BEST: International Journal of Management, Information Technology and Engineering (BEST: IJMITE)
ISSN 2348-0513
Vol. 2, Issue 2, Feb 2014, 31-38

SMART VOTING SYSTEM WITH FACE RECOGNITION

NIRANJAN MALWADE, MAHESH TAVARE, AKSHAY KAMBLE & ANIRUDDHA KAKRAMBE

University of Pune, Maharashtra, India

ABSTRACT

Manual voting system has been deployed for many years in our country. However in many parts of our country people cannot attend the voting because of several reasons. In order to solve these problems there is a need of online election voting system in addition to manual voting system. The long-term goal of this project is to greatly reduce the cost and complexity of running elections and increase the accuracy of results by removing the direct involvement of humans in gathering and counting of votes. The voters will be able to give their votes at any field areas by using the system if they prefer online voting

In this system, the user votes using android application which can be downloaded over internet. The user has to login before giving his vote. Authentication will be done using SMS confirmation and face recognition. Users can view parties, enter votes, can verify vote via SMS or e-mail and can view result after result is declared.

KEYWORDS*:* Smart Voting System, Face Recognition, Android Application, Mobile Voting

I. INTRODUCTION

Now a day's mobile phones are widely used, it is possible the development of applications for worldwide popular participation by digital vote using mobiles. E-Voting system is the application for Android Mobile Operating System Platform. It is application for voting (polling) purpose. Application is based on Client-Server Architecture. At server site, we are storing a database globally. System is divided into three main parts one is Server, second is Java client (Registration Centre) and third is Android client (Voter). In Java client application, Voters can register their name, documents and face image for further authentication. The system can have numbers of Android clients. Each Android client can view candidate information, information about different parties and can submit the vote and also can see newly updated status of election.

Mobile Voting has been used in lots of countries with the development of the E-government technologies in the past years. Generally in these countries the electronic voting is supervised by the presence of the independent electoral authorities. The specific electronic voting machines are used at polling stations for the voting operation.

Manual voting system has been deployed for many years in our country. However in many parts of our country people cannot attend the voting because of several reasons, sometimes people may not be in their own region and due to this fact they cannot do voting. In order to solve these problems there is a need of electronic voting system in addition to manual voting system. After registering to system, the voters will vote from any region by using proposed system from their android device if they are valid voters.

II. PREVIOUS WORK

We introduce *UVote* as a ubiquitous e-voting system that addresses different security aspects of a remote e-voting system. *UVote* leverages redundancy as a key design principle at different components of its architecture, including redundancy at the voting devices, vote verification, and multiple time voting. In *UVote*, a user can vote form

multiple standard devices, such as a laptop or cell phone, in addition to going to the physical voting location. This, along with verification sent to multiple devices, help *UVote* combat malicious software on a certain device. A voter can cast her vote multiple times, reducing the effect of vote-selling and user coercion. Redundancy in the different ways of voting ensures that the system can combat malicious network nodes. This leads to a system where election process errors can be detected before it is over, removing the overhead of repeating it, increasing the users trust in the system, and hence their participation. Further, *UVote* can work as a frontend to many of the current traditional e-voting systems, providing the user with a convenient voting solution with guarantees on security, achieving the best of both worlds. [1].

In a democracy the most important aspect of the democracy is the ability of the people to choose their ruler by vote. The makes the electoral process of utmost importance and increasing its requirement to the strictest levels. With the advent of technology, a number of e-voting systems have come into existence. The term e-voting includes several types of voting including when user is casting the vote and the counting of the votes. This technology includes punch scan, optical scan and specialized voting kiosks. One of the methods is that of online voting using internet. Such type of election system are used by corporations and organizations where the members are in a far off location and is also as a substitute similar to postal ballots for voters at different unreachable locations, for general elections in a country [2].

In [3] it is revealed that there is a raising interest for voting on SMS cell phones, and through social networking tools like Facebook or Twitter. It´s believed that the voting process by cell phones give some decision power to the citizens, which can actuate directly on decisions of their concerns. The voting process also can give ways for numerical information surveillance about social phenomena. For this reason, the following research is being developed with the intention to survey future scenarios which can occur during the international voting process by mobile devices. The experiment has two steps: The first step refers to the voting process with the SMS protocol, using Web 2.0 tools to help on information exchange during the voting process and the second step is the construction of a prototype system which allows users of Android platform smart phones send their votes directly to web services, for its computation, also allowing these users to exchange information by a discussion list, accessible from the given device.

