****

**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

1，，，，安全

安全的(Secure)：Java是被设计用于网络及分布式的环境中，安全性自必是一个很重要的考虑。Java拥有数个阶层的互锁(Interlocking)保护措施，能有效地防止病毒的侵入和破坏行为的发生。

    Java语言的特点七：安全性   
  
    在类似Internet的这种分布式网络环境中，安全性是个不容忽视的问题。Java语言在安全性方面引入了实时内存分配及布局来防止程序员直接修改物理内存布局；通过字节代码验证器对字节代码的检验，以防止网络病毒及其它非法代码侵入。此外，Java语言还采用了许多面向对象的异常处理机制，负责对一些异常事件进行处理，如内存空间不够，程序异常中止等的处理

2，，，，跨平台

其一，java具有良好的跨平台编程能力，即无需考虑编程的特定平台，它的执行时依赖于JVM（Java虚拟机）编译，即只要在装有JVM的机器上，就可以对java语言进行编译，生成字节码文件的转换，这种特性也为java语言的可移植性提供了保障。

结构中立的(Architecture Neutral)：一般而言，网络是由很多不同机型的机器所 组合而成的，CPU和作业系统体系结构均有所不同;因此，如何使一个应用程序可以 在每一种机器上执行，是一个难题。所幸，Java的编译器产生一种结构中立的目标 文件格式(Object File Format);这使得编译码得以在很多种处理器中执行。

Java语言的特点三：平台无关性   
  
java语言经编译后生成与计算机硬件结构无关的字节代码（Bytecode），这些字节代码被定义为不依赖任何硬件平台和操作系统。当Java程序在运行时，需要由一个解释程序对生成的字节代码解释执行。这体现了Java语言的与平台无关性，使得Java程序可以在任何平台上运行，如MS-DOS, Windows，Unix等，因此具有很强的移植性。

3，，，，动态的内存管理机制

Java语言的特点六:动态的内存管理机制  
Java语言采用了自动垃圾回收机制进行内存的管理。在C＋＋语言中，程序员在编写程序时要仔细地处理内存的使用，例如当某个内存快使用完毕时，要及时释放，以供其它程序使用，一旦内存管理不当，就有可能造成内存空间浪费或程序运行故障。 在Java系统中包括了一个自动垃圾回收程序，它可以自动、安全地回收不再使用的内存块，这样，程序员在编程时就无需担心内存的管理问题，从而使Java程序的编写变得简单，同时也减少了内存管理方面出错的可能性。

其三，它具有良好的健壮性，在编译和运行程序的过程中，都要对可能出现的问题进行检查，消除错误的产生，并提供了内存管理机制进行内存垃圾的自动回收，防止内容动态变化而导致的内存溢出等问题。  
动态的(Dynamic)：Java比C或C++语言更具有动态性，更能适应时刻在变的环境， Java不会因程序库的更新，而必须重新编译程序。

4，，，，多线程

多线程的(Multi threaded)：Java语言具有多线程的功能，这对于交互回应能力及即时执行行为是有帮助的。

其四，它是多线程编程，即允许同一个程序有多个执行进程，允许多个线程共存在一块内存中，共享CPU分配的时间片资源。从而使用户感觉到是同时执行的，使软件具有实时响应能力。

   Java语言的特点五：多线程机制  
  
   Java语言支持多线程机制，多线程机制使得Java程序能够并行处理多项任务。Java程序可以设计成具有多个线程，例如让一个线程负责数据的检索、查寻，另一个线程与用户进行交互，这样，两个线程得以并行执行。多线程机制可以很容易地实现网络上的交互式操作。

5，，，，简单高效可移植

其二，java是面向对象编程，类似于早期的汇编语言是针对机器编程，这对人工编程难度较高，它将数据封装于类，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人工呈现。并且它所有函数都必须和对象相关，必须捆绑在类中。

高效能的(High Performance)：Java位元码迅速地能被转换成机器码(Machine Code)， 从位元码转换到机器码的效能几乎与C与C++没有分别。

可移植的(Portable)：原始资料型式的大小是被指定的，例如"float"一直是表示一个32位元IEEE 754浮点运算数字，因绝大多数的CPU都具有此共同特征。程序库属于 系统的一部份，它定义了一些可移植的程序接口，Java本身具备有很好的可移植性。

