****

**本科毕业设计**

**（2015届）**

|  |  |
| --- | --- |
| **题 目** | 基于Android系统的图像插值实现 |
| **学 院** | 通信工程学院 |
| **专 业** | 通信工程 |
| **班 级** | 12083415 |
| **学 号** | 12081523 |
| **学生姓名** | 林江 |
| **指导教师** | 陈华华 |
| **完成日期** | 2015年12月 |

摘要  
**ABSTRACT**

目录

1 绪论

1.1 引言

1.2 课题背景及意义

1.3 双线性插值算法的研究现状及发展趋势

1.4 课题研究目标、方法和步骤

1.4.1 研究目标

1.4.2 研究方法和步骤

2 java语言概述

3 android平台概述

4 双线性差值算法概述

5 IDE工具

6 实现

7 总结与展望

7.1 总结

7.2 展望

致谢

参考文献

绪论

1.1 引言

随着社会的发展，智能手机越来越贴近人们的生活，人们对于手机中图像的要求越来越高。而现今手机中，android系统占据了智能手机系统的半壁江山，满足android手机用户对手机图像的需求是必要的。在数字图像处理中，图像插值是图像分辨率处理的重要环节，而对于基于电脑端的图像处理不一定适合于移动终端，因此，基于android系统的图像插值研究变得很有必要。

插值是一种最基本的常用的几个运算，对于数字图像处理是一种基础的数据操作。数字图像插值有很多的应用领域，其中图像缩放和图像旋转是最典型的应用案例，后文主要研究内容在于图像的缩放，并尽可能普及到图像旋转等需要用到图像插值的技术的实现。

本文分析了android系统的发展以及现状，基于android系统的图像插值算法。在基于android系统的图像插值研究中，通过对图像基本元素的定量分析，可以得出影响图像显示的参数，如GRB值，alpha值等，这些参数的获取可以帮助我们定性定量的分析图像，并为图像插值做基本的理论基础。从未为图像插值的实现起到积极的作用

本文的主要工作为：1）研究现有的图像插值算法，主要集中在双线性算法；2）研究在android系统上图像的显示；3）研究android系统上图像的插值实现。

1.2 课题背景及意义

1.3 双线性插值算法的研究现状及发展趋势

1.4 课题研究目标、方法和步骤

1.4.1 研究目标

1.4.2 研究方法

1)文献研究法：根据所研究的课题，通过查阅文献来获得资料，从而全面地、正确地了解掌握所要研究的课题。

2)模拟法（模型方法）：依据双线性算法的特性以及一些参数，对图像进行一定的分析，并模拟出插值之后的可能的结果。

3)定性分析法：通过不同的计算方式对双线性插值的效率，定性分析，得知图像插值后的平滑程度，抗锯齿能力。

1.4.3 研究步骤

1）通过查阅一定的数量的相关文献，对所做的课题有一个宏观上的理解，并理清研究目标、研究方向以及研究的细节步骤。

2）对查阅的资料进行整理、过滤，获取到自己所需要的信息。做好外文翻译，撰写好开题报告、文献综述等。

3）了解Android系统，学习java语言规则，熟悉Android开发工具使用，了解双线性算法原理。

4）分析Bitmap数据，编写双线性算法程序，并应用双线性算法对图像的Bitmap数据插值处理，以实现图像的变换。

5）程序调试阶段，查看程序运行过程，修改程序bug，尽可能优化程序方案，做好程序的封装等工作。

6）将资料汇总，在Android系统上实现双线性算法，撰写毕业论文。

2 java语言概述

2.1 发展简史

Java语言是由美国计算机公司SUN研究而成的语言，它随着互联网的发展而诞生，发展至今，已成为使用率仅次于C语言的一门计算机编程语言。Java的发展最早是在1991年4月份，美国SUN公司开始发展消费性电子产品，例如电子游戏机、电视机机顶盒等，开始使用的语言是C与C++，但由于语言的问题，从而诞生了Java语言的前身--Oak。并经过多年发展，先后为图形用户界面编程增加了事件处理模型，推出针对不同使用人群的标准工具箱，扩展基本类库等更新，java现今早已不仅仅限于电子游戏机等消费性电子产品。

