International Journal of Physical Sciences

Volume 11 Number 13 16 July , 2016

ISSN 1992-1950

Academic Journals

ABOUT IJPS

The **International Journal of Physical Sciences (IJPS)** is published weekly (one volume per year) by Academic Journals.

International Journal of Physical Sciences (IJPS) is an open access journal that publishes high-quality solicited and unsolicited articles, in English, in all Physics and chemistry including artificial intelligence, neural processing, nuclear and particle physics, geophysics, physics in medicine and biology, plasma physics, semiconductor science and technology, wireless and optical communications, materials science, energy and fuels, environmental science and technology, combinatorial chemistry, natural products, molecular therapeutics, geochemistry, cement and concrete research, metallurgy, crystallography and computer-aided materials design. All articles published in IJPS are peer-reviewed.

Contact Us

Editorial Office:	ijps@academicjournals.org
Help Desk:	helpdesk@academicjournals.org
Website:	http://www.academicjournals.org/journal/IJPS
Submit manuscript online	http://ms.academicjournals.me/

Editors

Editorial Board

International Journal of Physical Sciences

Table of Contents: Volume 11 Number 13, 16 July, 2016

ARTICLES

academicJournals

Vol. 11(13), pp. 157-167, 16 July, 2016
DOI: 10.5897/IJPS2016.4482
Article Number: 850997559614
ISSN 1992 - 1950
Copyright ©2016
Author(s) retain the copyright of this article
http://www.academicjournals.org/IJPS

International Journal of Physical Sciences

Full Length Research Paper

Development of an identity management system for a web proxy server in a tertiary institution using anonymity technology

Fashoto Stephen Gbenga[1]*, Adekoya Adekunle[2], Owolabi Olumide[3], Ogunleye Opeyemi[2], Adediran Saseyi[2] and Tomori Rasheed[4]

[1]Department of Computer Science, Kampala International University, Kampala, Uganda.
[2]Department of Computer Science, Redeemer's University, Ede, Osun State, Nigeria.
[3]Department of Computer Science, University of Abuja, Abuja, Nigeria.
[4]Department of Computer Science, University of Ilorin, Ilorin, Nigeria.

Received 2 March, 2016; Accepted 4 July, 2016

The inability of a region to access a webpage, because of the ban being placed on users from that region as a result of its location policy, has led to this study. This problem is often solved by anonymizing web traffic by using The Onion Router (TOR). These tools, however, suffer from the problem of exposure of identity and also lack the ability to monitor web users. This study describes in detail a web proxy server service solution within the context of a tertiary institution in Nigeria and explains how this service improves the user experience. An identity management system using a web proxy server was developed to tackle these problems. The new system proxy was designed using a transparent proxy model with some additional translational features where no modification was done to the response or request of resources, other than the addition of its identification information or that of the server from which the message was recovered, and mediation of resources. Redeemer's University proxy was used as a case study in this research work. This system is also able to effectively monitor users' (staffs and students) operations on the web.

Key words: Web proxy, web anonymity, identity management, The Onion Router (TOR).

INTRODUCTION

Anonymizers allow internet users to surf the web anonymously. They also enable internet users maintain a certain amount of privacy which deter gathering of known information like internet protocol (IP) address when browsing (Li et al., 2011). These anonymity services are offered by profit-making organization propelled by subscription fees, non-profit making organizations benefitting through marketing, and home-brewed services through open source anonymous tools. Examples of Community contributed systems are the onion router

*Corresponding author E-mail. gbengafash@yahoo.com.

(TOR) (Dingledine et al., 2004), the Invisible Internet Project (I2P) and the Java Anon Proxy (JAP) (Berthold et al., 2001). Proxy servers are deployed to keep clients anonymous, to obstruct unnecessary content on a network, to save network bandwidth by supplying generally accessed data to clients, and to log and audit client usage (Squid-Cache Wiki, 2014). There are four main types of proxy servers used to gain a degree of anonymity. They are Transparent proxy, Anonymous proxy, Distorting proxy and High Anonymity proxy.

1) Transparent Proxy: It is a form of proxy server that enables the primary IP address to be accessible via the http headers. It is often used due to its capability to cache websites by providing anonymity for its users. Transparent Proxy is known for its transparency because the IP address is visible to everyone. However, it could also be tagged non transparent because its users may not be aware that they are using it.

2) Anonymous proxy: Anonymous proxy associates itself with a proxy server. It does not make its primary IP address visible. Though, this kind of proxy server is detectable, but provides a kind of security to most of its users.

3) Distorting proxy: This is another form of proxy server that allows a wrong IP address obtainable through the http headers.

4) High anonymity proxy: Lastly, this type of proxy server does not in any way relate to proxy server. It does not make its original IP address available. In this study we consider the anonymous proxy.

In the context of TOR, anonymity means preventing the dissemination of a user's internet protocol address. The anonymity set is an assemblage of the sender (source), receivers (destinations) and the servers in a communication network. Anonymity creates two types of users: unidentified users and unlinkable users. Unidentified users are those who post messages without their real name or agent name while unlinkable users are those who post messages without their real name being linked to their account or agent. Anonymity technologies can be used for legal and illegal purposes. Examples of anonymity for legal purposes are privacy, freedom of speech, anti-censorship, and so on, while protection to criminals in facilitating online crimes such as spam, piracy, identity theft, prevention of web filters from monitoring, exposing organization to malicious activities and finally abusing organization resources such as the use of YouTube are examples of anonymity for illegal purposes.

The inability of a region to access a webpage because of the ban being placed on users from that region as a result of its location policy and, also, the effective tracking of users using a network led us to formulate this study. For example, a user trying to access a website which has barred users from certain locations from gaining access and permission to its webpage as a result of its privacy policies would be unable to access the services offered by the site. In addition, the possibility of monitoring the activities of users on a network without a monitoring system can overwhelm and limit the usage and effectiveness of the network.

An identity management system refers to a set of technologies or an information system that is used for project or cross-network identity management. It involves the organization of an individual entity's identity, authorization and authenticating, and also privileges contained in or across system boundaries with the purpose of escalating security and output while diminishing cost, downtime, and monotonous tasks (Wikipedia, 2014). This identity management system would be useful to users from states that are being barred from accessing some web pages in other regions that does not recognize that particular state in context. The introduction of a web proxy in the system for the case of the web will help user agents anonymize their identity. Proxy servers are systems positioned to act as an intermediary for clients seeking resources from other servers or clients to connect to the World Wide Web. This research work intends to design and develop an identity management system using web proxy. From previous research work done in this area, several systems have been developed using diverse models. Over the past years, researches have been conducted in the area of web proxy such as the performance analysis and optimization of web proxy servers and mirror sites by Gautam, et al. (2013) (Improving Performance on WWW using Intelligent Predictive Caching for Web Proxy Servers) which is being integrated into different identity management systems as a method for tracking visits to a web page and anonymization of clients. For instance, when a web request is relayed through the identity system proxy to a server, the server or website sees the proxy as the requesting client. More so, when the server or website tries to identify the name and location of the user from which it received a request, it finds the proxy instead of the client behind the proxy. In 2009, Jelenkovic and Radovanovic reported that web caching improves the web performance by storing web objects close to clients, which reduces the latency of delivering web objects to the end-user (Jelenkovic and Radovanovic, 2009). In addition, it makes the most of the bandwidth of the network which is considered as an important goal for network administrators and it also reduces the load on the origin servers. An additional tool can be used in attaining anonymization of clients in the protection of details about the request originator from the target server which makes such a disguise required in situations, particularly in the case of web browsing since web traffic anonymization is not part of the http specification (Sochor, 2013).

It has been observed that students and some staff of Redeemers University of Nigeria (RUN) have been facing some of these drawbacks in terms of gaining scholarly materials from regions that has not recognized or banned them for obtaining their services and also, the system was used to monitor users (staff and students) operations effectively on the web. This study developed identity management system using web proxy server to tackle these problems.

This study was based on utilizing some set of data to assist users in gaining access and also network administrators in tracking the operations of users on the network. The aim was to develop a robust system that satisfies both students and staff of RUN in all aspects of gaining scholarly materials from all parts of the Web with the effective tracking of user operations on the web by the use of a proxy.

The next section introduces us to the review of relevant works, the concept of web proxy and identity management system, while subsequent sections present the methodology, implementation, conclusion, and references, in that order.

REVIEW OF RELATED WORKS

The emergence of the connected world has brought about a wave of trends in human day to day interactions. In recent years a great deal of attention is being placed on security in the information society to the problems being faced in the management and protection of vital details and identity. Several identity management systems have been developed by various individuals and communities to suit their needs for the protection of critical details. The information gotten from these works has been used effectively in the creation and improvement of diverse identity management systems in different fields.

The growing popularity of the Web necessitated the roles of proxies on the internet. Web proxies emerged as one of the most common intermediaries in the transmission of web messages between user agents and origin servers as shown in Figure 1. Proxies began to play a vital role as early as 1994 (Balachander and Rexford, 2001).

The original purpose of a web proxy was to provide access to the web for clients who are behind an organizational firewall by ensuring the clients did not lose any functionality when requests and responses are being routed (Luotonen and Altis, 1994). Thus, the first stated design criterion for a web proxy was to ensure that clients do not lose any functionality by having their requests and responses routed through a firewall. The original web proxy was actually a version of a gateway server at the CERN laboratory, where the world’s first web server was created (Luotonen and Altis, 1994). The CERN http server (the first web proxy) arranged caches hierarchically (Mahanti et al., 2000). A web proxy’s role in information hiding and encapsulation is described in Shapiro (1986). Web proxies were also originally designed to allow network administrators control internet access from within the intranet (Baentsch et al., 1997). On the other hand, resulting inquiries about proxies show that they serves as intermediaries for commonly requested documents. Thus, web proxies, which started as a gateway server, have now become a vital part of web user’s experience on the Web.

Web anonymity

A web proxy can be defined as an application program that accepts the retrieval of documents from some clients, and relay these requests to the suitable servers (if need be), and then send the requested documents back to the clients (Fielding et al., 1998).

Web proxy is relatively efficient and fast. It aids the anonymization of clients behind it, monitors and helps the administrator to effectively track its users and performs many other functions.

Squid-Cache Wiki website in 2013 defined a proxy as a popular and useful intermediary on the web that enables a large number of clients behind it to share access. Proxy servers are systems deployed to act as a mediator for customers looking for assets from different servers or clients to connect to the World Wide Web. Proxy servers are deployed to keep clients anonymous, to block unwanted content on a network, to save network bandwidth by supplying commonly accessed data to clients, and to log and audit client usage. A proxy basically goes about as a system 'go between'. As opposed to asking for a web object straightforwardly from a remote host, the client may ask for it from an intermediary, which thus acquires the article itself and advances the response to the requesting client (Piatek, 2004).

Identity management

Identity has become a new focal point in today’s global world; likewise, Identity management has become a significant factor in this era due to its potential value as it aids the proper handling of sensitive data (Gaurav and Pruthi, 2012). The management of different identities is a key element of information system safety. It involves the protection of various individuals’ digital identities. Digital identity alludes to ascribed qualities credited to a person, which are instantly available by specialized means (Müller and Böhm, 2011). Alternatively, digital identity refers to a situated set of qualities and properties around an individual that are related together and accessible in an electronic structure to build trusted digital certifications (Al-Khouri, 2012). The identity of an individual consists of

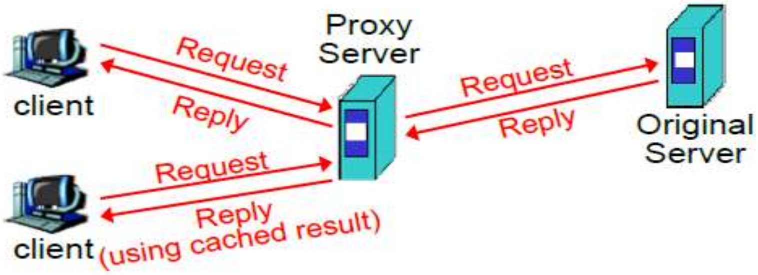


Figure 1. A Web Proxy server (NCTU-Education, 2007).

a vast number of private information with respect to the person. Identities of individuals are exposed in different contexts; that is why an identity management system is used in managing various partial identities.

