

Dynamic Face Recognition Surveillance Management Platform

Hui Xing, Xin Nin, and Dong Yi

Institute of Automation Chinese Academy of Sciences, Beijing, Beijing, China

Abstract—Video-based face recognition technology is an emerging research direction and topic in the field of computer vision. Face recognition surveillance system is an important application by using this technology. For most of the current face surveillance systems, the running status is unknown and the information inquiry is unavailable. In this paper, a face recognition surveillance management platform is designed and implemented. This platform provides the functions of system's status monitoring, persistent data maintenance, parameters setting and integrates the face surveillance system with the management system. Experimental results show that the surveillance system's data is managed effectively and relevant information is inquired easily by the platform. The proposed platform plays an important role to ensure the stable operation of face surveillance system.

Keywords—Face recognition, Face surveillance system, Management platform, ASP.NET, MySQL

动态人脸识别监控管理平台

邢辉 倪鑫 易东

中国科学院自动化研究所, 北京, 北京, 中国

摘要 基于视频的人脸识别技术是计算机视觉领域一个新兴的研究方向和备受关注的前沿课题, 动态人脸识别监控系统是该技术的一项重大应用。针对大部分监控系统运行状态未知、识别精度不可控制、硬件环境依赖性强、系统周边自然环境复杂多变以及信息无法反查等特点, 设计并实现了一套基于 ASP.NET 技术的动态人脸识别监控管理平台, 该平台提供了监督监控系统状态、维护持久化数据、设置系统参数等功能, 实现了监控系统与管理系统的对接。实验结果表明, 采用该方法能有效的管理监控系统数据和查询相关信息, 并对监控系统的稳定运行起到积极的作用。

关键词 人脸识别, 人脸监控, 管理平台, ASP.NET, MySQL

1. 引言

随着全球信息化程度的逐渐提高和人脸识别技术的快速发展, 将日渐成熟的人脸识别技术应用到实际的生产环境中已是该技术的下一步发展方向^[1]。动态人脸识别监控系统就是人脸识别技术在实际中的典型应用(“动态”即图像可以实时获取, 获取过程无人为干预), 该系统可以在视频中提取出人脸图像, 其集成的人脸识别算法能快速对图像进行人脸识别。监控系统的作用是巨大的, 但它也存在一些限制性因素, 比如人脸识别算法的速度和精度、系统对硬件环境的依赖性较大以及系统运行后运行参数不易改变等。为了控制监控系统的运行、完善监控系统功能、管理大量的人脸特征数据和报警信息, 开发一套功能完善的动态人脸识别监控管理平台成为了一件可行的、有必要的

工作。本文以此为背景, 详细介绍了应用 ASP.NET、Emgu CV 技术和人脸识别技术进行管理平台设计和实现的方法。

2. 系统需求分析

通过前期对本系统潜在使用人员的调研和分析, 管理平台所应提供的功能有人员注册、在库人员分组管理、设备管理、报警信息统计查询、人像比对、模块参数设置等。

2.1 系统数据流分析

数据流程图是描述系统数据流程的工具, 它将数据从系统中独立出来, 以图形的方式, 阐述了数据流的走向和功能流程。对于复杂的管理平台来说, 将系统按照功能进行分解, 能够更加清楚的表达和理解整个系统。管理平台

