 单位代码 **10006**

学 号 **39061512**

1分类号 **TP391**

****

毕业设计(论文)

程序代码相似性检测技术

研究与实现

|  |  |
| --- | --- |
| 学院名称 | 计算机学院 |
| 专业名称 | 计算机科学与技术 |
| 学生姓名 | 杨博洋 |
| 指导教师 | 李舟军 |

程序代码相似性检测技术研究与实现 杨博洋 北京航空航天大学

北京航空航天大学

**本科生毕业设计（论文）任务书**

Ⅰ、毕业设计（论文）题目：

程序代码相似性检测技术研究与实现

Ⅱ、毕业设计（论文）使用的原始资料（数据）及设计技术要求：

1、原始材料：BuaaSim相似代码集

2、设计技术要求：设计的应用程序要能够满足代码抄袭监测、代码相似聚类，兼容C和C++语言，并保证效率可以满足实际应用的需求。

Ⅲ、毕业设计（论文）工作内容：

首先针对在线游戏等具有高帧率和高动态性的远程显示应用，移动终端展现延迟高、速率慢等问题，扩展现有Android移动客户端远程交互协议以支持H.264编解码，实现高性能、低带宽的远程交互。

其次，综合考虑移动终端用户远程交互体验、设备资源占用等多因素，设计了一种自适应交互方法，根据检测屏幕像素变化特征，提出屏幕压缩模式的切换算法，从而在传统VNC图像压缩与H.264视频压缩模式之间自适应切换；

最后，在Android 3.2系统实现了MobileMuse系统，并与Android现有远程交互系统VNC进行对比测试，

Ⅳ、主要参考资料：

Yan Lu, Shipeng Li, Huifeng Shen. Virtualized Screen A Third Element for Cloud-Mobile Convergence; Journal of IEEE Computer; Feb.2011

Weiren Yu; Jianxin Li; Chunming Hu; Liang Zhong; Muse: a multimedia streaming enabled remote interactivity system for mobile devices; ACM MUM 11’

计算机学院计算机科学与技术 专业类 390615 班

学生 杨博洋

毕业设计（论文）时间： 2013年2月20日至2013年6月10日

答辩时间： 年 月 日

成 绩：

指导教师：

兼职教师或答疑教师（并指出所负责部分）：

系（教研室） 主任（签字）：

**本人声明**

我声明，本论文及其研究工作是由本人在导师指导下独立完成的，在完成论文时所利用的一切资料均已在参考文献中列出。

作者：杨博洋

签字：

时间：2013年 6 月

**程序代码相似性检测技术研究与实现**

学 生：杨博洋

面向移动终端的自适应远程交互系统研究与实现 宋骐 北京航空航天大学

指导教师：李舟军

摘 要

关键词：移动互联网，移动终端，远程交互，用户体验，资源消耗

**Research and Implement of Detecting Plagiarism in Source Code**

Author :Boyang Yang

Advisor :Zhoujun Li

**Abstract**

**Key words:** mobile network, mobile client, remote interaction, user experience, resource consumption

目 录

[本人声明 I](#_Toc326704749)

[摘 要 I](#_Toc326704750)

[目 录 V](#_Toc326704751)

[1 绪论 1](#_Toc326704752)

[1.1 论文选题的背景及意义 1](#_Toc326704753)

[1.2 国内外研究现状 3](#_Toc326704754)

[1.2.1远程交互协议比较 3](#_Toc326704755)

[1.2.2移动终端远程交互相关研究 6](#_Toc326704756)

[1.2.3商业化远程交互系统 8](#_Toc326704757)

[1.3 研究内容与研究目标 9](#_Toc326704758)

[1.4 课题来源 11](#_Toc326704759)

[1.5 论文组织结构 11](#_Toc326704760)

[2 相关概念和技术 13](#_Toc326704761)

[2.1 RFB协议简介 13](#_Toc326704762)