Identity management systems furnish the administrator with the apparatuses and tools expected to change a user's part, monitor client exercises and to implement policies on client premises. These frameworks are intended to give a method for regulating client accesses over a whole venture and to guarantee consistence with approaches and government regulations. The objective of identity management is to offer extensive entities with the capacity to utilize a concentrated database of client characters to streamline work process and computerize undertakings managing client confirmation, access rights, arrangement authorization, and provisioning of both physical and electronic assets (SANS-Institute, 2003).

Web privacy has been a noteworthy topic amongst the World Wide Web community. Anonymity is being applied in diverse ways across the web due to the privacy concerns raised. Ruiz-Martinez, in his work “A survey on solutions and main free tools for privacy enhancing Web communications”, emphasized the need for users to perform anonymous communications when surfing the internet in other to protect their ever increasing digital identity. This significance is also mutual to both the users and the research community (Gross and Rosson, 2007). Gross and Rosson expatiated on different privacy enhancing tools, among which is the web proxy. An anonymous Web proxy(also known as anonymizer) behaves as a TCP proxy and gets rid of headers with client’s information (or fake them), rephrases HTML pages so that when the client connects to a link on that webpage, the request is requested through the proxy and hides the client’s identity on the network (Shubina and Smith, 2003).

In general, these systems also manage cookies for the client but, at the end, the demand originates from the same IP address and as a result, the IP address of the client can be known. Furthermore, if the intermediary is a third party, then, the address of the client on the network cannot be known (Edman and Yener, 2009; Li et al., 2011). The advantages are the towering effectiveness they proffer, simple to contact and to use and ease. Web proxies do notrequire extra components (Edman and Yener, 2009). The principal shortcoming of using web proxies is that a straightforward anonymizerdoes not guard against traffic examination even though aSSL/TLS link is being used as researched by several researchers in the web community.

Sochor (2013) reported in originator from the intended server. In his study, the focus was on the anonymization tool which is known as TOR (The Onion Routing). TOR can be said to be an extra tool since web traffic anonymization is not a component of the http specification, to create a disguise especially in the case of web browsing. Noteworthy deceleration of anonymized traffic contrasted with ordinary activity is inescapable yet it can be controlled now and again as this study proposes. The outcomes exhibited in the study concentrates on measuring the parameters of such deceleration regarding response time, transmission speed and latency and proposing routines on the most proficient method to control it. The study concentrates on TOR mainly in light of the fact that recent studies by (Liska et al., 2010) and (Sochor, 2012) have reasoned that different tools (like I2P and JAP) give worse services. Sets of 14 record areas and 30 web pages have been shaped and the dormancy, reaction time and transmission rate amid the page or document download were measured over and over both with TOR dynamic in different designs and without TOR. The major result offered includes various ways on the most proficient method to enhance the TOR anonymization proficiency and the proposition for its programmed control. Disregarding the way that productivity still remains too low when compared to ordinary web activity for standard use, its programmed control could make TOR a functional instrument in uncommon cases. The study also conducted a deeper analysis of TOR behavior, especially with respect to the possibility of improving the TOR behavior and efficiency. Finally, the study reported that in order to achieve and sustain the best results for TOR, it must ordinarily be used in the environment of permanently changing www communication. Automatic control using a fuzzy controller or an artificial neural network was proposed with the expectation that this will allow for the automatic tuning of anonymization parameters.

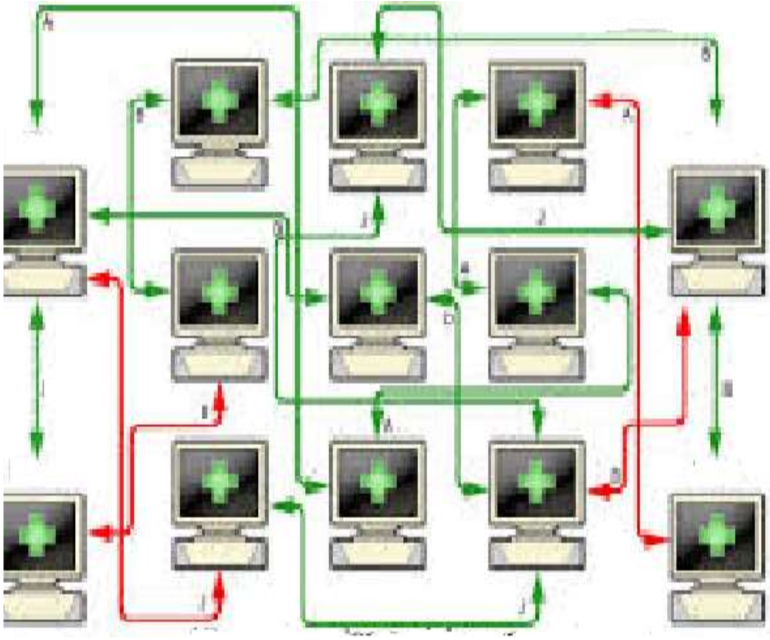


Figure 2. Architecture of The Onion Router (TOR).

METHODOLOGY

To achieve our solution we employed the Rapid Application Development (RAD) methodology for this study. It provides the ability to change the system design as demanded by the user. In addition, the Rapid Application Methodology has a focused scope, such that the purpose is well defined and narrow, providing a comprehensive functionality of the system which are clearly noticeable at the users interface. MySQL was used for the database (the back-end design) while Java Development Kit (JDK) using *Netbeans 6.8 IDE* and PHP was used for the front-end design.

Description of current system: The Onion Router (TOR)

TOR is a popular free software with a network of servers that allows web users perform anonymous communication. The Onion Router is known to be one of the most accepted overlay networks for anonymizing TCP traffic and also for safe and secure online browsing (Bauer et al., 2007). It is noteworthy to mention that TOR does not provide internet security but it offers a wide range of anonymity. It is perceived tough anonymity properties and its moderately low latency service makes it very useful as an anonymizing tool. Low latency is reached through TOR's ability to poise the traffic load by bettering the TOR router selection to probabilistically support routers with high-bandwidth potentials. It directs internet traffic through a liberated, worldwide, volunteer network consisting of over six thousand relays to hide a user's location and usage from any person conducting network observation or traffic analysis. The onion routing design is simply wrapping traffic in encrypted layers in order to guard the contents of the data as well as the anonymity of the sender and recipient.

In the TOR architecture, shown in Figure 2, several basic models are defined as follows: A TOR proxy is the client component of the network that places the user's traffic into the network of TOR routers. A TOR router is the server section of the network that is responsible for promoting traffic within the central fraction of the network. We can analyze the Tor proxy as a service that runs on the user's computer.

Solution provided by the new system

This study is focused on improving and proffering solutions to the problem faced by the current system. The current system is faced with the problem of exposure of identity and it also lacks the ability to monitor web users. The technical details in solving the problem are discussed below.

Firstly, the solution to the problem above is to create a web proxy that provides and guarantees the security of users surfing anonymously on World Wide Web.

The second aspect to the solution is to identify the best possible way to browse anonymously by developing a transparent proxy. The virtualisation of the proxy will allow as many users as possible to surf the internet securely through a more effective transmission speed and make anonymous surfing easier unlike TOR which will only allow a limited amount of security and reduce the transmission speed.

Overview of the new system

The new system, RUN Proxy, was developed basically to implement identity management with features to aid the anonymization of users to enable them surf the internet anonymously without being barred. An added advantage of the system is that it can track and monitor the operations of the user.

It was designed and developed essentially to interface with the client system (that is, web browser). The RUN Proxy was designed using a transparent proxy model with some additional translational features where no modification was done to the response or request of resources, other than the addition of its identification information or that of the server from which the message was recovered and mediation of resource. The RUN Proxy would certify that the length of the message remains unchanged also.

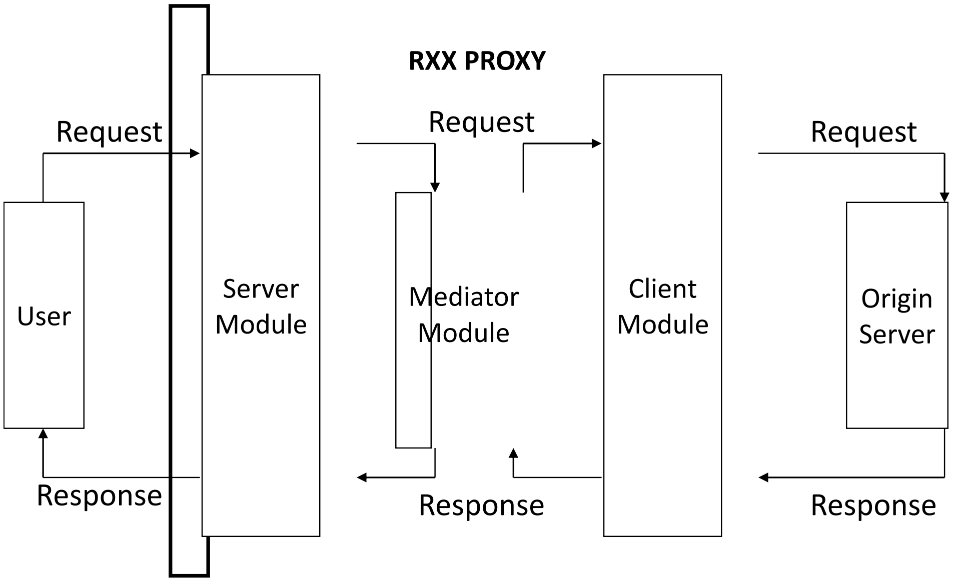


Figure 3. RUN Proxy Architecture Dataflow diagram.

Description of the proposed system

The RUN Proxy, shown in Figure 3, was designed and implemented with some features of HTTP 1.1 and performs the same function as the HTTP Proxy Server. The HTTP protocol is a request/response protocol communicating between the client and server. The client launches a request to the server in the form of a request method, followed by a message containing request modifiers, client information, and possible body content over a connection with a server. The server responds with a response line, including the message's protocol version and a success or error code, trailed by a message having the information about the server, and possible entity-body content. The RUN proxy will act as an intermediary between the client and the server, acting as a server to the client and as client to the server.

Algorithm for the RUN Proxy

```
1. Start
2. Proxy listens to connections from clients (web browser)
3. Proxy accepts the connection;
        If connection is found, then proxy accepts the connection
4. Proxy creates a new thread for the connection
5. Proxy gets a request from the client after the connection has been established
6. Proxy creates a default url "" and default port "80"
7. Proxy tests the connection by performing the following;
        a. Proxy checks if the first line is "connect",
                If connect exist, proxy closes the connection
        b. Else, proxy checks if the first line is a "request line"
        c. Else, proxy prints "unknown first line"
8. The proxy processes the request line
9. Proxy enters a loop in other to test the request message
10. Proxy tries to fork a connection with the origin server
11. Origin server creates a new thread for connection between proxy and itself if connection is accepted
12. Proxy relays the request message to the origin server via the created connection
13. The origin server processes the request message from the proxy
14. Origin server sends a response message containing the message header and message body
15. Proxy deciphers the message header from the message body via the last line which is "<cr><lf>"
16. Proxy processes the response header from the origin server by reading each line of the header field until it reaches an empty line
17. Proxy processes the response body by determining whether the message body is chunked or not from the content length read in the header.
18. Proxy processes the response body from the server. It performs the following;
                a. If message body is not chunked, the proxy processes the body
                b. Else, if the message proxy is chunked, the proxy processes it until a "0" is found.
19. Proxy sends response message to the client
20. Proxy requests from client if the connection should be closed or not;
        If connection should be closed, proxy closes the connection
        Else proxy maintains the connection if the following are true;
                a. If server connection is lost
                b. If clients does not respond to proxy in 10 seconds
21. Stop
```

Flowchart for the RUN Proxy

The RUN Proxy flowchart is a form of diagram that represents the steps in the algorithm showing a diagrammatic illustration of the steps taken by the proxy in relaying the request message from the client (web browser) to the origin server, and also the sending of the response message from the origin server back to the client. The RUN proxy flow chart is shown in Figure 4.