This paper is divided in eight parts: This section describes the intention to create an electronic voting process using mobile telephony, given the emergent opportunity of programmable platform of Google Android smart phones. Section II describes previous jobs about e-Voting systems. Section III describes the first part of the experience: The International Direct Digital Elections (ID2E), with help of Web 2.0 tools to communicate the main theme for elections between voters and interested people. Section IV describes the second part of the experience: The construction of a prototype system for e-Voting, and a discussion list to use from inside Android smart phones, and a Web service for vote tally information. Section V presents the results from the two experiences. Section VI presents the conclusions. Section VII presents the intentions for future work. At the end it is presented the consulted references.

To recognize human faces, the prominent characteristics on the face like eyes, nose and mouth are extracted together with their geometry distribution and the shape of the face. There are differences in shape, size and structure of these organs, so the faces are differ in thousands ways, and we can describe them with the shape and structure of these organs in order to recognize them. These feature points and relative distances between them make some patterns in every input signal. These characteristic features are called eigenfaces in the facial recognition domain (or principal components). Once the boundary of the face is established and feature points are extracted, the eigenface approach can be used to extract features from the face [4].

III. SYSTEM ARCHITECTURE

This system is a web-based system so fundamental features related with web-based technologies such as client-server, database, image procession properties determine the software requirement of the system.

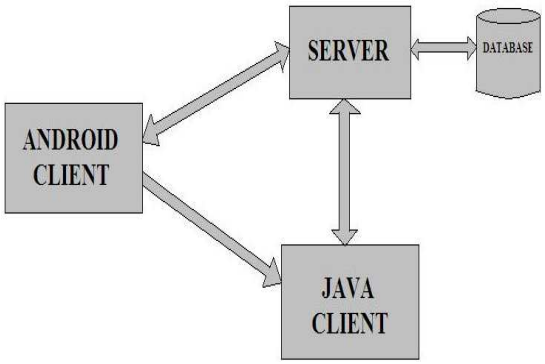

Figure 1: Block Diagram for Smart Voting System

The software product is a standalone system and not a part of a larger system. The system will be made up of two parts. Before the Election Day the application will be used for general purposes such as viewing candidates' profiles and past years' election results. On Election Day another independent Android application will be used for voting operations. This application will be available online for free download on authorized government sites. This application will be installed on voter's Android mobile. The voters cast their votes using this application. The voter is verified using face recognition. These votes are accepted by the system on the server. The Election Commission Authority configures the whole system according to its needs on the server where the system is running.

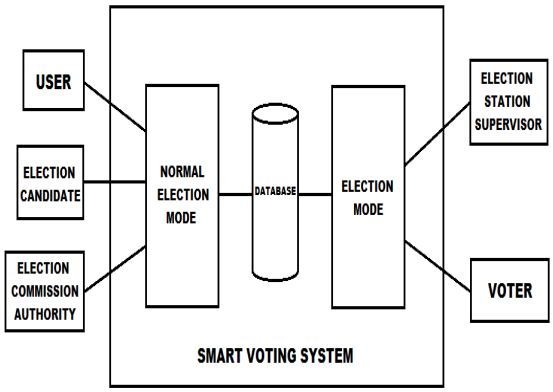

Figure 2: Basic Architecture of Smart Voting System

IV. IMAGE PROCESSING

In this system, face recognition is playing main role for authentication. Live image will be captured by mobile frontal camera. This captured image will be sent to the server for further processing. By using this image, server checks whether user is authorized or not. User is permitted for voting only if he is authorized.

There are many techniques to recognize face by using image processing. We are using some standard algorithms for face recognition. And they are Grayscale, Threshold, Blurring, Scaling, Template generation & matching. These algorithms should process sequentially on image.

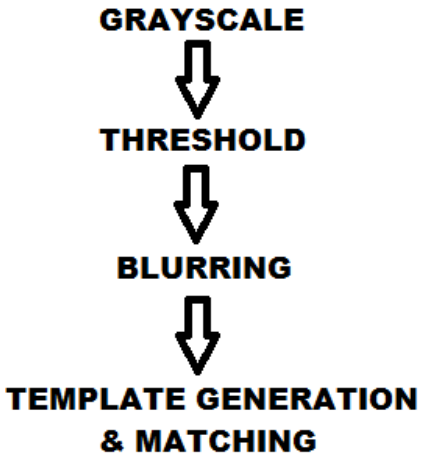

Figure 3: Sequence of Image Processing Algorithms

A. Grayscale Algorithm

Grayscale images have many shades of gray in between. Grayscale images are also called monochromatic, denoting the absence of any chromatic variation (i.e., one color). In this system averaging method is used to grayscale the image.

In averaging method, RGB values are separated from pixel and calculate the average value by using following formula,

GS = (R+G+B) / 3.

Where GS: Average value of Red, Green and Blue.

This GS value is replaced by original pixel value. By doing this we get the image in grayscale format.

B. Threshold Algorithm

Threshold is the simplest method of image segmentation. From a grayscale image, threshold can be used to create binary images.