[2.1.1 RFB协议的消息类型 13](#_Toc326704763)

[2.1.2 RFB协议标准编码格式 14](#_Toc326704764)

[2.2 H264编码标准及开源编码器简介 14](#_Toc326704765)

[2.2.1 H264编码标准简介 14](#_Toc326704766)

[2.2.2 H264开源解码器简介 16](#_Toc326704767)

[2.3 Android系统简介 17](#_Toc326704768)

[2.4 Android 多媒体框架（Stagefright）简介 19](#_Toc326704769)

[2.5本章小结 19](#_Toc326704770)

[3 自适应远程交互系统设计 21](#_Toc326704771)

[3.1 MobileMuse系统设计 22](#_Toc326704772)

[3.1.1 系统需求分析与设计原则 22](#_Toc326704773)

[3.1.2 远程交互框架设计 23](#_Toc326704774)

[3.1.3 H.264编码应用模式设计 25](#_Toc326704775)

[3.1.4 编码器像素填充设计 26](#_Toc326704776)

[3.1.5 消息格式设计 26](#_Toc326704777)

[3.1.6 Android客户端H.264软件解码设计 27](#_Toc326704778)

[3.1.7 Android客户端H.264硬件解码设计 30](#_Toc326704779)

[3.1.8 版本协议设计 31](#_Toc326704780)

[3.2 启发式算法设计 31](#_Toc326704781)

[3.2.1 系统需求分析与设计原则 31](#_Toc326704782)

[3.2.2 系统结构设计 32](#_Toc326704783)

[3.2.3 屏幕更新检测算法设计 33](#_Toc326704784)

[3.2.4 延迟量计算设计 34](#_Toc326704785)

[3.2.5 压缩模式切换设计 36](#_Toc326704786)

[3.3 小结 36](#_Toc326704787)

[4自适应远程交互系统实现 38](#_Toc326704788)

[4.1 Android平台远程交互系统实现 38](#_Toc326704789)

[4.1.1服务器端实现 38](#_Toc326704790)

[4.1.2客户端实现 39](#_Toc326704791)

[4.2 启发式算法实现 42](#_Toc326704792)

[4.2.1 服务器端实现 42](#_Toc326704793)

[4.2.2 客户端实现 44](#_Toc326704794)

[4.3 小结 45](#_Toc326704795)

[5 系统试验设计与分析 46](#_Toc326704796)

[5.1 MobileMuse功能性试验 46](#_Toc326704797)

[5.1.1 实验目的 46](#_Toc326704798)

[5.1.2 实验环境及配置 46](#_Toc326704799)

[5.1.3 试验方法 46](#_Toc326704800)

[5.1.4 测试结果及分析 46](#_Toc326704801)

[5.2 不同应用场景性能对比分析 48](#_Toc326704802)

[5.2.1 实验目的 48](#_Toc326704803)

[5.2.2 实验环境及硬件配置 48](#_Toc326704804)

[5.2.3 实验方法 49](#_Toc326704805)

[5.2.4 测试结果及分析 49](#_Toc326704806)

[5.3 不同应用场景下展现质量分析 55](#_Toc326704807)

[5.3.1 实验目的 55](#_Toc326704808)

[5.3.2 实验环境及硬件配置 55](#_Toc326704809)

[5.3.3 实验方法 55](#_Toc326704810)

[5.3.4 实验结果及分析 56](#_Toc326704811)

[5.4 启发式算法测试 57](#_Toc326704812)

[5.4.1实验目的 57](#_Toc326704813)

[5.4.2 实验环境及硬件配置 57](#_Toc326704814)

[5.4.3 实验方法 57](#_Toc326704815)

[5.4.4 实验结果及分析 58](#_Toc326704816)

[5.5 小结 59](#_Toc326704817)

[结论 61](#_Toc326704818)

[致谢 63](#_Toc326704819)

[参考文献 64](#_Toc326704820)