系统数据流如图 1 所示。

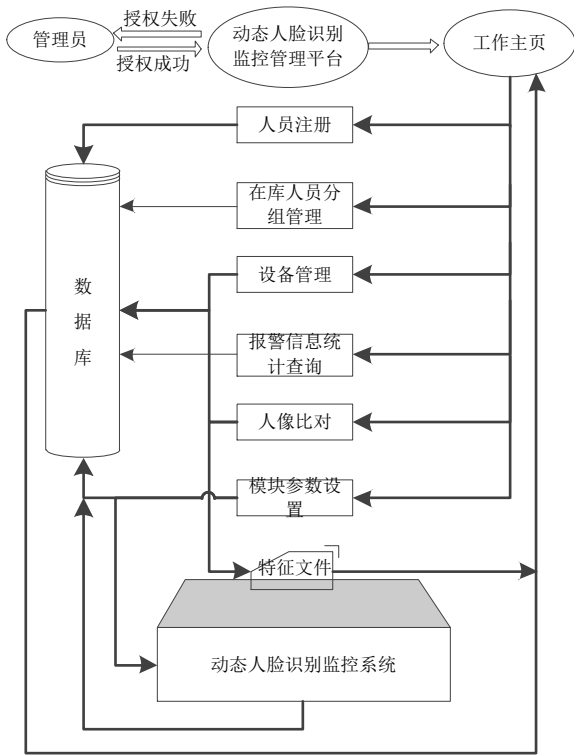


图 1 管理平台数据流图

2.2 系统功能需求分析

管理平台提供多种功能，这些功能共同完成了对动态人脸识别监控系统的状态监督、功能扩展、参数设置以及数据管理。

(1) 人员注册功能主要提供的是对在库人员基本信息进行录入的接口。管理员可以通过管理平台录入人员信息，平台将人员信息保存至数据库并更新人脸特征数据文件。

(2) 在库人员分组管理功能提供的是管理员对在库人员进行分组查看并管理的接口。在库人员分为很多种，比如常驻人口、暂住人口、嫌疑犯等，管理员通过设置组别即可查看属于本组的人员基本信息并进行相应的管理。

(3) 设备管理主要提供了对监控系统所依赖的硬件设备进行统一管理的接口。管理员通过管理平台获取各个设备的状态信息，并对信息进行分析来判断硬件设备是否稳定，若出现问题操作人员可及时采取措施，为监控系统的稳定运行提供了保障。

(4) 报警信息查询提供了对报警信息进行查看的接口。通过此功能，管理员可以指定报警时间段和报警类型等条件灵活的查看报警信息。此功能一方面方便管理员随时查看监控系统的报警记录，另一方面也为办案人员进行报警

信息反查提供了支持。

(5) 人像比对功能提供了静态人脸比对的接口(本文所谓“静态”，即图像的获取过程由人为控制)^[2]。通过此功能，管理员可以上传人脸照片，指定特征库进行人脸比对并显示比对结果。该功能为办案人员进行嫌疑人识别提供支持，办案人员通过上传嫌疑人照片，可很快锁定其基本信息。

(6) 模块参数设置为管理员提供了调节监控系统运行参数的接口。管理员通过管理平台手动设置参数后，管理平台与监控系统进行通信，将这些设置传送给监控系统，从而使这些参数立即生效。

2.3 系统数据库分析

管理平台所维护和管理的数据库是监控系统运行的保障^[3]。监控系统在实际运行过程中，需要两种库的支持，一种是黑名单库，一种是现场人员库。黑名单库是由国家公安部下发的在逃的具有犯罪嫌疑的人员组成，通过将这些人员的信息进行入库并对其照片进行特征提取，组成了黑名单库和黑名单特征文件。而现场照片库是由出现在监控区域内的由摄像机捕捉到的人脸照片及其相关信息组成。这些信息为管理平台的人像比对功能和监控系统提供了数据支持。

3. 系统设计

管理平台基于微软的 ASP.NET 框架^[4]，使用 Socket 技术与动态人脸识别监控系统进行数据通信，使用 Emgu CV 技术进行图像处理，使用人脸识别算法进行人脸特征抽取和人脸比对。

3.1 系统设计思想

随着软件规模和软件复杂性的增加，软件体系结构的选择成为了软件开发必不可少的步骤。基于 B/S 模式的 MVC 三层架构是当下非常流行的软件体系结构框架，这种模式在逻辑上将系统功能分为三层：客户显示层、业务逻辑层和数据操作层^[5]。客户显示层为客户提供应用服务的图形化界面，给用户以最直观的感受。业务逻辑层位于显示层和数据操作层之间，是专门进行软件内部业务逻辑处理的层次，衔接了显示层和数据层。数据操作层是三层模式中的最底层，它用来定义、维护、访问和更新数据并管理和满足应用服务对数据的请求。