解释的(Interpreted)：Java解释器能直接地在任何机器上执行Java位元码(Bytecodes)， 因此在进行程序连结时，时间的节省，这对于缩短程序的开发过程，有极大的帮助。  
 java语言的特点一：简单、高效   
   Java语言与C＋＋类似，如果用户了解C＋＋和面向对象的概念，就可以很快编写出Java程序；此外，Java又不同于诸如C＋＋语言提供的各种各样的方法，它只提供了基本的方法，这样就减少了编程的复杂性，例如去掉了头文件、指针变量、结构、运算符重载、多重继承等复杂特性。Java语言虽然简单，却很高效，它可以用面向对象的方法来描述用户的每一个动作。  
  
  
6，，，，Java语言的特点四：交互式特性

Java是面向对象的网络编程语言，由于它支持TCP／IP协议，使得用户可以通过浏览器访问到Internet上的各种动态对象。并且在网络上用户可以交互式地进行各种动作，而多线程技术的引入使得这种交互式操作更为容易。

Java语言的特点二：面向对象   
Java语言是一种新的面向对象的程序设计语言，它除了几种基本的数据类型外，大都是类似c＋＋中的对象和方法，程序代码大多体现了类机制，以类的形式组织，由类来定义对象的各种行为。Java同样支持类继承，这样也减少了程序设计的复杂性。

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。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，正如名字所表达的一样，它是标准的java平台。而java也为一些嵌入式的消费产品中定制了微型版，它是一种较小的运行环境，例如智能手机以及各种轻量智能设备。但这两种对于要求功能强的大型企业依旧不够，而企业版就能够构建企业级的服务应用，增加了附加类库，以便支持强大的管理目录以及消息处理等功能。JDK则是与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是一个可以在不同操作系统（例如windows,[unix](http://baike.baidu.com/subview/8095/8095.htm" \t "http://baike.baidu.com/_blank)等）上运行的编程语言，java平台是指运行在各种终端（如pc、服务器、移动设备、[嵌入式设备](http://baike.baidu.com/view/2778983.htm" \t "http://baike.baidu.com/_blank)）的系统软件。PC机上的java程序是用javaSE开发出来的，服务器端的java程序是用[javaEE](http://baike.baidu.com/view/169790.htm" \t "http://baike.baidu.com/_blank)开发出来的，移动设备的java程序是用javaME开发出来的。

## 移动平台[编辑](http://baike.baidu.com/javascript:;)

Java 2 Micro Edition 概念

J2ME(Java 2 Micro Edition)是Java 2的一个组成部分，它与[J2SE](http://baike.baidu.com/subview/160728/160728.htm" \t "http://baike.baidu.com/_blank)、[J2EE](http://baike.baidu.com/subview/1507/1507.htm" \t "http://baike.baidu.com/_blank)并称。根据Sun的定义：J2ME是一种高度优化的[Java](http://baike.baidu.com/subview/29/12654100.htm" \t "http://baike.baidu.com/_blank)运行环境，主要针对消费类电子设备的，例如[蜂窝电话](http://baike.baidu.com/view/800226.htm" \t "http://baike.baidu.com/_blank)和可视电话、[数字机顶盒](http://baike.baidu.com/view/1116925.htm" \t "http://baike.baidu.com/_blank)、[汽车导航系统](http://baike.baidu.com/view/122441.htm" \t "http://baike.baidu.com/_blank)等等。J2ME技术在1999年的JavaOne Developer Conference大会上正式推出，它将[Java语言](http://baike.baidu.com/view/229611.htm" \t "http://baike.baidu.com/_blank)的与平台无关的特性移植到小型电子设备上,允许移动无线设备之间共享应用程序。

## 桌面应用平台[编辑](http://baike.baidu.com/javascript:;)

[J2SE](http://baike.baidu.com/subview/160728/160728.htm" \t "http://baike.baidu.com/_blank)商业版本，标准版本 (Java2 Standard Edition) 定位在客户端，主要用于桌面应用软件的编程。

J2SE 包含那些构成[Java语言](http://baike.baidu.com/subview/229611/229611.htm" \t "http://baike.baidu.com/_blank)核心的类。

## 企业级平台[编辑](http://baike.baidu.com/javascript:;)

J2EE Java2平台企业版（Java 2 Platform,Enterprise Edition）[1]