# 1 绪论

## 1.1 论文选题的背景及意义

程序代码相似性的检测最初是从重复代码检测和代码的优化演变而来。重复代码是指在同一个程序中重复出现的代码块，其最初的出现大多是出于程序需要而频繁调用，但却增加了程序代码的容量和程序运行的时间，这就需要进行代码优化，即对一份源程序文件进行相似代码检测识别。在当今技术水平日益发展的今天，高校大学生的抄袭手段日益先进且日趋普遍，尤其是计算机专业的程序设计课程，抄袭现象更加严重。国外很多教育机构针对程序设计课程的源代码抄袭现象进行的调查显示，高达85.4％的学生承认抄袭过别人的编程作业[7]。相对于自然语言，程序语言语法非常规则，抄袭起来简单的多，完全不用理解程序，通过文本编辑器进行简单的变量替换、添加冗余代码、变换代码顺序等手段就可以改变代码的外观，且不影响程序的正常运行。

与此同时，全球计算机用户数量在不断增长，软件行业的发展达到了一个空前的规模。由于一些软件的功能的共通性(如教务管理管理软件等) [2]，使得在大型软件系统中发生抄袭、雷同现象时有发生，导致软件侵权案件屡有发生，给正当软件公司造成严重影响和危害。如不久前发生的百度搜索和360搜索的纠纷，有网友就通过分析网页源代码后指出360抄袭百度搜索的前端代码，如下图所示：



图表 1 网友贴图举证360搜索只是将代码中的“百度一下”改为“搜索一下”

当人们逐渐形成知识产权保护意识过程时，开始实施相关防伪技术、识别技术甚至使用更先进的智能识别的自动化检测技术达到自主权的保护目的、提高识别效率。同样，计算机教学中也不例外的在不同程度上存在票窃他人作业或作品行为，最常见的现象就是在完成上机编程任务或编程训练考核中，被考核者抄袭他人程序代码后，稍作修改甚至不做任何修改便作为自主设计作品并提交，采用人工目测方法检测很难准确度量这种行为性质以及程度，尤其在计算机程序设计的上机考核成绩的评价中，对于此类问题除了完全抄袭行为外，由于在考核过程中存在人为主观性的影响，使得考核标准带有很大的模糊性和不确定性，在一定程度上限制了考核准确性和效率，导致增加考核真实性难度，降低考核成绩的可信度。由此引出需要解决计算机程序代码以下简称程序代码相似性的检测问题。

### 1.2.1国外研究现状

程序代码的相似度检测的研究在国外起步较早，在20世纪70年代就有大量学者从事代码相似度检测的技术研究。1976年，Purdue大学的Ottenstein[1]首次提出了应用属性计数法(attribute counting)获取相似度的方法，并用Halstead程序度量方法进行程序相似度计算，开发了一个用来检测Fortran程序相似度的系统。该方法的核心思想是提取代码的特征属性，Halstead系统中提取了四个基于长度的软件科学参数，而后Grier和Faidhi分别使用了20和24种参数[3]来检测代码的相似性。这种方法的优点是效率高，但缺点也很明显，就是无法检测出对程序片段的抄袭，并且该方法之后的研究方向走入了死胡同，仅靠增加属性度量是无法提高系统的检测效果的。之后的学者就从属性计数的方法向基于结构度量的相似性检测方法发展。

1990年，Komondoor和Howitz提出的使用程序切片技术进行检测。通过构造出程序依赖图PDG(Program Dependence Graph)，然后使用文本比较方法，找到相似重复的子句。1992年，B.S.Baker通过将源代码中的函数名称、变量、常量、类型等各种标识符转换为参数化特征而提出了基于Parameterized Matching的重复代码检测工具DUP，用于实现文件的比较[2]。