基于以上分析可见，管理平台需要图形界面，业务逻辑和数据的支撑，因此基于 B/S 模式的三层框架结构非常适合本管理平台。管理平台逻辑架构如图 2 所示。

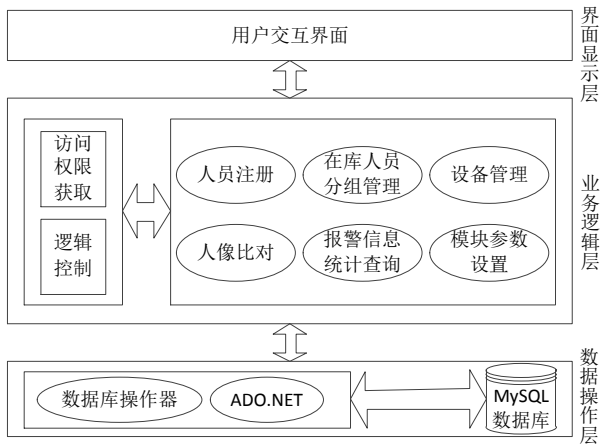


图2 逻辑架构图

3.2 数据库设计

数据库是管理平台的基础和核心，为监控系统和管理平台的运行提供数据支持。管理平台所涉及的数据类型多种多样，根据需求分析，所需要的数据库表如下：在库人员信息表；人员照片表；报警信息表；设备信息表；人员类型表；现场照片表等。限于篇幅，只给出在库人员信息表格式如表1所示。

表1 在库人员信息表

字段	类型	默认值	备注
uid	int	自增	主键
name	varchar(20)		姓名
id_card	varchar(18)		身份号码
gender	tinyint		性别
birthday	datetime		出生日期
height	int		身高
face_type	varchar(20)		脸型
body_type	varchar(20)		体型
fangyan	varchar(20)		方言
pre_name	varchar(20)		曾用名
household_address	varchar(256)		户籍住址
current_address	varchar(256)		现在住址
lib_type	tinyint	允许空	人员所属库
remarks	text		备注

3.3 系统框架结构设计

管理平台的功能由六大模块组成，管理平台的数据源由 MySQL 数据库及其维护的数据表提供，数据源中的数

据一方面来自建库时导入的数据，另一方面来自于通过管理平台新增的数据（例如通过管理平台注册的人员信息）。动态人脸识别监控管理平台的系统架构图如图3所示。

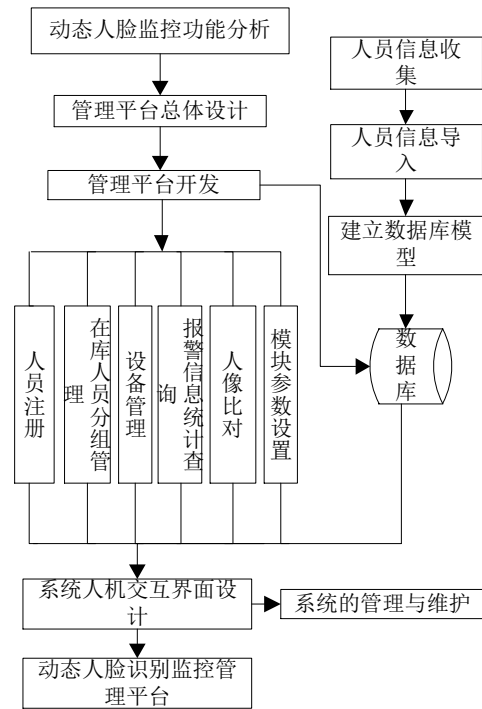


图3 动态人脸识别监控管理平台框架结构图

4. 系统功能实现

本系统基于 ASP.NET 技术，使用了 B/S 模式的 MVC 三层框架结构。管理平台提供的六大模块的功能可以满足对动态人脸识别监控系统的状态监督、对数据的管理以及帮助办案人员进行信息反查的需求。管理平台主界面如图4所示。

4.1 人员信息注册

人员信息注册是更新数据库数据的一条重要途径，管理员手动录入人员信息以及提交人员照片，系统会自动将信息和照片保存到 MySQL 数据库中。然后，通过特征抽取算法提取照片特征，最后将照片特征保存到数据库以及特征文件中^[6]。

4.2 在库人员分组管理

我国地域广阔，人口众多，人员的多样性也使得在对人员信息进行管理时，要尽量做到分类管理。管理人员查询在库人员信息的时候可以选择子库类别，平台会条件查询属于此库的所有人员信息。除此之外，管理人员还能以