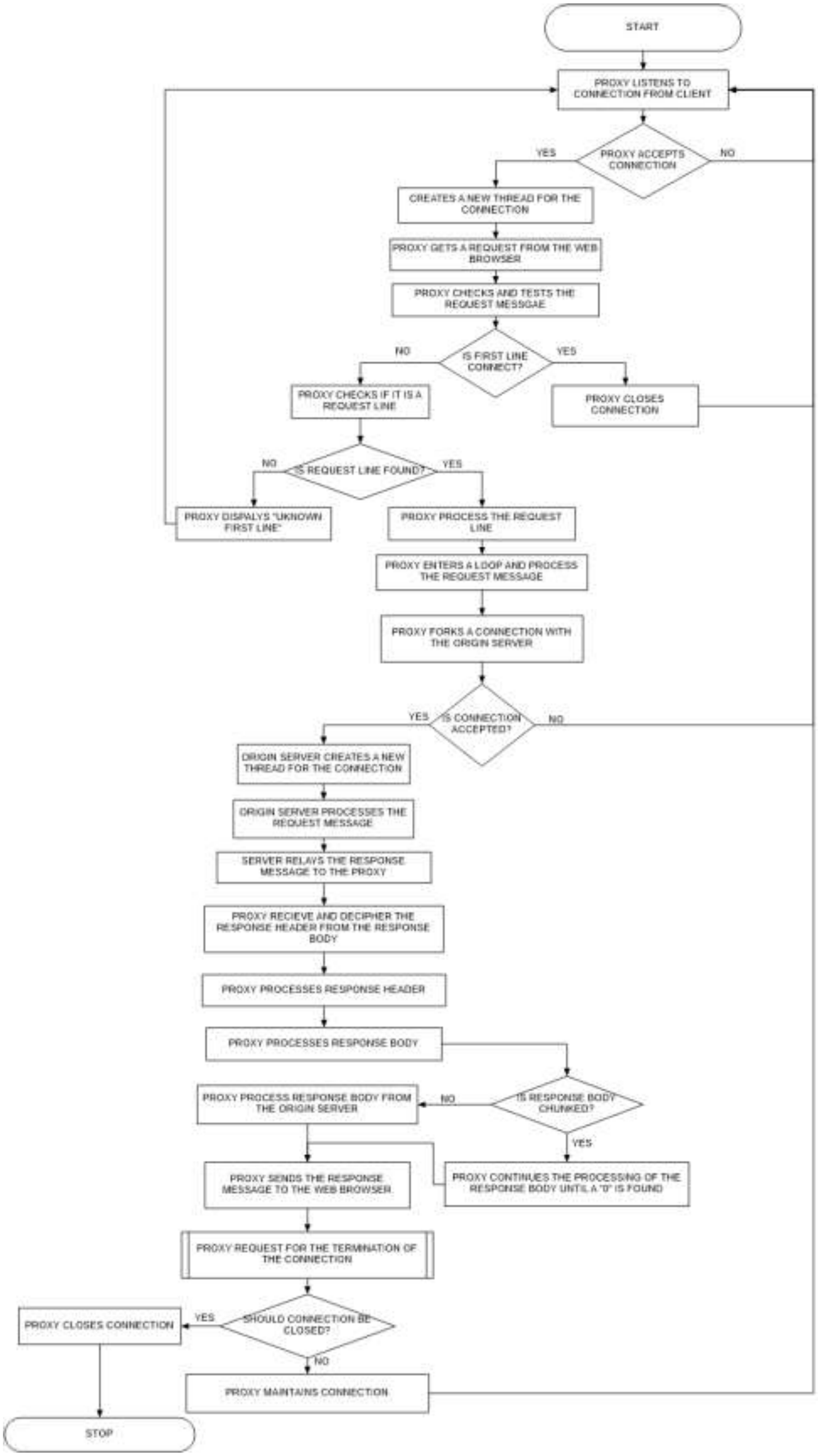

START
PROXY LISTENS TO CONNECTION FROM CLIENT
PROXY ACCEPTS CONNECTION
YES
NO
CREATES A NEW THREAD FOR THE CONNECTION
PROXY GETS A REQUEST FROM THE WEB BROWSER
PROXY CHECKS AND TESTS THE REQUEST MESSGAE
IS FIRST LINE CONNECT?
NO
YES
PROXY CHECKS IF IT IS A REQUEST LINE
PROXY CLOSES CONNECTION
IS REQUEST LINE FOUND?
NO
YES
PROXY DISPALYS "UKNOWN FIRST LINE"
PROXY PROCESS THE REQUEST LINE
PROXY ENTERS A LOOP AND PROCESS THE REQUEST MESSAGE
PROXY FORKS A CONNECTION WITH THE ORIGIN SERVER
IS CONNECTION ACCEPTED?
YES
NO
ORIGIN SERVER CREATES A NEW THREAD FOR THE CONNECTION
ORIGIN SERVER PROCESSES THE REQUEST MESSAGE
SERVER RELAYS THE RESPONSE MESSAGE TO THE PROXY
PROXY RECIEVE AND DECIPHER THE RESPONSE HEADER FROM THE RESPONSE BODY
PROXY PROCESSES RESPONSE HEADER
PROXY PROCESSES RESPONSE BODY
IS RESPONSE BODY CHUNKED?
NO
YES
PROXY PROCESS RESPONSE BODY FROM THE ORIGIN SERVER
PROXY CONTINUES THE PROCESSING OF THE RESPONSE BODY UNTIL A "0" IS FOUND
PROXY SENDS THE RESPONSE MESSAGE TO THE WEB BROWSER
PROXY REQUEST FOR THE TERMINATION OF THE CONNECTION
SHOULD CONNECTION BE CLOSED?
YES
NO
PROXY CLOSES CONNECTION
PROXY MAINTAINS CONNECTION
STOP

Figure 4. RUN Proxy flowchart.

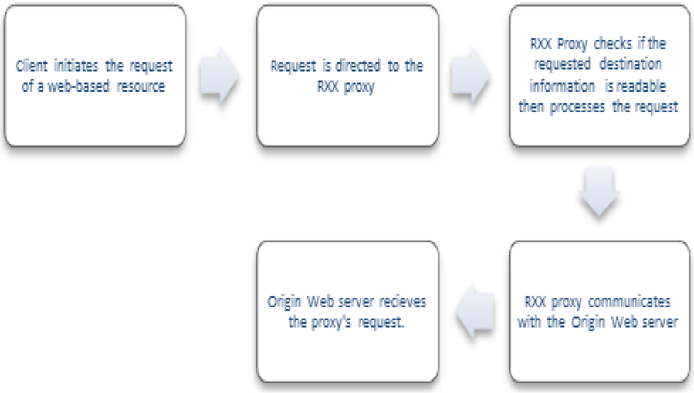

Client initiates the request of a web-based resource
Request is directed to the RXX proxy
RXX Proxy checks if the requested destination information is readable then processes the request
RXX proxy communicates with the Origin Web server
Origin Web server recieves the proxy's request.

Figure 5. Client (Web brower) requesting for a resource through the RUN proxy.

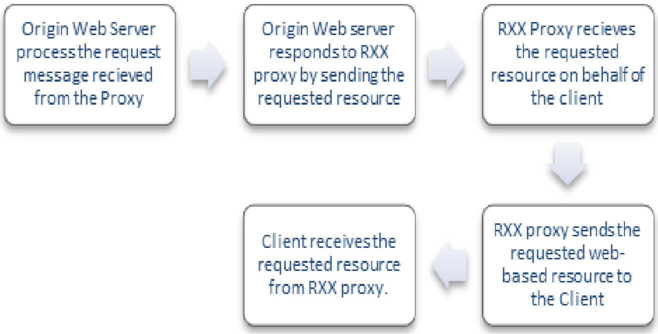

Origin Web Server process the request message recieved from the Proxy
Origin Web server responds to RXX proxy by sending the requested resource
RXX Proxy recieves the requested resource on behalf of the client
RXX proxy sends the requested web-based resource to the Client
Client receives the requested resource from RXX proxy.

Figure 6. Server responding to the resource requested by the Client via RUN proxy.

Process flow diagrams

User requirements

The user requirement for the RUN Proxy is that the user must ensure its web browser is configured to use the proxy (Figure 5). This is done by setting the browser to accept HTTP traffic on the proxy's port and IP address (Figure 6).

FINDINGS AND DISCUSSION OF SYSTEM TESTING RESULTS

Here describes the results obtained from the tests carried out on the system, and it gives a detail report on the tests and also the interface which displays the usage.

The RUN Proxy interface displays how it listens to connections. Figure 7 illustrates how RUN Proxy behaves when no connection has been established. It displays an empty screen because there is no connection of any sort with the proxy at the current time. To establish a connection, a web client (browser) must initiate a connection with the proxy. The configuration of a web proxy can be done by setting the LAN setting of the computer system to the proxy's IP address and port number.

Figure 8 displays the RUN Proxy's functionalities when the web browser whose connection was routed through the proxy establishes a connection. It shows how a resource is requested by the client to an origin server via the RUN Proxy. The URL accessed by the proxy for mediation is (http://www.greens.org/about/software/editor.txt). It also shows the information of both the client and the origin server which aids the effective tracking of client's web patterns and activities.

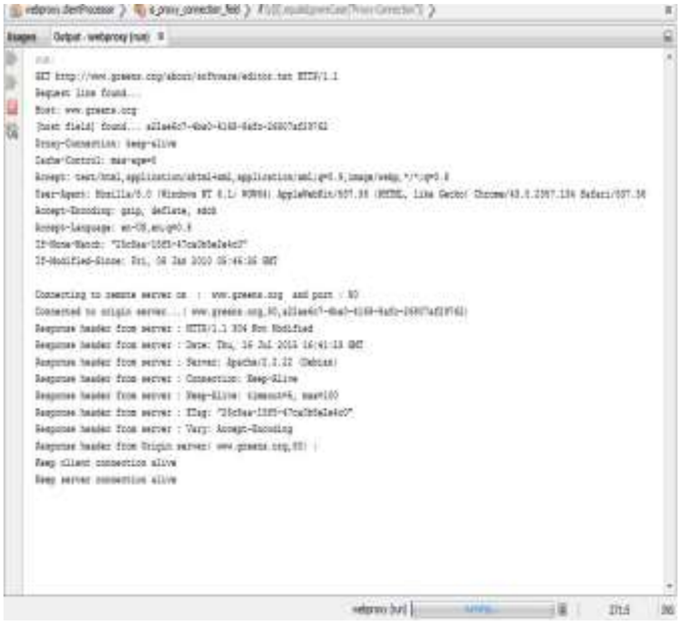
The highlighted portion in Figure 9 is responsible for the creation of a socket which is an abstraction of a client–server communication. The portion presents how the RUN proxy checks if the request line was found.

Figure 10 explains the process on how the RUN Proxy detects the request line in a request message. The source code illustrates the method it uses to check for some HTTP methods to determine the presence of a request line.

Figure 11 shows how the RUN proxy displays the processing of resources that are chunked. The URL for

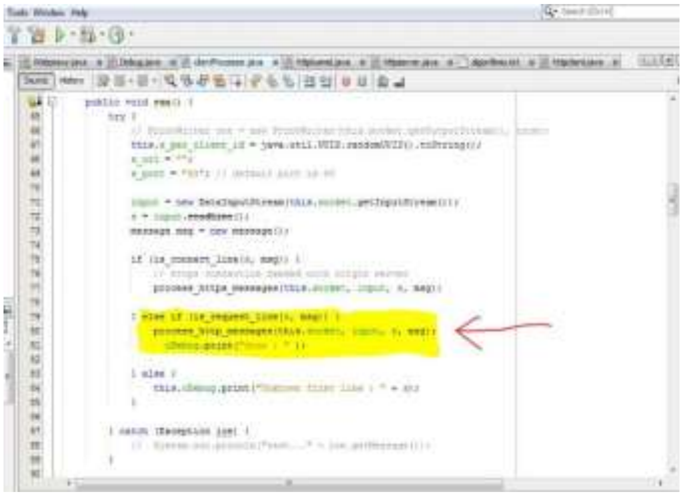
```
run:
```

Figure 7. RUN Proxy waiting to establish a connection.