1996年，Verco和Wise指出对于仅仅使用属性计数法的算法，增加向量维数不能改善正确率。改进属性计数法的措施就是加入程序的结构信息，应该结合结构度量(structure metrics)来检测相似度。基于结构度量的相似性检测方法主要是分析程序的结构信息以及执行流程。1998年，D.Baxter等提出了基于抽象语法树(Abstract Syntax Tree)的重复代码检测技术。这种检测技术对C语言程序进行语法解析，建立完备的语法树，随后应用三种算法进行重复代码检测。1998年，Matthias Reiger和Stephane Ducasse提出了一种独立于语言的相似代码检测方法，以及一个用于相似文件检测的工具DUPLOC。

除以上两类算法之外，还有学者提出了新的检测方法。比如基于案例推理的Cogger工具、基于神经网络的Plague Doctor、基于优化编译与反汇编的BuaaSim和基于XML框架的XPDec等等。目前在网上提供代码相似性检测在线接口的工具有卡尔斯鲁厄大学的JPlag、斯坦福大学的MOSS、威奇塔州立大学的Sim和悉尼大学的YAP3等。这些工具普遍采用属性计数和结构分析相结合的方法进行代码相似度计算[1]。

### 1.2.2国内研究现状

国内的相关研究较少，1988年，中国人民警官大学的张文典和任冬伟研制了一个判断Pascal代码抄袭的系统[4]。该系统采用的是属性计数法，主要通过四大类属性：控制类、变量类、运算符类和标准过程的统计来进行相似度计算。

2007年北邮的王继远也研发出自己的一套程序设计课程代码抄袭检测系统，是在现有结构度量法研究的基础上，在分析结构的过程中，将标记字符串描述为XML文档，通过XML文档结构化的树形结构来描述程序。其中XML描述了程序的结构信息，以C代码为例就是宏定义、全局变量、函数定义、函数实现等[5]。在函数实现内，按照变量、控制流程等划分下一层结构，在每一个控制流程内，通过提取初始及其执行部分来划分再下一层，以此类推直到无法继续分解，并提取为纯文本的元素。通过这样规一化的过程，进而通过两个代码规一化后的文本的公共长度以及两代码提交时间的时间戳进行计算。在建立XML结构的过程中可以通过控制粒度来对性能和准确率进行平衡。

北航的相似代码检测工具BuaaSim[7]是我们学校设计的一套先进的系统。通过引入编译优化技术和反汇编技术，从程序语义层面消除类似代码冗余、语句拆分、控制结构等价替换等高级抄袭手段带来的干扰，并给出一个相似度经验阈值和一个基于该经验阈值的简单有效的聚类算法，BuaaSim系统便是基于该相似性检测方法的一个实现，应用在北京航空航天大学高级语言程序设计教学辅助平台[6]。北航的熊皓在2010年发表的硕士论文中[1]，详细阐述了基于静态词法树、基于BP神经网络、基于抽象语法树和基于CIL、抽象语法树和串切片的四种算法，并最终实现了一个C/C++代码相似性检测工具BUAA\_AntiPlagiarism[1]。