During the threshold process, individual pixels in an image are marked as "object" pixels if their value is greater than some threshold value (assuming an object to be brighter than the background) and as "background" pixels otherwise. This convention is known as *threshold above*. Variants include *threshold below*, which is opposite of threshold above; *threshold inside*, where a pixel is labeled "object" if its value is between two thresholds; and *threshold outside*, which is the opposite of threshold inside. Typically, an object pixel is given a value of “1” while a background pixel is given a value of “0.” Finally, a binary image is created by coloring each pixel white or black, depending on a pixel's labels.

The key parameter in the threshold process is the choice of the threshold value. Several different methods for choosing a threshold exist; users can manually choose a threshold value, or a threshold algorithm can compute a value automatically, which is known as *automatic threshold*. A simple method would be to choose the mean or median value, the rationale being that if the object pixels are brighter than the background, they should also be brighter than the average.

C. Blurring Algorithm

In image terms blurring means that each pixel in the source image gets spread over and mixed into surrounding pixels. Another way to look at this is that each pixel in the destination image is made up out of a mixture of surrounding pixels from the source image.

Blurring an image reduces the sharpening effect; this makes the detection more accurate. We are doing blur by calculating the average of surrounding 8 pixels that is 3*3 windows. To increase the blur effect we can scan surrounding 5 pixels that is 5*5 windows.

D. Scaling Algorithm

Image scaling is the process of resizing a digital image. Scaling is a non-trivial process that involves a trade-off between efficiency, smoothness and sharpness. Image is scaled into some standard size by using different scaling methods.

E. Template Generation & Matching

Template matching is a technique in digital image processing for finding small parts of an image which match a template image. It can be used in manufacturing as a part of quality control, a way to navigate a mobile robot, or as a way to detect edges in images. Template matching can be subdivided between two approaches: feature-based and template-based matching.

The feature-based approach uses the features of the search and template image, such as edges or corners, as the primary match-measuring metrics to find the best matching location of the template in the source image.

The template-based, or global, approach uses the entire template, with generally a sum-comparing metric (using SAD, SSD, cross-correlation, etc.) that determines the best location by testing all or a sample of the viable test locations within the search image that the template image may match up to.

Improvements can be made to the matching method by using more than one template; these other templates can have different scales and rotations. It is also possible to improve the accuracy of the matching method by hybridizing the feature-based and template-based approaches. Naturally, this requires that the search and template images have features that are apparent enough to support feature matching.

V. UML DIAGRAM

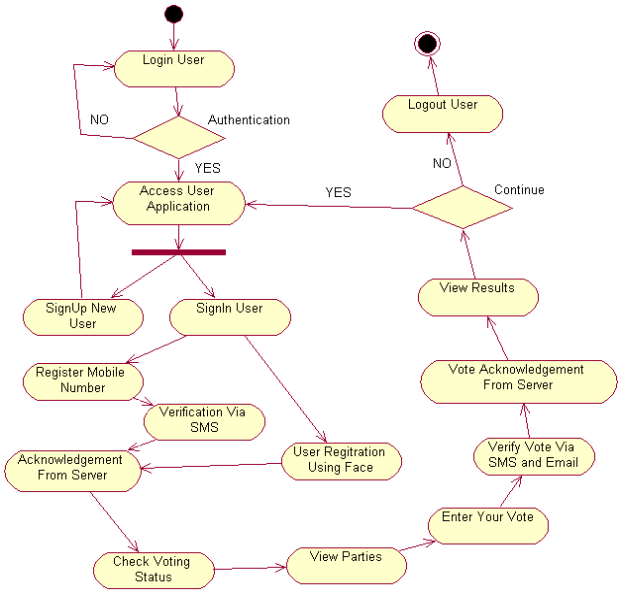

Figure 4: Activity Diagram of Smart Voting System

VI. MATHEMATICAL MODEL

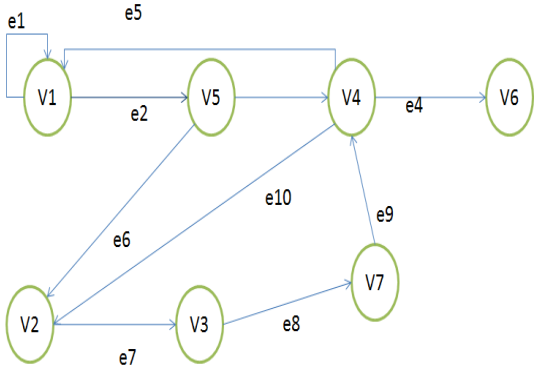

Figure 5: State Diagram of Smart Voting System

Let G be a closed graph that represents our system; such that G = {E, V} where E represents the set of edges; E = {*e0, e1, e2, e3....., e10*} and V is a set of vertices; V = {*v0, v1, v2, v3...., v7*}. In the graphical representation of the system, vertices in the set V represent the modules which are connected through directed edges in the set E representing the input/output of modules. We define the vertices as,