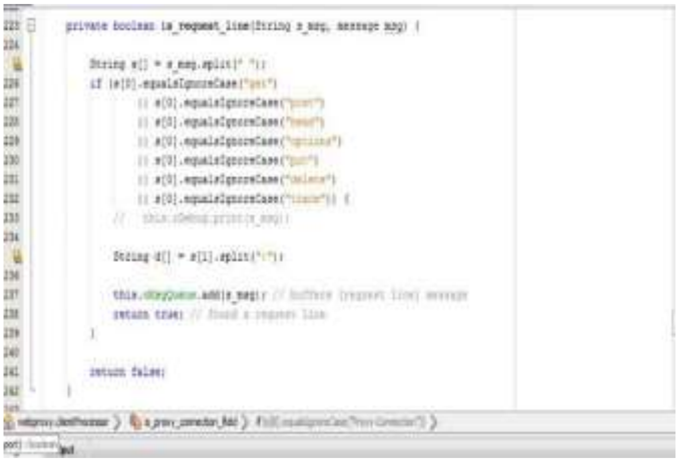
```
run:
GET http://www.greens.org/about/software/editor.txt HTTP/1.1
Request line found...
Host: www.greens.org
[host field] found... a21ae6c7-4ba0-4168-8afb-26807af3f762
Proxy-Connection: keep-alive
Cache-Control: max-age=0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/43.0.2357.134 Safari/537.36
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8
If-None-Match: "25c8ae-1585-47ca2b5e1e4c0"
If-Modified-Since: Fri, 08 Jan 2010 05:46:36 GMT

Connecting to remote server on  :  www.greens.org  and port : 80
Connected to origin server...( www.greens.org,80,a21ae6c7-4ba0-4168-8afb-26807af3f762)
Response header from server : HTTP/1.1 304 Not Modified
Response header from server : Date: Thu, 16 Jul 2015 16:41:13 GMT
Response header from server : Server: Apache/2.2.22 (Debian)
Response header from server : Connection: Keep-Alive
Response header from server : Keep-Alive: timeout=5, max=100
Response header from server : ETag: "25c8ae-1585-47ca2b5e1e4c0"
Response header from server : Vary: Accept-Encoding
Response header from Origin server( www.greens.org,80) :
Keep client connection alive
Keep server connection alive
```

Figure 8. RUN proxy functionalities.



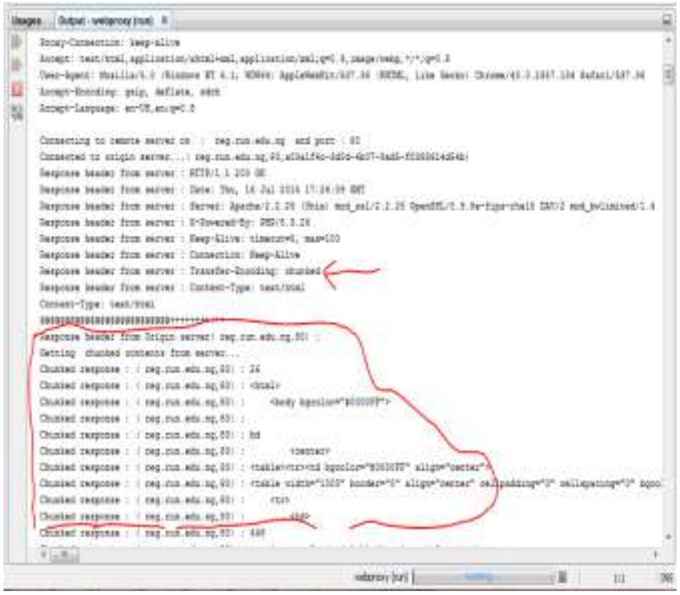
```
    public void run() {
        try {
            // PrintWriter out = new PrintWriter(this.socket.getOutputStream(), true);
            this.s_per_client_id = java.util.UUID.randomUUID().toString();//
            s_url = "";
            s_port = "80"; // default port is 80

            input = new DataInputStream(this.socket.getInputStream());
            s = input.readLine();
            message msg = new message();

            if (is_connect_line(s, msg)) {
                // https connection needed with origin server
                process_https_messages(this.socket, input, s, msg);

            } else if (is_request_line(s, msg)) {
                process_http_messages(this.socket, input, s, msg);
                  sDebug.print("Done : ");

            } else {
                this.sDebug.print("Unknown first line : " + s);
            }

        } catch (Exception ioe) {
            // System.out.println("Exit..." + ioe.getMessage());
        }
```

Figure 9. Source code for the creation of a socket.



```
    private boolean is_request_line(String s_msg, message msg) {

        String s[] = s_msg.split(" ");
        if (s[0].equalsIgnoreCase("get")
                || s[0].equalsIgnoreCase("post")
                || s[0].equalsIgnoreCase("head")
                || s[0].equalsIgnoreCase("options")
                || s[0].equalsIgnoreCase("put")
                || s[0].equalsIgnoreCase("delete")
                || s[0].equalsIgnoreCase("trace")) {
          //   this.sDebug.print(s_msg);

          String d[] = s[1].split(":");

          this.sKeyQueue.add(s_msg); // buffers (request line) message
          return true; // found a request line
        }

        return false;
    }
```

Figure 10. Source code portion on how the request line is detected.



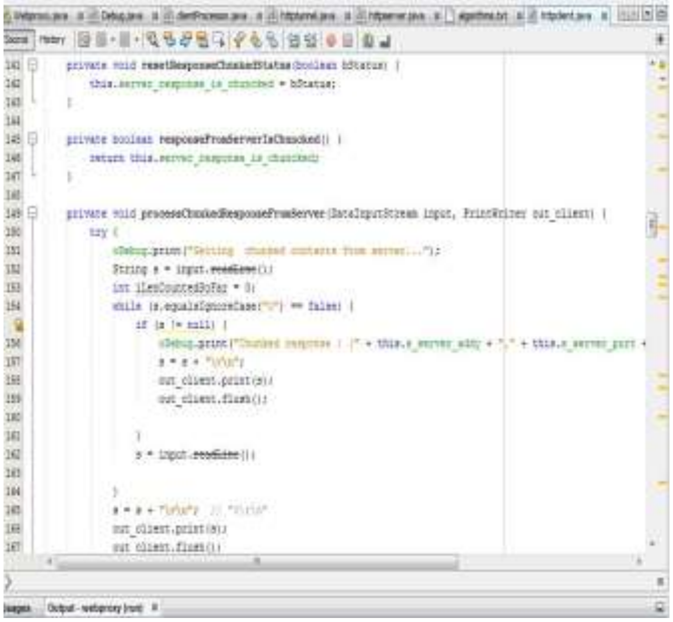
```
Proxy-Connection: keep-alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/43.0.2357.134 Safari/537.36
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8

Connecting to remote server on  :  reg.run.edu.ng  and port : 80
Connected to origin server...( reg.run.edu.ng,80,a03a1f4c-2d0d-4b07-8ad5-f0000614d54b)
Response header from server : HTTP/1.1 200 OK
Response header from server : Date: Thu, 14 Jul 2016 17:26:39 GMT
Response header from server : Server: Apache/2.2.29 (Unix) mod_ssl/2.2.29 OpenSSL/1.0.1e-fips-rhel5 DAV/2 mod_bwlimited/1.4
Response header from server : X-Powered-By: PHP/5.3.29
Response header from server : Keep-Alive: timeout=5, max=100
Response header from server : Connection: Keep-Alive
Response header from server : Transfer-Encoding: chunked
Response header from server : Content-Type: text/html
Content-Type: text/html
$$$$$$$$$$$$$$$$$$$$$$$$$$+++++++
Response header from Origin server( reg.run.edu.ng,80) :
Getting  chunked contents from server...
Chunked response : ( reg.run.edu.ng,80) : 26
Chunked response : ( reg.run.edu.ng,80) : <html>
Chunked response : ( reg.run.edu.ng,80) :     <body bgcolor="#0000FF">
Chunked response : ( reg.run.edu.ng,80) :
Chunked response : ( reg.run.edu.ng,80) : bd
Chunked response : ( reg.run.edu.ng,80) :         <center>
Chunked response : ( reg.run.edu.ng,80) : <table><tr><td bgcolor="#0000FF" align="center">
Chunked response : ( reg.run.edu.ng,80) : <table width="1000" border="0" align="center" cellpadding="0" cellspacing="0" bgco
Chunked response : ( reg.run.edu.ng,80) :     <tr>
Chunked response : ( reg.run.edu.ng,80) :         <td>
Chunked response : ( reg.run.edu.ng,80) : 48f
```

Figure 11. RUN Proxy displaying the chunk process.

this testing is (http://reg.run.edu.ng/FP_growth/FP_growth.php?mat_no=run-300/kunle).

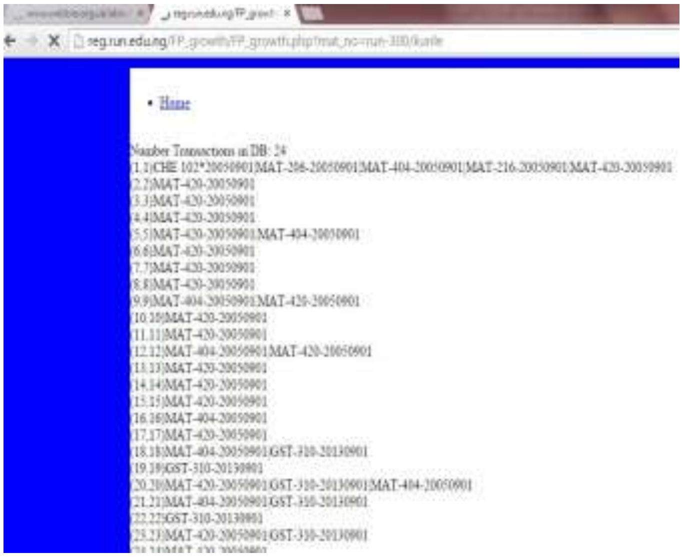
Figure 12 is the portion of the source code that handles the chunked messages. Firstly, the proxy detects if the response message is chunked from the content length received from the origin server. After ascertaining that the response message received from the origin server is chunked, it processes it by reading the input DataStream line by line until a "0" is found which lets the proxy know that all the chunked response has been read. Finally, the proxy prints the chunked response and relays it to the client.