Java是一种简化的C++语言，它在设计之初就将安全性列为第一有限考虑，其具有防止计算机病毒传播的能力，它生动、速度并且交互性好，在互联网发展过程中，它在网页由静态转变为动态的过程中，发挥出重要作用，使网页具有丰富多彩的动画。Sun是要Java成为一个简单、面向对象、分布式的、解释的、健壮的、安全的、结构中立的、可移植的、高效能的、多线程的、动态的程序语言(摘译自 The Java-Language: A White-Pager 1995)。

Java是适应于建立企业级的应用程序的，一般而言，适合于大型程序，则不会适合于应用于Web浏览器和Applet小应用程序，但Java能够被拓展。使用Java作为用户接口，而所有的处理工作都是用GCI码，并且它能够在小应用程序上做所有事情，例如播放电影、产生音频、建立3D模型等。

Java是基于C/C++的编程语言，而不是类似HTML一类的描述性语言，因此Java相比较C和C++容易学，但比较HTML教难。Java并不需要使用者画较多的时间在学习语法上，不需要考虑软件和硬件平台就可以安全的编译执行，如此就突破了用户端计算机环境和CPU结构。并且，由于JVM（java虚拟机）的存在，它将java编译成中间码，从而达到java能在所有的机器上执行，只要那部机器上有Java解释器，因此Java是极其方便移植的。

2.2 基本特点

Java的目标在于适应动态变化的外界环境，现今的Java是安全的、跨平台的、简单高效可移植的，并具备跨平台能力和动态内存管理机制，这些特点为Java现今所达到的高度奠定了技术基础与必然因素。

2.2.1安全

Java设计的初衷，是适用于网络及分布式的环境中，安全必须是其优先考虑的因素。Java语言引入了实时内存分配及布局来杜绝程序员直接修改物理内存布局；拥有多个层级的互锁机制，能有效地减少病毒对计算机程序的破坏；通过JVM虚拟机生成中间码的过程，对字节代码检验，防止病毒的侵入。除此之外，Java语言采用了许多的异常处理类库，对一些程序异常内部处理或者抛出，内部处理如内存不够，应用异常终止；异常抛出如存入数据失败，交给上层处理。

2.2.2跨平台

Java具有良好的跨平台编程能力，而无需考虑编程的平台。一般而言，市面上有许多不同型号的机器，其对应的CPU、内存、主板结构等都不同，而适用于所有机型编程是一件很困难的事而又必要的事情。Java通过JVM（java虚拟机）编辑，实现了只需要在装有JVM的机器上，就可以对java语言进行编译，产生一种结构中立的中间码，这一种中间码不依赖于任何的硬件平台和操作系统并且其编译成机器语言的效率与C语言基本一致。因此，通过中间码的方式，java实现了良好的跨平台开发的能力，并为可移植性提供了保障。

2.2.3动态的内存管理机制

Java语言具有良好的健壮性，在编译和运行程序中采用自动垃圾回收机制对内存进行管理，这使得java比C或C++语言更能适应时刻变化的环境，并且不要求程序员在编写程序时，对处理内存方面要时刻注意。它能够动态的，安全的回收不再使用的内存模块，做到在编程时，无需考虑到内存的管理问题，减少编程的难度。这也使得Java时刻都要对可能出现的问题进行检查，消除错误的产生，防止内容动态变化而导致的内存管理过程中的出错以及内存溢出等问题。

2.2.4多线程功能

多线程要求在同一个应用程序中，多个功能能够同时的执行，从而达到在同时实现多个任务，一般线程用于重复的工作（实现动画、声音播放等），耗时的操作（网络连接、加载本地数据文件等）。Java具有的多线程功能，对于其实现交互能力以及及时响应能力是极大的促进，通过允许多个线程共同存在于同一块内存中，并对CPU分配的时间片资源实现共享，并且可以对线程进行优先级管理，使较为关键的程序优先执行，提高了任务管理的灵活能力，并且使用户感觉到当前任务是被优先执行的并且同时执行的，使应用程序具有实时响应能力。这样的多线程机制有利于实现网络上的交互式操作与响应。

