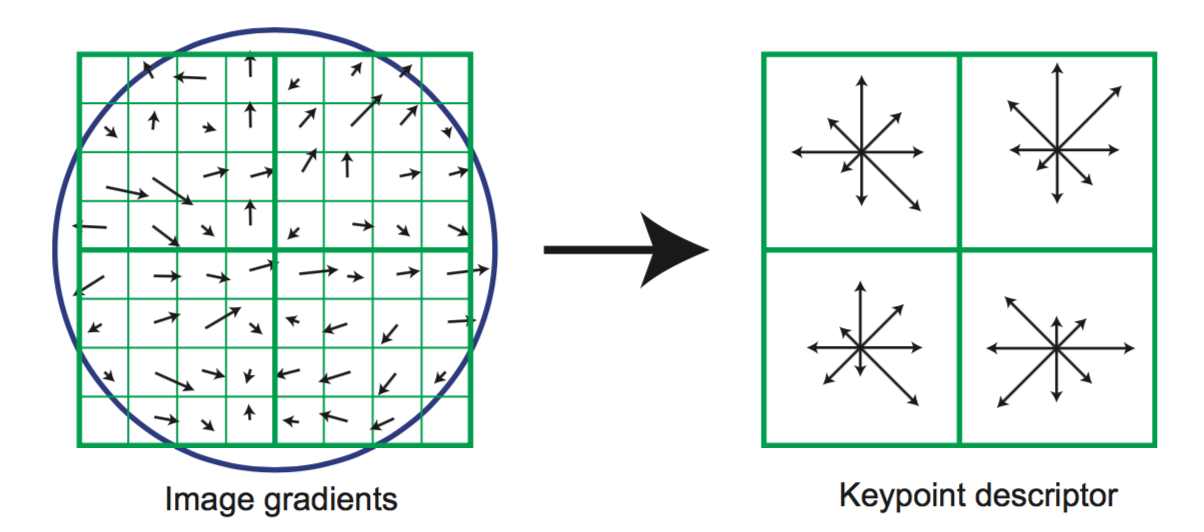
在实际计算时，我们在以关键点为中心的邻域窗口内采样，并用直方图统计邻域像素的梯度方向。梯度直方图的范围是0～360度，其中每45度一个柱，总共8个柱, 或者每10度一个柱，总共36个柱。Lowe在论文中建议使用高斯函数对直方图进行平滑，减少突变的影响。直方图的峰值代表该关键点处邻域梯度的主方向，用作该关键点的方向。图x是采用7个柱来统计邻域像素梯度方向并确定主方向的示例。



图x确定关键点方向

1. 关键点描述子的生成

首先将坐标轴旋转为关键点的方向，以确保旋转不变性。以关键点为中心取8×8的窗口。



图ｘ关键点描述子的生成

图左部分的中央为当前关键点的位置，每个小格代表关键点邻域所在尺度空间的一个像素，利用公式x求得每个像素的梯度幅值，利用公式ｘ求得每个像素的梯度方向，箭头方向代表该像素的梯度方向，箭头长度代表该像素的梯度模值，然后用高斯窗口对其进行加权运算。图中圈代表高斯加权的范围（越靠近关键点的像素梯度方向贡献越大）。然后在每4×4的小块上计算8个方向的梯度方向直方图，绘制每个梯度方向的累加值，即可形成一个种子点，如图右部分示。此图中一个关键点由2×2共4个种子点组成，每个种子点有8个方向信息。这种邻域方向性信息联合的思想增强了算法抗噪声的能力，同时对于含有定位误差的特征匹配也提供了较好的容错性。

计算关键点周围的16×16的窗口中每一个像素的梯度，并使用高斯下降函数降低远离中心的权重。这样就可以对每个关键点形成一个4×4×8=128维的描述子。将这一向量归一化之后，可进一步去除了光照的影响。