人员姓名、性别、住址等作为查询条件，缩小查询范围，加快查询速度。



图4 管理平台主界面

4.3 设备管理

动态人脸识别监控系统是一套模块化的系统，主要分为前端图像采集器、人脸比对模块、报警显示子系统三大模块。前端图像采集器主要由摄像机、MiniPC 和前端采集软件组成。人脸比对模块由比对服务器（一台或者多台）以及人脸比对程序组成。报警显示模块由普通 PC（一台或者多台）和报警显示程序组成。管理平台会监督硬件设备的状态并将信息展现给管理员，由管理员判断硬件设备是否在正常运行。

4.4 报警信息统计查询

报警信息统计查询是监控系统报警子系统模块的功能延伸。管理平台提供报警信息的查询功能，管理员可以指定报警时间段和报警摄像机 IP 以缩小搜索范围，加快查询速度。

4.5 人像比对

人像比对是管理平台提供的跟动态人脸识别监控相辅相成的属于静态人脸识别范畴的功能。管理员通过上传人员照片，系统后台使用 Emgu CV 技术将照片转换成 byte 数组[7]，然后调用人脸特征提取函数提取出照片中人脸图像的特征，最后使用人脸引擎的特征比对函数，将该特征与特征库中的信息进行 1:N 快速比对，比对结果由相似度进行标识[8]，返回前 N 个相似度最高的人脸信息。

人脸识别的核心代码如下：

```
Public ResultSetType FaceRecognition(FeatureType
feature, ValueType count)
{
FaceHandleType handle = 初始化人脸引擎();
ResultSetType result = new ResultSetType ();
If 缓存特征数>0 Then
    Foreach(FeatureType featureBuff in 缓存特征)
    {
        ValueType score = 人脸特征比对函数
(feature, featureBuff);
        result.Add(userId, score);
    }
If result.Size() > count Then
    Return result 前 count 个结果;
End If
End If
Return result;
}
```

人像比对功能模块图如图 5 所示。

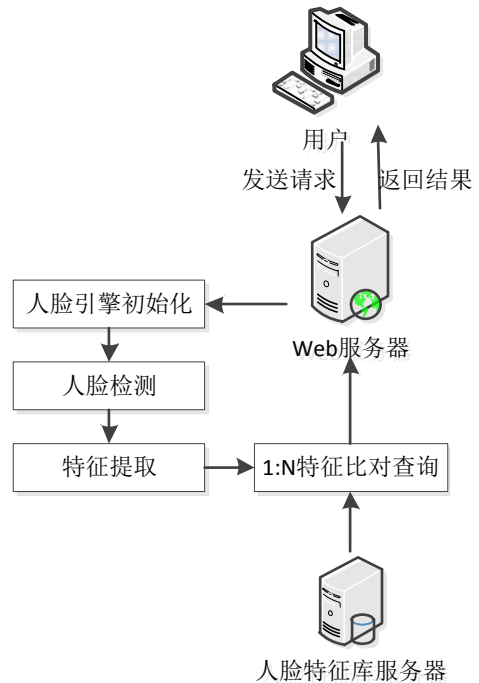


图5 人像比对功能模块图

4.6 模块参数设置

模块参数设置是体现管理平台对监控系统的运行进行控制的功能。管理员录入相关参数，管理平台会跟监控系

统建立 TCP 连接，通过 Socket 将数据发送给监控系统，系统根据参数调整自身状态。

5. 系统安全

系统安全设计要达到的目的是只允许合法用户使用 web service 和应用程序，禁止非授权的用户擅自使用系统。除了网络自身的安全配置(如防火墙)和基础系统(如 server 服务器、操作系统)提供的安全性外，管理平台所使用的安全技术有：对用户进行身份验证，防止 SQL 注入式攻击以及配置 web 服务器的 TCP/IP 筛选等。

5.1 身份验证

使用者通过向管理平台出示自己身份证明以及系统核查使用者合法性两个方面来判断其真实有效身份。系统分别使用 Session 会话对象和 Cookie 在服务器端和客户端保存登录用户的信息，当用户长时间没有新的访问操作时，系统会自动断开与用户浏览器的链接，以保护系统。管理平台身份认证流程如图 6 所示。