2.2.5简单高效可移植

Java是集成C和C++特点发展而来，同时如果了解C和C++，就可以很快的上手Java程序，同时Java又提供各式各样的方法，减少编程的复杂性，去掉头文件、指针变量、结构、运算符重载等复杂特性，除此之外，Java将数据以及数据工具封装于类中，是完全面向对象的，除了少数的基本数据类型，所有的数据都作为对象为用户服务，并且它所有函数都必须和对象相关，使得用户编程起来容易上手，但执行依旧高效。Java的高效是通过Java解释器实现的，它能够在任何机器上执行Java字节码，然后通过高效能的讲Java字节码转换成机器码，从而在对程序连接时，节省时间，这对开发人员而已，是很有实际效应的。Java的可移植性主要表现为源代码可移植、CPU可移植、操作系统可移植。源代码可移植这一点和C语言一样，即在不同的操作系统上对源码的执行都可以得到相同的效果；而Java的CPU可移植性是将Java作为一个虚拟机而言的，java通过java虚拟机实现可在x86、SPARC和MIPS等CPU上运转同一程序，并且只需要编译一次即可，而且这编译是安全的，不受病毒威胁的中间节码。Java的操作系统可移植性是将Java作为一种图形用户界面和虚拟操作系统而言的，Java采用一套与平台无关的库函数，通过提供虚拟的GUI环境，而使操作系统调用的并不是java源码，而是作为java源码组合的库函数，从而实现了java的从左系统的可移植性。

2.3 Java平台

Java开发平台（JDK）指的是用于构建应用程序的软件包，其中包含了JVM（java虚拟机），Java编译器（Javac），Java归档文件（JAR），java文档等。Javac是javac语言编译编译器，它是一个工具，由java语言编写的接口的定义和类，并将它们编译成中间节码的.class文件；Java归档文件（JAR），jar与平常使用的zip文件差别不大，只是jar文件是讲多个class文件打包到一个文件中去，并且将每个文件分别压缩，在其中添加一个manifest文件，允许开发人员添加指明文件间开始文件等与内容相关的其它星系；java文档，通常称为Javadoc，它从程序源代码中抽取成员、方法、类等注释形成一个和源码配套使用的帮助文档，即它是用来帮助开发人员理解并使用java语言的基本类库的。

从发布以来，java现今主要为三个版本：Java标准版，Java微型版，Java企业版。

标准版一般用于开发桌面应用程序、服务器应用程序以及Applet程序，其中包含构成java语言核心类库，正如名字所表达的一样，它是标准的java平台。

同时java也为一些嵌入式的设备提供了微型版，它是一种高度优化的较小的环境，主要针对娱乐消费类型的电子产品，例如智能手机、汽车导航系统以及各种轻量只能设备。Java微型版是于1999年大会上推出，它允许移动无线设备中间的互相通信，实现共享应用程序。

但这两种对于要求功能强大的大型企业依旧不够，Java企业版采用与传统应用开发截然不同的架构，其中包含许多的类库，可简化、便捷、清晰、规范应用系统的部署与开发，从而更简单的提高安全性以及提高可移植能力。较高的安全性、功能强大、可移植能力外加java与平台无关的特性，使Java企业版能够解决过去企业后端使用的电子信息产品彼此之间无法兼容，解决了企业内部以及外部相互连通的问题。

JDK是与CPU相关的，它与具体的操作系统相关，隔离开发人员对硬件方面的操作。由Java[虚拟机](http://baike.baidu.com/view/1132.htm" \t "http://baike.baidu.com/_blank)和[Java](http://baike.baidu.com/subview/29/12654100.htm" \t "http://baike.baidu.com/_blank)核心类所构成。它为纯Java程序提供了统一的[编程接口](http://baike.baidu.com/view/897136.htm" \t "http://baike.baidu.com/_blank)，而不管下层操作系统是什么。Java成为一个可以在不同操作系统上运行的语言，与Java平台的息息相关。总之，开发PC机上程序使用的是javaSE，开发服务器端的程序使用的是javaEE，移动设备的程序使用的是javaME。

2.4 面向对象编程

面向对象编程是java语言的对开发人员的最大的福利，当然也是最大的难点。面向对象指的是封装、继承与多态。

2.4.1 封装

封装是在面对对象编程的思想上，首先应该想到的方法，简而言之，封装就是将数据或者函数等集合在一个个的方法或者类中。封装在java中的表现为对方法的封装和对类的封装。

假设需要对图片进行操作，其中的可能的操作为放大，缩小，旋转等等，那么放大，缩小等都可以写成一个个的方法，这就是对方法的封装；而一个个的方法即函数都是对图片进行操作，就可以封装在一个类中，这就是对类的封装。在封装的过程中，由于已经实现了对图片的操作，而不需要关心代码中的实现更加不需要去修改其中的代码，从而起到了保护该函数不会受到其它函数的影响。而这一个个的操作对象是图片，因此封装也是面对对象编程的第一步。