Figure 13 displays the requested resource by a web client on the web browser via the RUN Proxy. The output



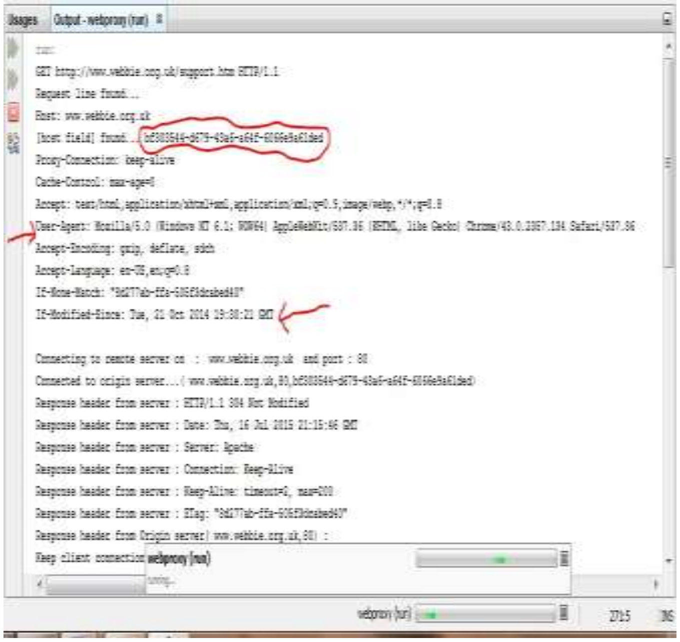
```
    private void resetResponseChunkedStatus(boolean bStatus) {
        this.server_response_is_chunked = bStatus;
    }

    private boolean responseFromServerIsChunked() {
        return this.server_response_is_chunked;
    }

    private void processChunkedResponseFromServer(DataInputStream input, PrintWriter out_client) {
        try {
            sDebug.print("Getting  chunked contents from server...");
            String s = input.readLine();
            int iLenCountedSoFar = 0;
            while (s.equalsIgnoreCase("0") == false) {
                if (s != null) {
                    sDebug.print("Chunked response : [" + this.s_server_addr + ":" + this.s_server_port +
                    s = s + "\r\n";
                    out_client.print(s);
                    out_client.flush();

                }
                s = input.readLine();

            }
            s = s + "\r\n"; // "\r\n"
            out_client.print(s);
            out_client.flush();
```

Figure 12. A view of the chunked process source code.



```
Number Transactions in DB: 24
1.1|CHE 102*20050901|MAT-298-20050901|MAT-404-20050901|MAT-216-20050901|MAT-420-20050901
2.2|MAT-420-20050901
3.3|MAT-420-20050901
4.4|MAT-420-20050901
5.5|MAT-430-20050901|MAT-404-20050901
6.6|MAT-420-20050901
7.7|MAT-420-20050901
8.8|MAT-430-20050901
9.9|MAT-404-20050901|MAT-420-20050901
10.10|MAT-420-20050901
11.11|MAT-420-20050901
12.12|MAT-404-20050901|MAT-420-20050901
13.13|MAT-420-20050901
14.14|MAT-420-20050901
15.15|MAT-420-20050901
16.16|MAT-404-20050901
17.17|MAT-420-20050901
18.18|MAT-404-20050901|GST-310-20130901
19.19|GST-310-20130901
20.20|MAT-420-20050901|GST-310-20130901|MAT-404-20050901
21.21|MAT-404-20050901|GST-310-20130901
22.22|GST-310-20130901
23.23|MAT-420-20050901|GST-310-20130901
24.24|MAT-420-20050901
```

Figure 13. The output result displayed on the web browser.



```
run:
GET http://www.webbie.org.uk/support.htm HTTP/1.1
Request line found...
Host: www.webbie.org.uk
[host field] found...bf383544-d679-43a6-a64f-6056e8a61ded
Proxy-Connection: keep-alive
Cache-Control: max-age=0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/43.0.2357.134 Safari/537.36
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8
If-None-Match: "3d277ab-ffa-5065f3dcabed40"
If-Modified-Since: Tue, 21 Oct 2014 19:38:21 GMT

Connecting to remote server on  :  www.webbie.org.uk  and port : 80
Connected to origin server...( www.webbie.org.uk,80,bf383544-d679-43a6-a64f-6056e8a61ded)
Response header from server : HTTP/1.1 304 Not Modified
Response header from server : Date: Thu, 16 Jul 2015 21:15:46 GMT
Response header from server : Server: Apache
Response header from server : Connection: Keep-Alive
Response header from server : Keep-Alive: timeout=2, max=200
Response header from server : ETag: "3d277ab-ffa-5065f3dcabed40"
Response header from Origin server( www.webbie.org.uk,80) :
Keep client connection
```

Figure 14. Proxy recording the process between itself and Origin Server.

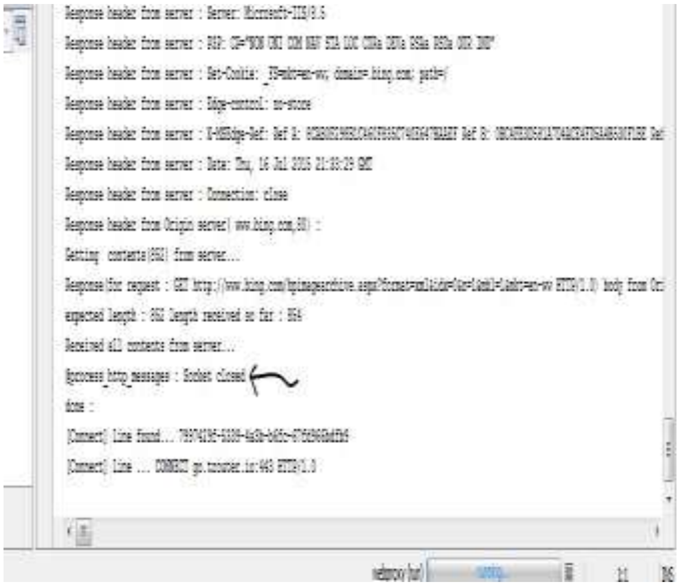
result received by the client when it request for the URL (http://reg.run.edu.ng/FP_growth/FP_growth.php?mat_no =run-300/kunle) from the Apache server via the proxy. The resource requested by the web browser via the RUN Proxy is displayed on the webpage.

Figure 14 show how the proxy keeps record of its connection with the Origin Server. The Proxy is constructed to keep records of some request-header fields such as the host field which represents the naming authority of the origin server or gateway given by the original URL, the user agent which is the means by which the resource was requested, the If-Modified-Since request-header field, which is used with a method to make it conditional; that is, if the requested resource has not been modified since the time specified in this field, an entity will not be returned from the server and a host of other fields. The pointers show the different header field used by the RUN Proxy.

Figure 15 shows the instance where a socket connection is closed between the requesting client and the responding server. When the socket is closed, it means that the established thread connection created by the proxy to communicate with both the client and the origin server is closed. A socket is an abstraction of client-server communication. The pointer in the Figure 15 implies that the socket connection is closed and the RUN proxy will then continue to listen for other connections.

Conclusion

The use of a proxy should be adapted by organizations and institutions, as this would save lots of time in tracking and monitoring the web activities of their clients. With the RUN Proxy, any client can have access to surf any website with barring policy anytime; so long as it requests the resource via the proxy. This helps to mitigate the loss of vital information provided by origin servers capable of supplying this resource. This study has tackled the problem of providing anonymous communication and that of monitoring clients' web activities. Furthermore, the



```
Response header from server : Server: Microsoft-IIS/8.5
Response header from server : P3P: CP="NON UNI COM NAV STA LOC CURa DEVa PSAa PSDa OUR IND"
Response header from server : Set-Cookie: _FS=mkt=en-ww; domain=.bing.com; path=/
Response header from server : Edge-control: no-store
Response header from server : X-MSEdge-Ref: Ref A: [illegible] Ref B: [illegible] Ref
Response header from server : Date: Thu, 16 Jul 2015 21:03:29 GMT
Response header from server : Connection: close
Response header from Origin server( www.bing.com,80) :
Getting  contents(852) from server...
Response(for request : GET http://www.bing.com/hpimagearchive.aspx?format=xml&idx=0&n=1&mbl=1&mkt=en-ww HTTP/1.0) body from Ori
expected length : 852 Length received so far : 854
Received all contents from server...
@process_http_messages : Socket closed
done :
[Connect] line found... [illegible]
[Connect] line ... CONNECT [illegible]:443 HTTP/1.0
```

Figure 15. RUN Proxy closing socket connection

benefits of using a proxy can be realized for large populations. These findings, coupled with intuition about the benefits of web proxies to motivate the adoption of anonymous communication in local environments.

The RUN Proxy is able to overcome the major problem with the TOR proxy, in that it can monitor the activities of its users on the Web. This is not possible with TOR as its distributed architecture makes user tracking virtually impossible. In this way, the TOR is often used for illegal activities.

The limitations of this study are that the RUN Proxy is faced with the following constraints as a result of restrictions. These are as follows:

i) The RUN Proxy would not be able to view SSL (Secure Socket Layer) websites;
ii) The RUN Proxy will not be able to effectively track websites with images and picturesque components;
iii) It is also limited by the use of one programming language.

The future direction of this work can be to apply it to increase proficiency in anonymous communications. The incorporation of a system with SSL (Secure Socket Layer) capabilities (that is, a web proxy that is capable of providing resources from SSL websites) and viewing of websites with images can be an important way to improve the system.

Conflict of interest

The authors have not declared any conflict of interest

REFERENCES

Al-Khouri AM (2012). PKI In Government Identity Management Systems. Emirates Identity Authority, Abu Dhabi, United Arab Emirates.

Baentsch M, Baum L, Molter G, Rothkugel S, Sturm P (1997). World Wide Web Caching - The Application level view of the Internet. IEEE Commun. 35:6.

Balachander K, Rexford J (2001). Web protocols and Practice. Addison Wesley.

Bauer K, McCoy D, Grunwald D, Kohno T, Sicker D (2007). Low-Resource Routing Attacks Against Tor. ACM 1-10.

Berthold O, Federrath H, Kpsell S (2001). Web mixes: A system for anonymous and unobservable internet access. In International Workshop on Designing Privacy Enhancing Technologies, pringer-Verlag New York, Inc. pp. 115-129.

Dingledine R, Mathewson N, Syverson P(2004). Tor: the second-generation onion router. In *SSYM'04: Proceedings of the 13th conference on USENIX Security ymposium*, Berkeley, CA, USA, 2004. USENIX Association pp. 21-21.

Edman M, Yener B (2009). On anonymity in an electronic society: A surveyof anonymous communication systems. ACM Computing Surveys pp. 1-35.

Fielding R, Gettys J, Mogul J, Frystyk H, Mansiter L, Leach P, Berners-Lee T (1998). Hypertext Transfer Protocol - HTTP/1.1. HTTP Working Group. Retrieved from http://www.w3.org/Protocols/HTTP/1.1

Gaurav S, Pruthi S (2012). A Review on Various Identity Management Systems. Int. J. Innovative Technol.Exploring Eng. (IJITEE) I(2):113-115.

Gautam N, Petander H, Noel J(2013). Comparison of the cost and energy efficiency of prefetching streaming of mobile video. In Proceedings of the 5thWorkshop on Mobile Video, ACM pp. 7-12.

Gross J, Rosson M (2007). End user concern about security and privacy threats. ACM pp. 167-168.

Jelenkovic RP, Radovanovic A (2009). Asymptotic optimality of the static frequency caching in the presence of correlated requests. Operations Res. Lett. 37(5):307-311.

Li B, Erdin E, Gunes MH, Bebis G, ShipleyT (2011). An analysis ofanonymity technology usage. Springer-Verlag 108-21.

Liska T, Sochor T, Sochorova H (2010). Comparison between normal and TOR-anonymized web client traffic. Procedia Social and Behavioral Sci. pp. 542-546.

Luotonen A, Altis K (1994). World Wide Web Proxies. *First International Conference on the World Wide Web.* Retrieved from http://www.cern.ch/papersWWW94/luotonen.ps

Müller J, Böhm K (2011). *Using Federated Identity Management in a Business-Process-Management System – Requirements, Architecture, and Implementation.* Faculty of Informatics, Karlsruhe Institute of Technology.

NCTU-education S (2007). Retrieved from http://people.cs.nctu.edu.tw: http://people.cs.nctu.edu.tw/~chwong/course/sysadm/slide/Web%20Proxy.pdf

Piatek M (2004). Distributed Web Proxy Caching in a Local Network Environment. Pittsburgh: Department of Mathematics and Computer Science, Duquesne University.

SANS-Institute (2003). Exploring Identtity Management. SANS reading room pp. 1-14.

Shapiro M (1986). Structure and encapsulation in distributed systems: The Proxy Principle. Int. Conf. distributed Computer Syst. pp. 198-204.

Shubina A, Smith S (2003). Using caching for browsing anonymity. SIGecom Exch pp. 11-20.

Sochor T (2012). Anonymization of web client traffic efficiency study. 19th International Conference on Computer Networks.Berlin Heidelberg: Springer Verlag pp. 237-246.