第三章 基于Android平台的服饰图像搜索系统实现

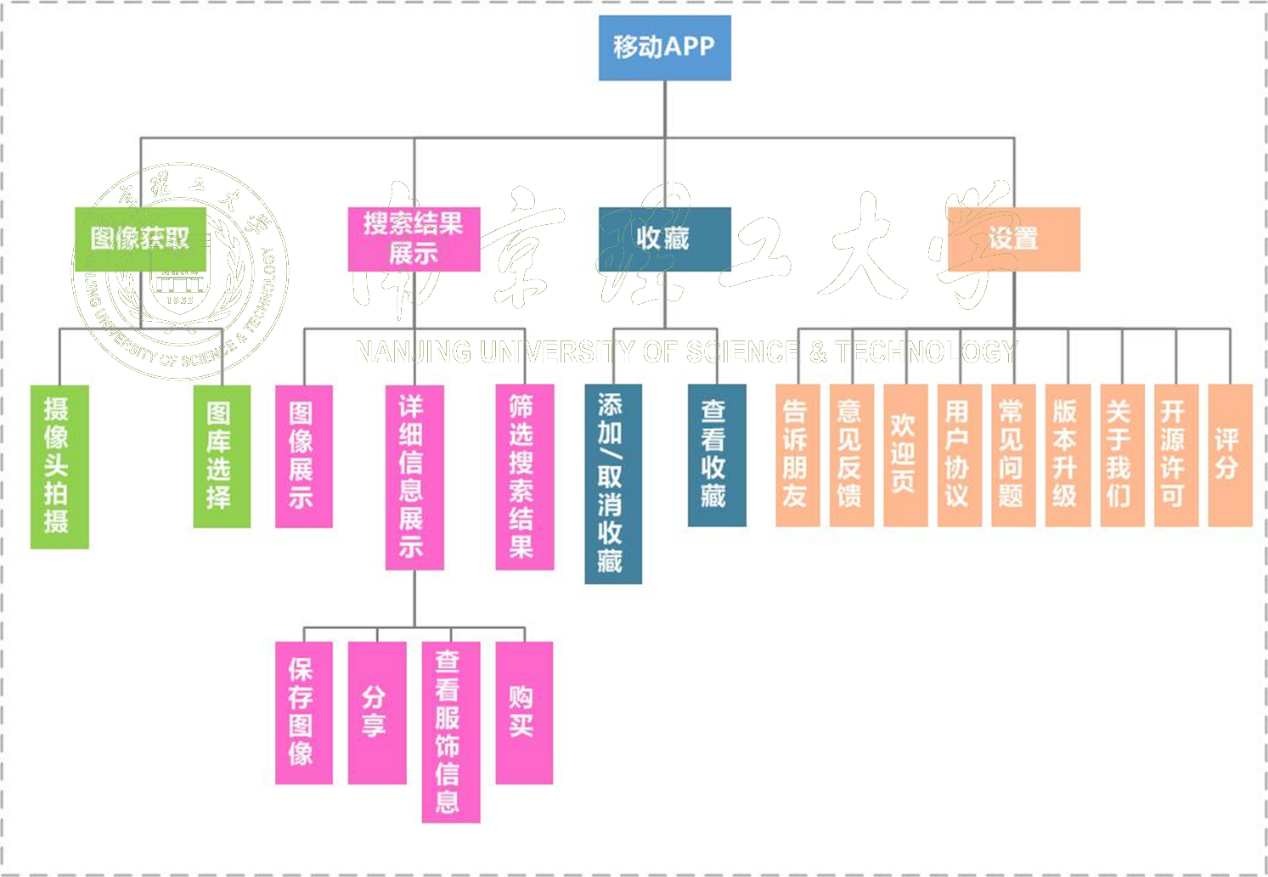
3.1系统架构

本文系统由三个部分组成。一Android客户端。二服饰图像数据库及特征池。三服饰图像的搜索。



3.2 Android客户端实现

Android客户端有样本图像的获取和搜索结果的展示的两个功能。



3.2 服饰图像集及特征池的建立

服饰图像集的建立和服饰图像特征池的建立是离线的。

3.2.1服饰图像集的建立

本文系统采用的服饰图像来自国内主流电商网站，通过网络爬虫程序得到。网络爬虫不仅爬取服饰图像本身，也爬取服饰相关信息，比如品牌、价格、材质、货号。

网络爬虫程序使用Python实现，借助pyquery库。pyquery是类似jquery的Python库。pyquery允许我们在xml文件上做jquery查询，API和jquery非常相似。pyquery利用lxml实现快速的xml和html操作。相比正则表达式，借助pyquery实现网络爬虫更加简洁高效。

网络爬虫程序流程图

3.2.2 服饰图像特征池的建立

服饰图像特征池具体是指存储服饰图像特征的数据库，本文系统使用MySQL数据库。特征提取使用Python实现，借助OpenCV库。OpenCV通过BSD协议开源。OpenCV旨在高效计算，并专注于实时应用。底层基于C/C++，OpenCV可以利用多核进行并行计算。如果启用OpenCL，OpenCV可以借助各式各样的计算平台的硬件加速。

3.3 服务器端实现

服务器端主要功能是特征距离的计算与排序。本文系统服务器采用Python Django架构。Django是基于Python的开源Web框架。它有部署快速、安全性高、伸缩性强三个主要特点。