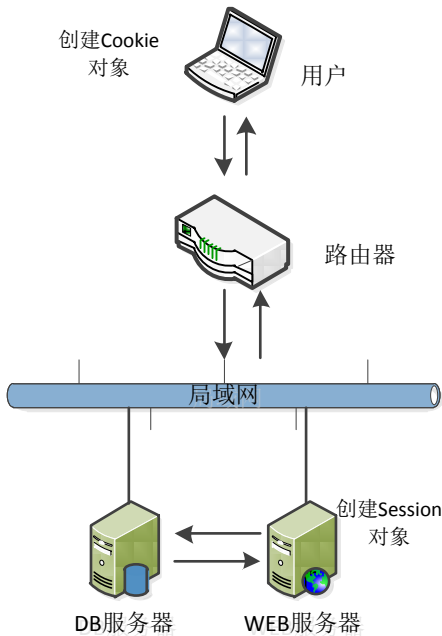


图 6 身份认证流程

5.2 防止 SQL 注入式攻击

管理平台检查用户输入信息的合法性，只有合法的信息才会被作为信息流的输入而被后台处理。数据合法性的检查在客户端和服务端都会被执行，服务器端执行信息验证是为了弥补客户端脆弱的验证机制。因为攻击者可以在客户端获得网页的源代码，然后修改合法性验证的脚本

或者直接删除脚本，使得非法内容可以通过修改后的表单提交给服务器端^[9]。

5.3 配置 web 服务器 IIS 限制连接端口

通过限制 web server 接受的连接种类，有效的为运行管理平台的计算机增加了一道防火墙。在 windows server2008 操作系统中，通过配置 Internet 协议属性的 TCP/IP 筛选,使得 web 服务器只允许在 TCP 端口 80 和 443 上分别进行 HTTP 和 HTTPS 连接。

6. 系统测试

系统测试是系统开发中一个十分重要的阶段，能保证系统质量。如果没有经过充分的测试，当问题在实际生产环境中暴露出来时，会造成很大的损失。本系统主要针对登录、查看在库人员信息以及人像比对三个典型功能进行性能和负载测试。

6.1 系统性能测试

系统性能测试主要是针对系统在正常情况下和并发情况下的请求时间、响应时间、系统状态等进行测试^[10]。测试结果如图 7 所示。

请求	状态	总时间	请求时间	请求字节	响应字节
http://localhost:5414/Login.aspx	200 OK	1.110 秒	1.053 秒	0	74,180
http://localhost:5414/CSS/master/style.css	200 OK	-	0.015 秒	0	8,509
http://localhost:5414/CSS/master/theme1.css	200 OK	-	0.043 秒	0	2,400
http://localhost:5414/Login.aspx	200 OK	-	0.046 秒	0	6,987
http://localhost:5414/WebResource.axd?d=SGI7	200 OK	-	0.048 秒	0	22,346
http://localhost:5414/WebResource.axd?d=P4jN	200 OK	-	0.051 秒	0	26,951
http://localhost:5414/Login.aspx	302 Found	0.111 秒	0.094 秒	350	131
http://localhost:5414/WebForm1.aspx	200 OK	-	0.007 秒	0	17,633
http://localhost:5414/PersonManage/PersonList.aspx	200 OK	0.045 秒	0.012 秒	0	29,432
http://localhost:5414/CSS/master/style.css	200 OK	-	0.007 秒	0	8,509
http://localhost:5414/CSS/master/theme.css	200 OK	-	0.032 秒	0	2,587
http://localhost:5414/PersonManage/DetectFace.aspx	200 OK	0.073 秒	0.023 秒	0	90,929
http://localhost:5414/CSS/master/style.css	200 OK	-	0.011 秒	0	8,509
http://localhost:5414/CSS/master/theme.css	200 OK	-	0.016 秒	0	2,587
http://localhost:5414/WebResource.axd?d=SGI7	200 OK	-	0.018 秒	0	22,346
http://localhost:5414/WebResource.axd?d=P4jN	200 OK	-	0.021 秒	0	26,951
http://localhost:5414/CSS/img_all.css	200 OK	-	0.025 秒	0	710
http://localhost:5414/CSS/img_fixajax.css	200 OK	-	0.047 秒	0	19,190

图 7 性能测试结果图

6.2 负载测试

负载测试是通过在被测系统上不断增加压力，直到达到预设的负载预期的测试方法。这种测试方法可以找到系统处理极限，为系统调优提供数据。

本次负载测试采用最大用户负载 25、本地运行的方式进行测试，测试总数 496 次，失败测试数为 0 次，平均测试时间 29.4S，处理器时间百分比 14.5，测试完成时可用内存容量 850M。测试信息如图 8 所示，测试关系图如图 9 所示。