Sochor T(2013). Automatic Control of Configuration of web anonymization. Int. J. New Computer Architectures Appl. (IJNCAA) pp. 1-10.

Squid-Cache WIki (2014). Retrieved from Wiki-Squid: http://wiki.squid-cache.org/WhySquid

Wikipedia (2014). Retrieved from Wikipedia: http://en.m.wikipedia.org/wiki/identity_management_system

academicJournals

Vol. 11(13), pp. 168-177, 16 July, 2016
DOI: 10.5897/IJPS2015.4274
Article Number: F492B6D59616
ISSN 1992 - 1950
Copyright ©2016
Author(s) retain the copyright of this article
http://www.academicjournals.org/IJPS

International Journal of Physical Sciences

Full Length Research Paper

Light source estimation using feature points from specular highlights and cast shadows

Anusorn Bunteong[1]* and Nopporn Chotikakamthorn[2]

Faculty of Information Technology, King Mongkut's Institute of Technology Ladkrabang, 1 Chalongkrung Road, Bangkok, 10520, Thailand.

Received 5 February, 2015; Accepted 3 May, 2016

A method for light sources estimation is proposed in this paper. The method utilizes feature points in cast shadows to estimate near light source positions from estimated source directions using specular highlights. There are several methods that can be used to estimate light sources from scene images, using either cast shadows or specular highlights. However, most of them are limited to directional light sources. The proposed method can estimate the positions and intensities of multiple near point light sources. Specular highlights on an object of known geometry are first used for light source direction estimation. Then, a discontinuity point in the object shape and the corresponding cast shadow on a ground plane are used for light source position estimation. Feature points obtained from an image of the cast shadow, however, can be inaccurate due to various factors. Information on diffused light reflected from Lambertian ground-plane surface is subsequently used to improve estimation accuracy. Experimental results were used to evaluate the performance of the proposed method.

Key words: Light source estimation, light source recovery, augmented reality.

INTRODUCTION

Light source estimation is a problem of interest in the fields of computer graphics and computer vision. In augmented reality applications, where generation of a mixed environment containing virtual objects in real scenes is needed, illumination information and surface reflectance properties of real objects are needed for consistent and realistic shading of virtual objects. Another application which requires lighting information is a retrieval of shape information from shading. The main goal of light source estimation is to recover location, direction and intensity of light source(s) given one or more images of a real scene.

Many methods have been developed to estimate properties of single and multiple light sources. These methods use information from either one or a combination of shading, cast shadows, and specular reflections. Majority of these methods are based on shading information. For example, Zhang and Yan (2001) developed a method for parallel light direction estimation using shading on surface of a spherical object in the

*Corresponding author. E-mail: anusorn.b@gmail.com.

scene. The sphere used in this method is assumed to have a Lambertian surface. With known sphere geometric and positional information in the scene, it is possible to estimate locations of pixels known as ‘critical points’ from a shaded image of the sphere. From the obtained critical points, light source directions can be estimated. Wang and Samaras (2002) extends the use of this method to an object of arbitrary shape. The visible points on the object are mapped to a virtual sphere by matching the normal direction at each point. Some critical boundaries may be lost during this process. They provide a method exploiting shadow information to solve the problem. Such hybrid method was reported to offer improved accuracy.

In the study of Bouganis and Brookes (2004), an attempt was made to increase accuracy and reduce limitations of the method in Zhang and Yan (2001). The method for critical point detection in Bouganis and Brookes (2004) method is different from the original one (Zhang and Yan, 2001), but is similar to that of Wei (2003).

Shadow and reflected light information has also been used for light source directions and intensities estimation. For example, Sato et al. (1999, 2001) provide a method for estimating an illumination distribution of a real scene using information of reflected light distribution over a Lambertian planar surface. By using the occlusion information of an incoming light caused by an object of known geometry and location, the method can provide sampled directional light source directions from estimated illumination distribution.

Specular highlights on a shiny surface in the scene can also provide information for light source estimation. For example, Powell et al. (2001) can estimate the positions of light sources from specular highlights on a pair of calibration spheres in the scene. This method estimates the positions and surface normals at the highlights in order to triangulate illuminants. Zhou and Kambhamettu (2002) present a method for locating multiple light sources and estimating their intensities from a pair of stereo images of a sphere. The sphere surface has both Lambertian and specular properties. The specular image is used to find the directions of the light sources, and the Lambertian image is used to find the intensities of the light sources. Another hybrid method is presented by Li et al. (2003). The method integrates cues from shading, shadow and specular reflections for estimating directional illumination in a textured scene.

Even though these previous methods are successful in estimating light source directions and intensities, they assume that the light sources in the scene are far-field so that each light source illuminates all objects at any point from the same direction. However, in a real scene (e.g., under indoor environment), it is quite common to have one or more near point light sources, which illuminate the objects from finite distance. In this case, a common parallel/directional light assumption of those methods is invalid.

Methods have been developed to estimate positions and intensities of near point light sources. Li et al. (2003) proposed two methods for recovering the surface reflectance properties of an object and the light source position from a single view without the distant illumination assumption. The first method is based on the use of the iterative separating-and-fitting relaxation algorithm. The second method estimates the specular reflection parameters and the light source position simultaneously by linearizing the Torrance-Sparrow specular reflection model and by optimizing the sample correlation coefficient. However, the methods were applied to a single light source case, and are applicable only to convex objects. Extension of the methods to the multiple light source case, if possible, can dramatically increase computational complexity.

Takai et al. (2009) present an approach for estimating light sources from a single image of a scene that is illuminated by major near point light sources, and some directional light sources as well as ambient light. They use a pair of reference spheres as light probes. Major step in the method involves differencing the intensities of two image regions of the reference spheres. From an image of a difference sphere, parameters of point light sources are estimated by an iterative operation. The input image is updated by eliminating the lighting effects that is due to the estimated point light sources and the parameters of directional light sources and ambient light are estimated by an iterative operation again. Schniders et al. (2010) proposed an empirical analysis, which shows that line estimation from a single view of a single sphere is not possible in practice and present a closed form solution for recovering a polygon light source from a single view of two spheres and an iterative approach for rectangular light source estimation based on two views of a single sphere.

These methods for near point light source estimation are either limited to a single light source, or applicable under specific and often complex shooting setup. In this paper, we present a method to estimate the position and intensity of multiple near point light sources from a single view image. Unlike the methods of Schniders et al. (2010) and Takai et al. (2009) that use either the two reference spheres in the scene or two views of a single sphere, our method use only a single object of known geometry with specular reflection. Feature points in a cast shadow of that object on a ground plane with diffused reflection are exploited for effective and efficient source position estimation. Use of cast-shadow feature points can also help speeding up the computation. Note that, although there exists a method that use feature points in cast shadows for light source estimation (Cao and Shah, 2005), such method assumes that a light source is directional and two perspective view images of a scene

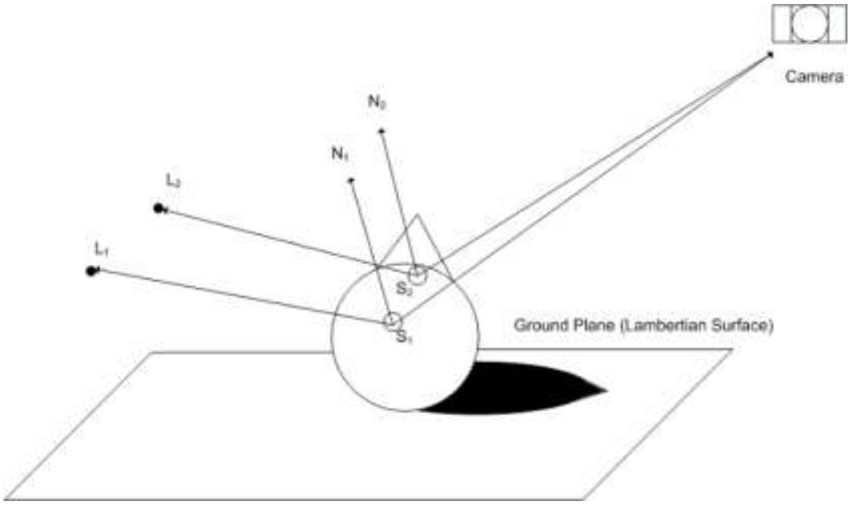


Figure 1. The system setup containing a spherical object of known geometric parameters, and a ground plane with Lambertian-type reflection.

are required. Note that, although using a pair of mirror spheres or cameras with fish-eye lens is an efficient method, it is inconvenient to setup specially when applied to near light source location estimation. The proposed method has the main benefit of simpler equipment setup.

An overview of the proposed method and the corresponding scene setup is first described. The method utilizes a specular highlight from an object of known geometry and location (a sphere in this case), and a ground plane with diffusive reflection (Figure 1 for the scene setup). The spherical object is required to contain some discontinuity on its surface (a sphere with a cone-shaped tip or a box corner in our case). Figure 1 illustrates a case where there are two point light sources at L_1 and L_2, and the corresponding specular peaks S_1 and S_2, as seen by a camera on the right.

USING SPECULAR HIGHLIGHT FOR LIGHT SOURCE DIRECTION ESTIMATION

Specular reflection is the mirror-like reflection of light from a surface, in which light from each incoming direction is reflected into a single outgoing direction. The directions of incoming and outgoing light rays have the same angle with respect to the surface normal (Figure 2). Like those of Hara et al. (2005), in this paper a specular highlight is utilized to estimate a point source direction. In doing so, a specular peak pixel on a taken image of a real scene is first identified. The direction of the corresponding light source, which produces a highlight on the surface point P, can be calculated as:

$$\mathbf{l}_P = 2(\mathbf{n}_P^T \mathbf{v}_P)\, \mathbf{n}_p - \mathbf{v}_P \qquad (1)$$

where $\mathbf{l}_P$ is a vector of the light source direction measured at the point P, $\mathbf{n}_P$ is a surface normal vector at P, and $\mathbf{v}_P$ is a unit vector of a viewing direction, looked from P.

The light source position that causes specular peak at P is located along the direction of $\mathbf{l}_P$. Subsequently, given the so obtained light source direction, a method for light source position estimation from feature points will be explained.

USE OF FEATURE POINTS FOR SOURCE POSITION ESTIMATION

Features are parts or aspects of the image which captures its salient characteristics. Features may be edges, corners, blobs or ridges. Feature detection is an image processing algorithm for identifying the presence and location of certain salient characteristics in an inspected image. In this paper, a corner detector is used to find a corner point(s) on a shadow edge of a spherical object cast on a ground plane. With known object geometry and location, the line connecting a detected corner point on the ground plane to the corresponding point on the object surface will pass through the light source position. When used in combination with the source direction estimated using a specular highlight as explained before, a source position can be obtained as shown in Figure 3.

In addition, with a priori estimate of a source direction obtained using a specular highlight, feature points can be efficiently located by searching for such features only along a certain contour on a shadow-casted surface. In this case where a shadow is casted on a ground plane, such contour can be easily estimated by first constructing

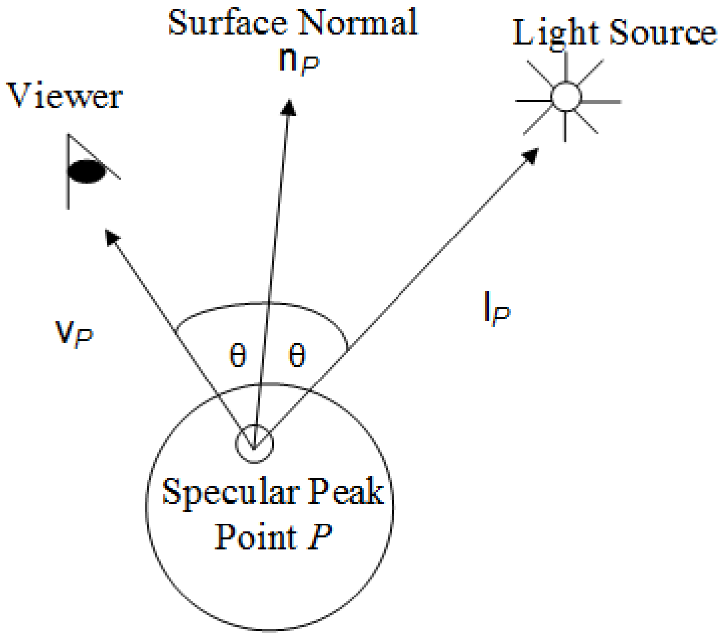


Figure 2. Relationship of three vectors involved in specular reflection.