2.4.2 继承

继承更多的是为了简化代码，提高代码的清晰度，提高代码的复用，节省开发时间而设立的。继承在java中的表现为对父类的继承并加以修改和扩展。这里介绍一下子类与父类的概念，简单的说就是子类实现父类的所有方法，并且扩展出自己的方法，值得注意的是，子类可以对父类的方法选择是否执行以及对父类的数据是否覆盖。

继承是可传递的。如果C和B的关系为C为父类，B为子类；那么可以有B和A的关系为B为父类，A为子类。也就是A继承于B，B继承于C。

继承在java中的表现为对父类的继承并加以修改和扩展。它是指子类可以继承父类的属性和功能，在实现上它继承了父类的所有的操作，并且可以对父类的方法选择是否执行以及对父类的数据操作。

2.4.3 多态

多态按字面上的意思就是拥有多种状态，具体在代码中表现为同一个方法可以作用于不同的对象，产生不同的结果。在java中的具体实现为方法的重载、类的继承和接口的复用。

对方法的重载是由于java对于同一个方法可以传不同的参数，如果一个方法名传参是一个整型，那么java中允许相同方法名而传参是一个字符串类型（不同于整型）的方法存在，从而表现为同一个方法具有不同的状态，即多态。类的继承是由于相同的父类可以派生出多个父类，那么实现父类的方法，不同的子类自然可以有不同的结果，而继承父类写的方法函数名相同的，从而表现为同一个方法具有不同的状态，即多态。接口的多态性，接口的多态性和类的继承在形式上是一样的，只是类的继承使用的extends关键字，而接口使用的implement实现关键字，接口是可以让不同的类的实现不同的功能，从而表现出同一个方法具有不同的状态，即多态。

Java的多态性，使方法直到系统运行才会依据实际情况决定具体实现的功能，使方法的复用性大幅度提高，便于运行效率的提高，高度灵活的使用场合以及抽象出父类的便利。

// 备用

2.5 开发规范

[Java](http://baike.baidu.com/subview/29/12654100.htm" \t "http://baike.baidu.com/_blank)的类库，包含了编译Java程序所需要的最核心文件。

1.核心库文件，其中有

(1)数据结构的库，包括列表、字典和树等

(2)[XML](http://baike.baidu.com/view/63.htm" \t "http://baike.baidu.com/_blank)分析库

(3)安全方面应用库

(4)国际化和[本地化](http://baike.baidu.com/subview/159886/159886.htm" \t "http://baike.baidu.com/_blank)应用库

2.综合库文件，包含了程序员和其他系统通信的功能文件。

(1)[JDBC](http://baike.baidu.com/view/25611.htm" \t "http://baike.baidu.com/_blank)，即Java数据库联通的[API](http://baike.baidu.com/view/16068.htm" \t "http://baike.baidu.com/_blank)

(2)[JNDI](http://baike.baidu.com/view/209575.htm" \t "http://baike.baidu.com/_blank)，即Java命名和目录接口

(3)[RMI](http://baike.baidu.com/view/99017.htm" \t "http://baike.baidu.com/_blank)和[CORBA](http://baike.baidu.com/view/153815.htm" \t "http://baike.baidu.com/_blank)用于重新分发软件

3.用户界面库文件，包含：

(1)[AWT](http://baike.baidu.com/view/209544.htm" \t "http://baike.baidu.com/_blank)，即抽象窗口开发包，提供了产生[图形用户界面](http://baike.baidu.com/view/185360.htm" \t "http://baike.baidu.com/_blank)所需要的功能

(2)[Swing](http://baike.baidu.com/view/895520.htm" \t "http://baike.baidu.com/_blank)库

(3)其他用于回访媒体文件、录音、截图的库

4.一个用于执行软件的[Java虚拟机](http://baike.baidu.com/view/374952.htm" \t "http://baike.baidu.com/_blank)([JVM](http://baike.baidu.com/subview/160708/160708.htm" \t "http://baike.baidu.com/_blank))

5.插件，可以在浏览器里面使用

6.JWS，可以让终端用户连接到互联网

7.许可文件和文档

3 android平台概述

3.1发展简史（发展，优势）

3.2系统架构（框架，架构）

3.3基本功能（布局，用户界面，图形动画，交互，内容）

3.4开发工具，eclipse sdk jdk

3.5开发规范

4 双线性差值算法概述

5 实现（图片分析，原理）

6 总结与展望

6.1 总结

6.2 展望