整体结果

最大用户负载	25
测试秒	0.83
未通过的测试	0
平均测试时间(秒)	29.4
每秒事务数	0
平均事务时间(秒)	0
每秒页数	3.39
平均页面时间(秒)	0.053
请求/秒	16.9
失败的请求数	0
缓存的请求数占总请求数的百分比	0
平均响应时间(秒)	0.020
平均内容长度(字节)	10,589

图 8 负载测试信息

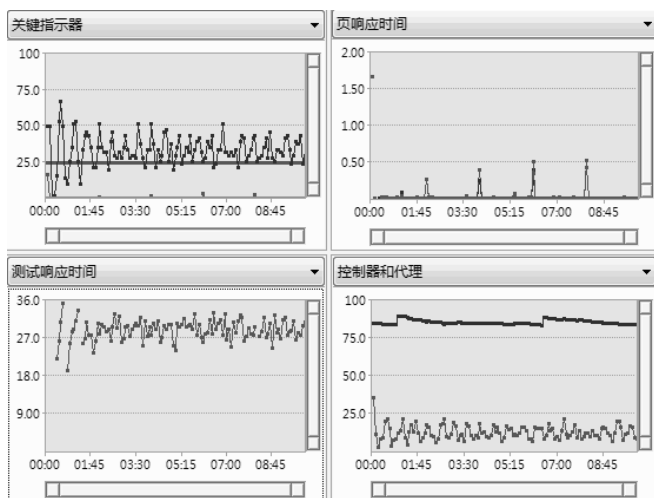


图 9 测试关系图

6.3 测试结果分析

由以上测试数据可知，管理平台在低配测试机服务器上运行稳定，页面响应时间控制在 2.5 秒以内，在设置了模拟思考时间后，连续测试运行 10 次，测试期间可用内存保持在 860M-900M 之间，系统性能基本保持稳定。负载测试中，并发数设为 25 的情况下，实际并发 260 左右，内存占用率较低，系统运行平稳。管理平台在大量用户使用和长时间反复运行中，未出现大的不良反应，CPU、磁盘 I/O、网络传输相对稳定，系统反应良好。在大吞吐量情况下系统响应时间较短，响应速度快，系统稳定性较高。

7. 结语

动态人脸识别监控管理平台是为了保证动态人脸识别

监控系统的稳定运行而进行开发的。在此基础上，进行了功能上的延伸和扩展，包括现场照片查询、报警信息查询、人脸检索等，这些功能为办案人员进行案件信息侦查提供了强有力的工具。管理平台为公共安全提供了有效保障，具有重大的社会意义。管理平台提供 web 访问方式，人机交互方式大众化，因此，管理人员能够利用管理平台准确、快速、合理的对监控系统的运行做出决策，最大限度的发挥了动态人脸识别监控系统的性能。

参考文献(References)

- [1] Bing Yang, Face recognition based multimedia database management system, Nanjing: Radio Engineering, Southeast University, 2005.
- [2] Zhenghai Yuan, Face recognition system and key technology research, Nanjing: Nanjing University of Posts and Telecommunications, 2013.
- [3] Shan Liu, Design and implementation of an improved mysql database access control, Wuhan: Huazhong University of Science & Technology, 2011.
- [4] Yanlan Yang, Xiaoxue Jin, Hua Ye, Asp.net ajax framework and its application in web development, Computer Applications and Software, 2011(28):195-198.
- [5] Weizheng Zhang, Design and implementation of the school network office systems, Computer Applications and Software, 2012(29):243-247.
- [6] Lu Yin, Xiaoguang He, Jie Tian, Design and implementation of multi-purpose face recognition system, Computer Engineering and Applications, 2007(20):225-228.
- [7] Chao Ren, Jianyong Feng, Emgu cv image processing library application in wpf, Computer Applications and Software, 2013(30):233-235.
- [8] Jie Tian, Xin Yang, Theoretical and applied of biometrics, Beijing: Tsinghua University Press, 2009.
- [9] Yong Luo, Shiwei Jiang, Junhua Zhang, Analysis based on asp.net website security vulnerabilities and prevention, Network security technology and Application, 2006(10):37-39.
- [10] Binwu Hui, Mingrui Chen, Dengpan Yang, Web application performance testing and application, Computer engineering and design, 2007(28):2284-2287.