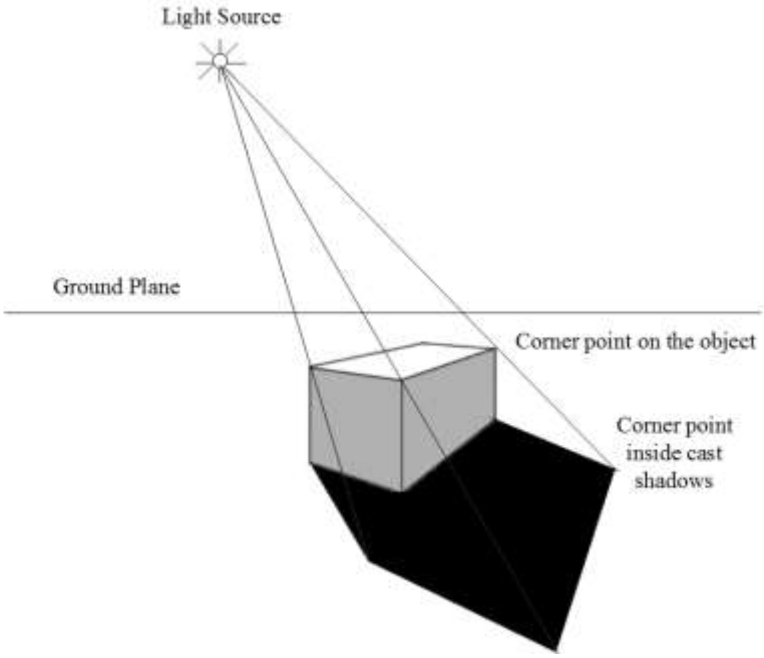


Figure 3. Corner points on the object surface and the corresponding points in the object's shadow, used for light source position estimation.

a plane consisting of an object corner point and any two arbitrary points along a light source path. The contour is then obtained as a line resulting from the intersection of that plane with the ground plane as shown in Figure 4. With this, a corner detector is applied along the contour's projected line on an image plane.

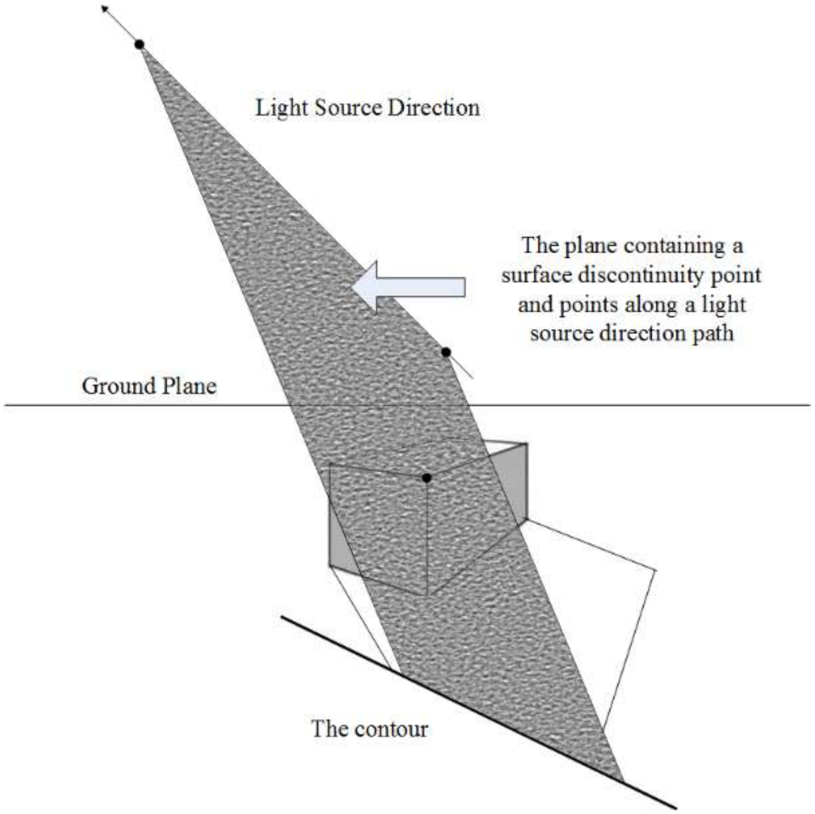

Light Source Direction
The plane containing a surface discontinuity point and points along a light source direction path
Ground Plane
The contour

Figure 4. The contour formed by the intersection of the plane containing a surface discontinuity point and points along a light source direction path, with the ground plane.

With all possible candidates of an actual feature point, a line is formed from each feature point candidate and the discontinuity point on the object surface. With respect to the discontinuity point, these lines are denoted by $\mathbf{f}_{m,n}$, *m* = 1, . . . , *M*, *n* = 1, . . . , *N*, where *M* is the number of detected feature point candidates and *N* is the number of light sources.

With respect to the specular peak position, the estimated n^{th} light source directions are denoted by $\mathbf{l}_{P,n}$. Let P_n be the position of the n^{th} specular peak with respect to the discontinuity point. Given a suitable scalar value β_m, the intersection points between $\mathbf{f}_{m,n}$ and $P_n + \beta_m.\mathbf{l}_{P,n}$ are *M* possible position of the n^{th} light source. These intersection points are denoted by $L'_{1,n}$, …, $L'_{M,n}$ (Figure 5).

In practice, soft shadow may appear instead of a hard one. Feature point detection can be less accurate under this scenario. Furthermore, in real scene, the depth value from depth camera may be inaccurate. This is the reason why some feature point candidates must be kept in the previous discussion. To identify an actual feature point from those candidates, information collected from ground-plane diffuse reflection is used for this purpose as explained subsequently.

LIGHT SOURCE POSITION ESTIMATION FROM GROUND-PLANE DIFFUSE REFLECTION

Similar to Sato et al. (1999, 2001) method, the proposed method utilizes shadow of a known-geometry object, casted on a ground plane. Different ground-plane illumination scenarios are as shown in Figure 6. From the figure, at the point P_1, lights coming from the point light sources at L_1 and L_2 are blocked. On the other hand, the point P_2 is illuminated by the light coming from L_2 alone. At P_3, no occlusion occurs for both light sources.

Here, the ground plane is assumed to have a Lambertian surface. An amount of ground-plane reflected light as observed by the k^{th} pixel is obtained as:

$$I_k = \sum_{n=1}^{N} \frac{K_d E_n \cos(\theta_{k,n}) S_{k,n}}{\|L_n - P_k\|^2} \tag{2}$$

From Equation 2, L_n and P_k are real-world positions of the n^{th} light source and the k^{th} pixel. In addition, E_n and $\theta_{k,n}$ are respectively, the light source intensity and the

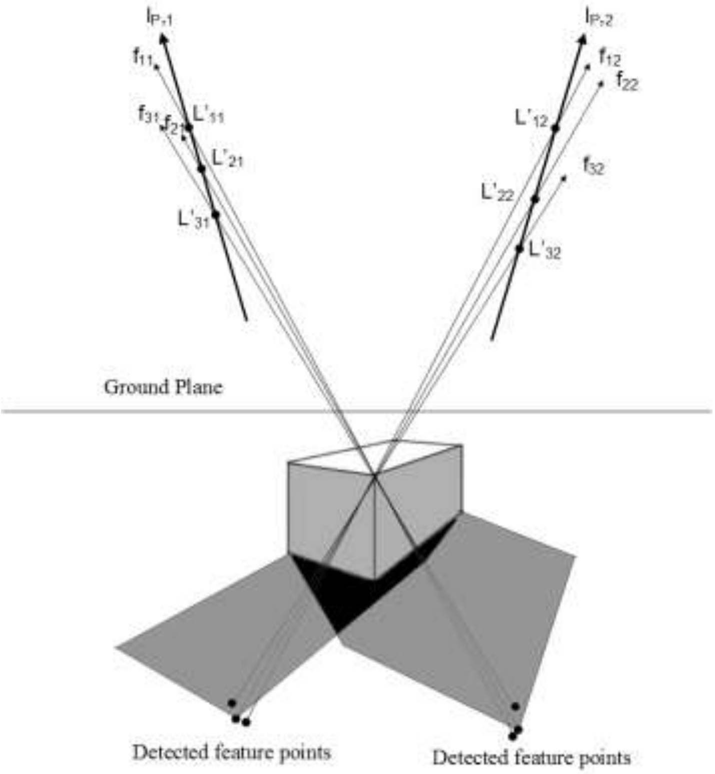


Figure 5. Possible positions of light sources estimated from feature point candidates.

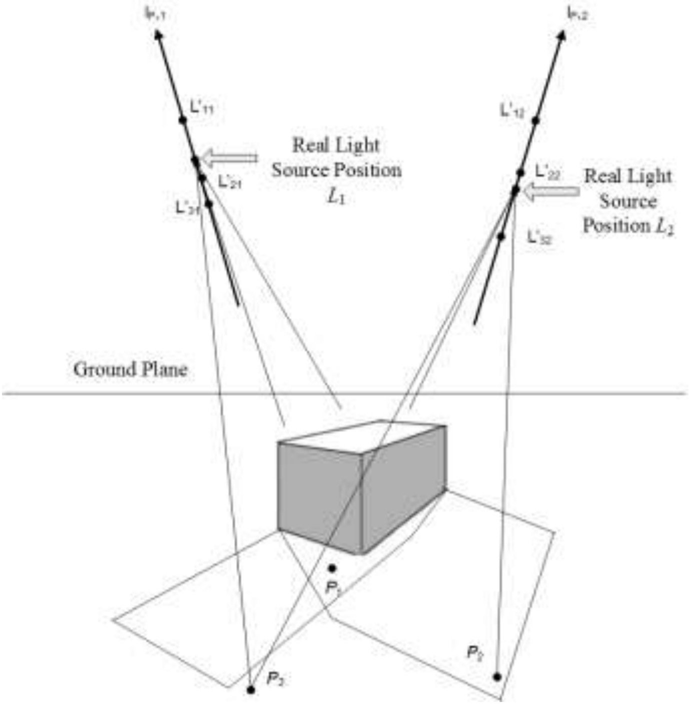


Figure 6. Light sources are occluded at some points on the ground plane.

angle with respect to the ground-plane surface normal. The parameter K_d is the surface diffuse reflection coefficient. The occlusion coefficient $S_{k,n}$ is zero if the n^{th} light source is occluded at the k^{th} pixel and one otherwise.

There is more than one possible value for L_n. For example, in a single light source case and three light source position candidates $L'_{1,n}$, $L'_{2,n}$, $L'_{3,n}$, L_n can take on any one of them. To find which one best represents an actual light source position, Equation 2 is evaluated for each possible value of L_n. The one best fitting Equation 2 in a non-negative least square sense is chosen as a light source position estimate. Similar method can be applied for the case of two or more light sources.