J2EE是一套全然不同于传统应用开发的技术架构，包含许多组件，主要可简化且规范应用系统的开发与部署，进而提高可移植性、安全与再用价值。

[J2EE](http://baike.baidu.com/subview/1507/1507.htm" \t "http://baike.baidu.com/_blank)核心是一组技术规范与指南，其中所包含的各类组件、服务架构及技术层次，均有共通的标准及规格，让各种依循[J2EE架构](http://baike.baidu.com/subview/8293871/8219176.htm" \t "http://baike.baidu.com/_blank)的不同平台之间，存在良好的兼容性，解决过去企业后端使用的信息产品彼此之间无法兼容，导致企业内部或外部难以互通的窘境

[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

2.3 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。

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。

封装在java中的表现为对方法的封装和对类的封装。它的定义是将数据和对数据的操作封装在一起，从具体的实例中抽取共同的性质形成一般的概念。

多态在java中的表现为方法的重载和接口的复用。

**一、封装：**

封装是实现面向对象程序设计的第一步，封装就是将数据或函数等集合在一个个的单元中（我们称之为类）。被封装的对象通常被称为抽象数据类型。

**封装的意义：**封装的意义在于保护或者防止代码（数据）被我们无意中破坏。在面向对象程序设计中数据被看作是一个中心的元素并且和使用它的函数结合的很密切，从而保护它不被其它的函数意外的修改。

封装提供了一个有效的途径来保护数据不被意外的破坏。相比我们将数据（用域来实现）在程序中定义为公用的（public）我们将它们(fields)定义为私有的（privat）在很多方面会更好。私有的数据可以用两种方式来间接的控制。第一种方法，我们使用传统的存、取方法。第二种方法我们用属性（property）。

使用属性不仅可以控制存取数据的合法性，同时也提供了“读写”、“只读”、“只写”灵活的操作方法。

访问修饰符：

Private:只有类本身能存取.  
Protected:类和派生类可以存取.  
Internal:只有同一个项目中的类可以存取.  
Protected Internal:是Protected和Internal的结合.   
Public:完全存取.

**二、继承：**

继承主要实现重用代码，节省开发时间。

１、C#中的继承符合下列规则：

继承是可传递的。如果C从B中派生，B又从A中派生，那么C不仅继承了B中声明的成员，同样也继承了A中的成员。Object类作为所有类的基类。

派生类应当是对基类的扩展。派生类可以添加新的成员，但不能除去已经继承的成员的定义。

构造函数和析构函数不能被继承。除此之外的其它成员，不论对它们定义了怎样的访问方式，都能被继承。基类中成员的访问方式只能决定派生类能否访问它们。

派生类如果定义了与继承而来的成员同名的新成员，就可以覆盖已继承的成员。但这并不因为这派生类删除了这些成员，只是不能再访问这些成员。

类可以定义虚文法、虚属性以及虚索引指示器，它的派生类能够重载这些成员，从而实现类可以展示出多态性。

２、new关键字

果父类中声明了一个没有friend修饰的protected或public方法，子类中也声明了同名的方法。则用new可以隐藏父类中的方法。（不建议使用）

３、base关键字

base 关键字用于从派生类中访问基类的成员：

* 1. 调用基类上已被其他方法重写的方法。
  2. 指定创建派生类实例时应调用的基类构造函数。

**三、多态**

１、多态：同一操作作用于不同的对象，可以有不同的解释，产生不同的执行结果。在运行时，可以通过指向基类的指针，来调用实现派生类中的方法。

编译时的多态性：  
  
编译时的多态性是通过重载来实现的。对于非虚的成员来说，系统在编译时，根据传递的参数、返回的类型等信息决定实现何种操作。   
  
运行时的多态性：  
  
运行时的多态性就是指直到系统运行时，才根据实际情况决定实现何种操作。C#中，运行时的多态性通过虚成员实现。   
  
编译时的多态性为我们提供了运行速度快的特点，而运行时的多态性则带来了高度灵活和抽象的特点。

２、实现多态：

* 1. 接口多态性。
  2. 继承多态性。
  3. 通过抽象类实现的多态性。

３、override关键字：

重写父类中的virtual修饰的方法，实现多态。

2.3 开发规范

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