## 1.3 研究目标与内容

研究目标：实现一个可对C代码集进行相似代码检测、并定位标注相似模块的工具。

研究内容：

①需求分析：调研代码相似度检测工具功能、性能上的需求。

②算法实现：设计并实现一种或多种相似代码判定算法。

③性能测试：通过现有的通用数据集和实际情景试用，得出准确性等结论。

④实用化：将实现的算法包装入可实用化的软件中。

## 1.4 课题来源

导师指定课题。

## 1.5 论文组织结构

论文组织结构如下：

第一章：介绍了课题的研究背景，对相关研究工作进行了分析和总结，明确了本文的研究目标和主要工作内容。

第二章：介绍并分析了与本项目相关的概念、系统实现过程中需要用到的mingw相关知识、当下主流的代码相似算法、以及各种基于生成的中间代码的相似度度量算法。

第三章：介绍了系统需求分析与设计原则，详细分析了代码相似度算法的原理与设计，以及软件的整体系统设计。

第四章：在第三章设计的基础上，介绍了启发式算法和MobileMuse系统及其中H.264编解码器的实现。

第五章：对系统进行综合性的功能和性能测试，通过对比测试体现出不同系统在资源消耗、视频质量等方面不同的表现，并以此对系统性能进行分析。

最后总结论文，对实现的面向移动终端的远程自适应交互系统MobileMuse进行总结分析。

# 2 相关概念和技术

## 2.5本章小结

# 3 系统设计

## 3.1 MobileMuse系统设计

### 3.1.1 系统需求分析与设计原则

### 3.1.2 远程交互框架设计

### 3.1.3 H.264编码应用模式设计

### 3.1.4 编码器像素填充设计

### 3.1.5 消息格式设计

### 3.1.6 Android客户端H.264软件解码设计

### 3.1.7 Android客户端H.264硬件解码设计

### 3.1.8 版本协议设计

## 3.2 启发式算法设计

## 3.3 小结

# 4系统实现

## 4.3 小结

本章首先介绍了启发式算法的实现，以上一章所描述设计原则为指导，阐述了服务器端和客户端的实现。

随后介绍了移动客户端的远程交互系统的实现，按照服务器、客户端的分类，分别针对实现进行了详细阐述。

本系统的实现遵循了前文提出的设计原则，并达到了目标设计要求，达到了预期效果。实验结果及分析将在第五章进行阐述。

# 5 系统试验设计与分析

本章主要针对移动互联网的场景，设计具体的实验对本文提出的远程交互协议系统进行了性能分析，并和现有Android客户端远程交互系统VNC进行了不同网络带宽条件下的对比，验证了其优势。

## 5.1 MobileMuse功能性试验

### 5.1.1 实验目的

本实验的目的在于验证MobileMuse在无线网络环境中是否可以正常工作，即保证服务器是否可以正确接收客户端的操作并将结果传输到客户端，客户端将输入发送至服务器并显示操作结果。

### 5.1.2 实验环境及配置

### 5.1.3 试验方法

在以上测试环境中，进行客户端连接、文本、浏览器和视频播放等操作，验证系统是否可以正常工作。

### 5.1.4 测试结果及分析

## 5.2 不同应用场景性能对比分析

### 5.2.1 实验目的

### 5.2.2 实验环境及硬件配置

### 5.2.3 实验方法

图 25 MobileMuse与Android VNC对比测试环境

### 5.2.4 测试结果及分析

## 5.3 不同应用场景下展现质量分析

### 5.3.1 实验目的

### 5.3.2 实验环境及硬件配置

### 5.3.3 实验方法

### 5.3.4 实验结果及分析

## 5.4 启发式算法测试

### 5.4.1实验目的

### 5.4.2 实验环境及硬件配置

### 5.4.3 实验方法

### 5.4.4 实验结果及分析

## 5.5 小结

本章对论文所实现的启发式算法和移动终端远程交互系统进行了详细的测试：

第一个实验测试了不同应用场景下移动客户端远程交互系统对运算和带宽资源的占用，实验结果表明，相对于Android VNC，采用H.264压缩传输会降低带宽消耗，在低运动量条件下计算资源消耗与Android VNC相当，只有在高运动量条件下才会产生较高的计算资源消耗。

第二个实验测试了不同应用场景下移动客户端远程交互系统的展现质量，实验表明，在高运动量条件下，H.264模式会显著提升视频质量，同时也能更好的适应低带宽网络环境。

第三个实验测试了PC平台远程交互系统系统中启发式算法的性能，实验表明，在屏幕运动量发生变化时，算法能及时检测到像素的变化数量并进行压缩模式切换，即在高运动量条件下切换到H.264模式，在低运动量条件下切换到VNC模式，取得了用户体验与资源消耗之间的平衡。

# 结论

本文针对当前远程展现协议中一些问题，特别移动客户端的特殊应用场景，提出了自己的远程交互协议方法，通过广泛的调研和辩证的设计，最终实现了一套性能优良的远程展现系统原型，实现了本课题的研究目标。