To compute which value is the best estimate of the light source position, solving Equation 2 is equivalent to finding the solution to the following set of linear equations (Schnieders et al., 2010):

$$\mathbf{Ax} = \mathbf{b} \quad (3)$$

where

$$\mathbf{A} = \begin{bmatrix} \alpha_{1,1} & \alpha_{1,2} & \alpha_{1,3} & \cdots & \alpha_{1,N} \\ \alpha_{2,1} & \alpha_{2,2} & \alpha_{2,3} & \cdots & \alpha_{2,N} \\ \alpha_{3,1} & \alpha_{3,2} & \alpha_{3,3} & \cdots & \alpha_{3,N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha_{K,1} & \alpha_{K,2} & \alpha_{K,3} & \cdots & \alpha_{K,N} \end{bmatrix} \quad (4)$$

$$\mathbf{b} = [I_1 \quad I_2 \quad I_3 \quad \cdots \quad I_K]^T \quad (5)$$

$$\mathbf{x} = [E_1 \quad E_2 \quad E_3 \quad \cdots \quad E_N]^T \quad (6)$$

$$\alpha_{k,n} = \frac{K_d \cos(\theta_{k,n}) S_{k,n}}{\|L_n - P_k\|^2} \quad (7)$$

Equation 3 can be solved for a non-negative least squares solution. Let $\mathbf{x}_{nnls}$ be such as solution. How good the solution fits the model is measured by the following RMSE.

$$e_{rmse} = \|\mathbf{Ax}_{\text{nnls}} - \mathbf{b}\| \quad (8)$$

When there are more than one candidate for a light source position, Equation 8 is computed for each candidate so that the one with the minimum error is chosen. As an example, first consider the single light source case. Let $L'_{1,1}$, $L'_{2,1}$, $L'_{3,1}$ be three possible light source positions. By using $L_1 = L'_{m,1}$, m = 1, 2, 3, three RMSE values $e_{rmse,m}$ are obtained. Then, select $L'_{m,1}$ whose $e_{rmse,m}$ is minimum. Extension of this procedure to the case of multiple light sources is straightforward.

STEPS FOR MULTIPLE LIGHT SOURCE POSITION ESTIMATION

In this method, the following requirements are assumed.

(1) The camera parameter and the geometry of objects in the scene are known.
(2) The scene is illuminated by one or more point light sources.
(3) All specular peaks corresponding to light sources are visible in the image used for the estimation.
(4) All cast shadows on the floor due all light sources are visible on the taken image.
(5) The reflectance properties of the spherical and plane objects are known.
(6) The floor plane has a Lambertian surface.
(7) Objects in the scene contain one or more surface discontinuity point, where there is an abrupt change in the surface normal vector.

In the proposed method, steps for light source position estimation can be summarized as the following.

(1) Acquire an image containing specular highlights and a floor plane with shadow casted by a spherical object on it.
(2) Find the specular peak pixels and calculate the directions of light sources (This step is explained in Section 2).
(3) Compute a contour on a ground plane, where a cast shadow of a feature point must lie upon (This step is described in Section 3)
(4) Detect possible corner points on the shadow along the contour obtained from Step 3.
(5) Pair each candidate corner point from Step 5 with the corresponding discontinuity point on the object to form a direction that possibly point to an actual source position.
(6) Find the intersection point of each line from step 5 and the line from step 2 to determine a set of possible location points of the light source. In practice, a pair of lines may not intersect in 3D space. In this case, a middle point on a shortest line connecting the two lines is chosen instead.
(7) Choose light source position candidates as obtained from Step 6 that best fit the ground-plane diffused light Equation 2 in a non-negative least square sense by comparing the corresponding error in Eq. (8).
(8) With a known ground-plane reflection coefficient, light source intensities can be estimated from a chosen solution to Equation 2. Otherwise, the intensities are known up to a scaling factor.

EXPERIMENT

The images used in this experiment are synthetic images at a resolution of 1000×1000 pixels, created using

Table 1. The estimation RMSEs for the single light source case.

Method	RMSE (%)
Without the reflected light fitting step	12.05
With the reflected light fitting step	3.47

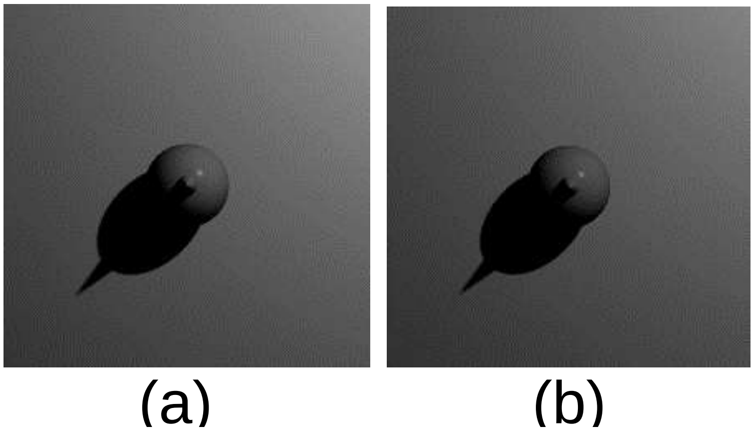


Figure 7. Sample images in the single light source case (a) original image (b) reconstructed image.

Blender 3D software. There were up to three point light sources and a single camera in the scene, all of which were placed above the object and looked down toward it. From this camera position, all specular highlights and cast shadows can be seen in the rendered pictures. The object was a sphere with a small conic shape on top. To evaluate the performance in the presence of soft shadow, nonpoint light sources were simulated by changing the Blender's 'Samples' parameter from the default value of 1 to 8 and 'Soft Size' parameter from the default value of 0.1 to 0.3. The experiments were performed to evaluate the accuracy of the proposed method by computing the Root Means Square Error (RMSE) of the estimated light position. The RMSE percentage was calculated by dividing the obtained RMSE with the actual light source distance from the plane center (the point where the sphere was placed on a ground). Improvement obtained by using the reflected light fitting step (Step 7) was also investigated.

One light source case

There was only one light source in this experiment. Based on results with twenty different light source positions, the RMSE of the distance between the position of the real light source and the estimated position was calculated and used as accuracy measure. The results using the methods with and without reflected light fitting step are shown in Table 1. Figure 7a shows one of the original images, while Figure 7b shows the reconstructed image obtained using the estimated light source position.

Two light sources case

There were two light sources in this experiment. Based on results with ten different two-light source positional settings, the RMSE was calculated as in the previous experiment. The results using the methods with and without the reflected light fitting step are shown in Table 2. Figure 8a shows one of the original images, while Figure 8b shows the reconstructed image obtained using the estimated light source position.

Three light sources case

There were three light sources in this experiment. Based on results with ten cases of three different light source positions, the RMSE of the distance between the position of the real light source and the estimated position was calculated and used as accuracy measure. The results using the methods with and without reflected light fitting step are shown in Table 3. Figure 9a shows one of the original images, while Figure 9b shows the reconstructed image obtained using the estimated light source position.

From the results of the experiments, it is seen that the

Table 2. The estimation RMSEs for the two light sources case.

Method	RMSE of light No. 1 (%)	RMSE of light No. 2 (%)
Without the reflected light fitting step	12.41	12.68
With the reflected light fitting step	4.17	4.20

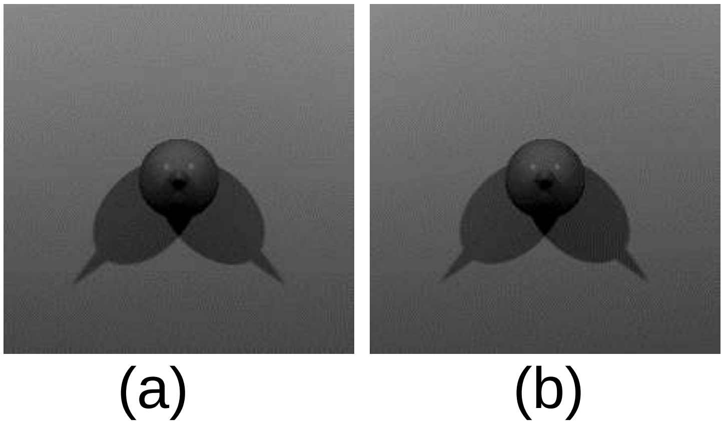


Figure 8. Sample images in the two light sources case: (a) original image (b) reconstructed image.

Table 3. The estimation RMSEs for the three light sources case.

Method	RMSE of light No. 1 (%)	RMSE of light No. 2 (%)	RMSE of light No. 3 (%)
Without the reflected light fitting step	13.33	14.18	14.04
With the reflected light fitting step	7.47	7.69	8.16

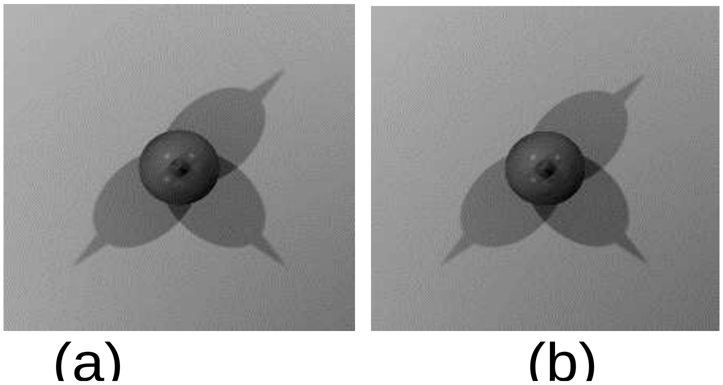


Figure 9. Sample images in the three light sources case: (a) original image and (b) reconstructed image.

estimated light source positions from the first method are less accurate than the second method because of soft shadows with blurry edges. Feature point detection was less accurate in this case. Using the additional reflected light fitting step noticeably improved the estimation accuracy.

Conclusions

In this paper, a method for multiple near point light sources estimation has been described. Our method uses information from specular highlights, feature points in cast shadows and diffuse component on the Lambertian ground plane. Specular highlight is used to estimate a light source direction. Feature points, along with an estimated light source direction, give an estimate of a light source position. For a case where feature points cannot be accurately detected, each possible solution is applied to fit a ground-plane light reflection solution in a non-negative least square sense. A solution candidate with a minimum least square error is chosen as an estimate of light source positions. Experimental results have been reported on the method's effectiveness.

Conflict of interests

The authors have not declared any conflict of interests.

REFERENCES

Bouganis CS, Brookes M (2004). Multiple light source detection. IEEE Trans. Pattern Anal. Mach. Intell. 26(4):509-514.

Cao X, Shah M (2005). Camera Calibration and Light Source Estimation from Images with Shadows. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit. 2:918-923.

Hara K, Nishino K, Ikeuchi K (2005). Light source position and reflectance estimation from a single view without the distant illumination assumption. IEEE Trans. Pattern Anal. Mach. Intell. 27(8):493-505.

Li Y, Lin S, Lu H, Shum H (2003). Multiple-cue illumination estimation in textured scenes. Proc. Ninth Int. Conf. Comput. Vis. 2:1366-1373.

Powell MW, Sarkar S, Goldgof D (2001). A simple Strategy for calibrating the geometry of light sources. IEEE Trans. Pattern Anal. Mach. Intell. 23(9):1022-1027.

Sato I, Sato Y, Ikeuchi K (1999). Illumination distribution from brightness in shadows: Adaptive estimation of illumination distribution with unknown reflectance properties in shadow regions. Int. Conf. Comput. Vis. 2:875-883.

Sato I, Sato Y, Ikeuchi K (2001). Stability issues in recovering illumination distribution from brightness in shadows. IEEE. Conf. Comput. Vis. Pattern Recognit. 2:400-407.

Schnieders D, Wong Kwan-Yee K, Dai Z (2010). Polygonal Light Source Estimation. Computer Vision – ACCV 2009. Lect. Notes Comput. Sci. 5996:966-107.

Takai T, Maki A, Niinuma K, Matsuyama T (2009). Difference sphere: An approach to near light source estimation. Comput. Vis. Image Unders. 113(9):966-978.

Wang Y, Samaras D (2002). Estimation of Multiple Directional Light Sources for Synthesis of Mixed Reality Images. The tenth Pacific Conf. Comput. Graph. Appl. 38-47.

Wei J (2003). Robust recovery of multiple light source based on local light source constant constraint. Pattern Recognit. Lett. 24(1-3):159-172.

Zhang Y, Yan YH (2001). Multiple illumination direction detection with application to image synthesis. IEEE Trans. Pattern Anal. Mach. Intell. 23(8):915-920.

Zhou W, Kambhamettu C (2002). Estimation of illuminant direction and intensity of multiple light sources. Eur. Conf. Comput. Vis. 4(LNCS 2353):206-222.

International Journal of Physical Sciences