Table 1: Vertex Definition

Vertex	Modules
V1	New user
V2	Voter & Application User
V3	Android Application
V4	Server
V5	Java Client
V6	Database
V7	Vote

Table 2: Edges Definition

Edges	Input/Output
e1	Register for new user
e2	Giving user data to Java client
e3	Fetching Server
e4	Storing Data in Database
e5	Acknowledgement to User
e6	Generate New User
e7	Fetching Application
e8	Entering Vote
e9	Sending Vote to Server
e10	Acknowledgement to User via SMS

Let ***fe*** be a rule of E ***into*** V such that for given edge; it returns vertices**.** ***fe*** (E) |→ V.

Thus, for our system,

fe (*e1*) = {*v1*}.....*v1 is called using e1 for registration of new user.*

fe (*e2*) = {*v5*}.....*user data is passed to v5 using e2 to verify user documents.*

fe (*e3*) = {*v4*}.....*user data is passed to v4 using e3 to fetch server.*

fe (*e4*) = {*v6*}..... *v6 is called using e4 to store data in database..*

fe (*e5*) = {*v1*}.....*v1 is called using e5 for giving acknowledgement to the user.*

fe (*e6*) = {*v2}.....v2 is called using e6 to create new user.*

fe (*e7*) = {*v3}..... v3 is called using e7 to fetch Android application.*

fe (*e8*) = {*v7}.....v7 is called using e8 to enter the vote of voter.*

fe (*e9*) = {*v4}..... user data is passed to v4 using e9 to fetch server.*

fe (*e10*) = {*v2}.....v2 is called using e10 for giving the acknowledgement via SMS to use.*

VII. CONCLUSIONS

This Project focused on the analysis of development of Smart-voting application on an android platform. The usability of this system is very high if it will use in real life election process. It will definitely helpful for the users who wish to vote and the voting process will be made very easy by using this application. However, after having tested the system, in future we tend to add additional functionality of image validation for the security constraint and uniqueness which will provide very strong security for the confidential information about vote.

VIII. REFERENCES

1. Reem Abdelkader, Moustafa Youssef, “UVote: A biquitous E-Voting System”, 2012 Third FTRA International Conference on Mobile, Ubiquitous, and Intelligent Computing.
2. Sanjay Saini, Dr. Joydip Dhar, “An eavesdropping proof secure online voting model”, 2008 International Conference on Computer Science and Software Engineering.
3. Vinícius O. Mirapalheta, Tânia C. D. Bueno, Thiago P. S. Oliveira, “Web 2.0 e-Voting System using Android Platform”,http://www.pcadvisor.co.uk/news/index.cfm?newsid=3213010 Accessed in February, 2010.
4. Sheetal Chaudhary, Rajender Nath, “A Multimodal Biometric Recognition System Based on Fusion of Palmprint, Fingerprint and Face”, 2009 International Conference on Advances in Recent Technologies in Communication and Computing.
5. Anil K. Jain, Arun Ross, and Sharath Pankanti, “Biometrics: A Tool for Information Security”, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 1, NO. 2, JUNE 2006.
6. Ben Adida and Ronald L. Rivest. Scratch & Vote: Self-Contained Paper-Based Cryptographic Voting, ACM Workshop on Privacy in the Electonic Society ACM, 2006.
7. Lazaros Kyrillidis, Sheila Cobourne, Keith Mayes, “Distributed e-Voting using the Smart Card Web Server”, 2012 7th International Conference on Risks and Security of Internet and Systems (CRiSIS).
8. L. Kyrillidis, K. Mayes, and K. Markantonakis, “Web Server on a SIM card”, Lecture Notes in Engineering and Computer Science, vol. 2183, 2010.
9. M. Bishop and D. Wagner, “Risks of e-voting”, Community ACM, vol.50, no. 11, pp. 120–120, 2007.
10. N. K. Sastry, “Verifying Security Properties in Electronic Voting Machines” Ph.D. dissertation, EECS Department, University of California, Berkeley, May 2007. Available: http://www.eecs.berkeley.edu/Pubs/TechRpts/2007/EECS-2007-61.html.

11. S. Campanelli, A. Falleni, F. Martinelli, M. Petrocchi, and A. Vaccarelli, "Mobile implementation and formal verification of an e-voting system", in *Proc. 2008 Third Int. Conf. Internet and Web Applications and Services (ICTW'08)*, Washington, DC, 2008, pp. 476–481, IEEE Computer Society.

12. Qinyuan Feng, Yan Lindsay Sun, "Voting Systems with Trust Mechanisms in Cyberspace: Vulnerabilities and Defenses", IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 22, NO. 12, DECEMBER 2010.