本论文的主要工作成果包括如下：

1、分析了当前云计算的发展情况和随着移动终端的普及所带来的新的挑战：核心问题在于资源密集型应用与移动终端设备资源有限的矛盾，进而介绍了解决这种矛盾所提出的远程交互解决方法。在总结了国内外相关针对解决上述问题的研究工作的基础上，重点分析了远程交互技术的可行性和已有远程交互技术的发展，指出了面向移动终端的远程交互技术存在的不足和可能的改进方向，为本文的研究和系统实现奠定了理论基础；

2、提出了采用H.264编解码的远程交互方案和综合优化性能和用户体验的启发式算法，分别制定了模块化、稳定性和实时性等设计原则和设计目标。在协议原型系统实现方面，从分析系统的基本功能需求入手，明确系统功能要求，根据系统的设计目标、原则和方法，将整体系统按照模块化结构进行设计实现，具备良好的可扩展性和稳定性；

3、结合具体的实验场景，经过精心的实验设计、测试和实验数据分析，验证了启发式算法的可用性，同时对比其他远程展现系统，验证了该协议下的远程交互系统性能，验证了该系统的在实际应用中的良好可用性和优异性能表现。

论文提出系统的意义在于：

1、实现了采用启发式算法的PC平台远程交互协议，并验证了启发式算法的功能特性，为解决用户体验和资源消耗之间的平衡问题探索了一条可能的解决方案；

2、对比测试表明，H.264方式优化的远程交互系统MobileMuse相比原始Android VNC系统在视频质量和带宽占用率方面占有明显的优势，同时也更能适应低带宽条件下的应用。；

3、该远程交互系统在移动客户端应用上可以最大程度的达到实时性的要求，在设计中也提出了诸如硬件解码等提升性能减少延迟的改进和设计思路，为下一步工作做好了准备。

# 致谢

非常高兴在本科毕业设计的阶段能够继续留在北京航空航天大学计算机新技术研究所学习，自从大二进入实验室以来，在这里完成了国家级SRTP项目和大三暑期的生产实习，接触到了很多很好的老师和同学，感受到了实验室内部浓厚的学习和讨论氛围。

首先衷心的感谢我的指导教师李建欣老师，李老师渊博的理论知识、深厚的学术功底、敏捷的思维方式和对技术方向敏锐的洞察力，都使我受益良多，他对事业锲而不舍的执着追求和忘我的奋斗精神是我学习的楷模，并将影响着我之后的学习生活。感谢李老师对我论文的选题、设计、完成以及系统试验和论文修改等各个方面都给予的悉心指导，同时也感谢李老师在我考研复试过程中也给与了我非常大的帮助。

感谢一下即将到来的研究生阶段的导师怀进鹏老师，是您的渊博学识、创新精神才让实验室有现在这样良好的学术科研环境，让我在其中收获颇多。

感谢ACT实验室的胡春明老师和沃天宇老师，是他们严谨的治学态度、对学生极度负责的责任心，让我在毕设工作中严谨地完成着毕业设计的每一步，他们对学术严格的要求和忘我的工作态度都给我树立了很好的榜样。

要特别感谢实验室的于伟仁师兄，他和我探讨了许多具体的问题，在系统的设计上提出了宝贵的建议，帮我解决了毕设过程中碰到的一些难题，同时也培养了我解决问题的能力，同时师兄严谨的科研精神和认真的工作态度也让我受益匪浅。

感谢一起在实验室参与毕设的08级本科生康健、王聿达、郑京生、李怡、张本龙、马晓航同学，他们在毕设过程也给予了我很多帮助，我将非常珍惜这段一起奋斗的日子。

还要将感谢献给我的家人，是他们给予的默默支持帮助我顺利的完成了毕业设计的工作。

最后感谢所有曾帮助和支持我的老师、同学和朋友们。

# 参考文献