

TEQZONE

2020-2021

Institute Vision

To be a Global Leader in imparting Quality Technical Education to produce Competent, Technically Innovative Engineers imbued with Research Aptitude, Entrepreneurship and Social Responsibility.

Institute Mission

- 1.To nurture the Students with Fundamental Engineering Knowledge enriched with Technical Skills.
- 2.To create Conducive Environment to nurture Innovation and Interdisciplinary Research.
- 3.To develop Professionals through Innovative Pedagogy focusing on Individual Growth, Discipline, Integrity, Ethics and Social Responsibility.
- 4.To foster Industry-Institution Partnerships Leading to Skill Development and Entrepreneurship.

DEPARTMENT VISION

To be a center for academic Excellence in Computer Science and engineering Education, Research and Consultancy

Contributing Effectively to meet industrial and social needs



DEPARTMENT MISSION

- i. To Impart quality technical education with global standards.
- ii. To Provide a platform for harnessing Industry oriented technical skills with inter – disciplinary research awareness.
- iii. To Promote entrepreneurship and leadership qualities imbued with professional ethics



ABOUT DEPART MENT

The Department of Computer Science and Engineering offers 4 year Degree which is established in the year 2007 with intake of 60 seats in CSE. It is approved by AICTE and Affiliated to JNTUA, Anantapur. In 2011 post graduate programe (M.Tech) in Computer Science & Engineering is introduced with an intake 18 seats. An additional intake of 6 seats was incorporated in 2013, total intake of M.tech program reaches to 24 seats. The course is flexible and has been structured to meet the evolving needs of the IT industry. Since the Management of this college includes the highly educated persons, it understands the value of the latest applications. employees or to turn as employers by taking up some entrepreneurial steps.



Program Educational Objectives (PEO)

PEO1: Graduates of the Program will have Strong fundamental knowledge in Computer Science & Engineering, technical competency and problem-solving skills to develop innovative solutions.

PEO2: Graduates of the Program will have Necessary domain knowledge and successful professional career in IT and allied fields of Computer Science & Engineering.

PEO3: Graduates of the Program will have Ability to pursue higher education and Entrepreneurship.

PEO4: Graduates of the Program will have Necessary skills for lifelong learning, teamwork and research to cater for real time needs of industry and society.

Programme Specific Outcomes (PSOs)

PSO1: Apply Software Engineering Principles and Practices to provide software solutions.

PSO2: Design and Develop Network and Mobile based applications.

PSO3: Design innovative algorithms and develop effective code for business applications.

ACKNOWLEDGEMENT

We extend our sincere thanks to

Honorable Chairman

Dr. K .V. Subba Reddy

Secretary & correspondent

Smt. S. VijayaLakshamma

Principal

Dr. L. Thimmaiah,

HOD

Dr. C.Md Gulzar

All our staff members for their humble

Co- operation and involvement in their creation of bytes,

For the year 2020-2021





Message from the Chair man

Its been a real pleasure to know that the Department of CSE is hosting their first ever National Level Technical Symposium “TEQZONE”, AND I’ am glad to hear that it is being organized wholly for the students with guidance of the staff members. Such combined effort is always encouraged and bring out good results.

The Department of computer Science and Engineering has always conducted activities which helps in Development of students into leaders, I hope TEQZONE2020 is a huge success and adds a new star in the history of the department

With Regards

Dr. K .V. Subba Reddy,

Founder – Chairman,

Dr. K.V. Subba Reddy Institute of Technology,

Kurnool- 518218,



Message from the Correspondent

I feel very proud that the Department of CSE is Organizing a national level technical symposium “TEQZONE” on 2020.

The 21st century is advancing rapidly by multipronged scientific inventions and discoveries in that the Computer Science and Engineering is playing the vital role in all scientific developments. The has Com that without Computer Science Engineering nothing is going to move I this universe. In this perspective the contribution the development of society by this department is vital in all sphere of life.

I heartily wish the staff and students of the department in their endeavor to bring in a house magazine which will otherwise contribute to the highest learning of this magnificent engineering.

With Regards

Secretary & correspondent

Smt. S. VijayaLakshamma,

Dr. K.V. Subba Reddy Institute of Technology,

Kurnool- 518218,



Message from the Principal

In the ever changing field of technical education, technology is moving at a very fast pace. What was break through yesterday is obsolete today. This has made it improve that future technocrats must be familiar not only with technical skills but also With the technology of tomorrow . I hope young engineers passing from this instigation will create difference in Indian and global scenario.

I expect my Students to be sincere in their work . They should have never give up attitude and unquenchable thirst of know ledge. I am sure that this magazine will provide platform to students to sharpen their skills.

With Regards

,
Dr . L. Thimmaiah,

Principal

Dr. K.V. Subba Reddy Institute of Technology,

Kurnool- 518218,



Message from the HOD

I wish that this seminar provide an opportunistic forum and vibrant platform for the engineers to share their original research work and practical development experiences and emerging issues.

With Regards

Dr . C. Md Gulzar,

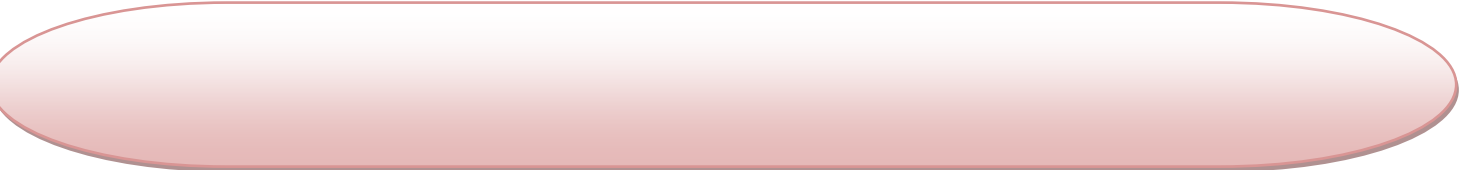
CSE-HOD

Dr. K.V. Subba Reddy Institute of Technology,

Kurnool- 518218,

ABOUT COLLEGE

This institution was established in the year 2007 by Dr. K.V . Subba Reddy ,Chairman, Dr K.V . Subba Reddy ,Chairman, Group of institutions and his wife Smt. S. Vijaya Lakshamma correspondent K.V. Subba Reddy group of institutions with an altruistic motive of providing deserved technical education to the students with humble education background . Dr. K V SRIT is located at Dupadu Village , On highway NH-44 towards Kurnool to Bangalore, in a sprawling area spreading over 40 acres amidst lush green fields the scenic landscape and the serene environment is absolutely conducive for academic pursuits . The institute which is 9km away from Kurnool city can be reached by bus or auto The institute offers 5 UG Programs in CSE, ECE, EEE, ME and CE and five PG Programs in MBA, MTech (ECE), MTech(EEE) MTech (CE) MTech (CSE) The Institution Has registered considerable growth in terms of infrastructure with untiring efforts of management , the departments are accommodated with highly qualified and experienced faculty members. The institute is marching a head by imparting quality technical education with an objective of producing young engineers and managers endowed with dynamic skills and prudence to meet their future challenges . For more details visit WWW.drkvsrit.in.



Student Articles



Detection of Digital Photo image forgery

Digital images can be obtained through a variety of sources including digital cameras. With rapidly increasing functionality and ease of use of image editing software, determining authenticity and identifying forged regions, if any, is becoming crucial for many applications. This paper presents methods for authenticating and identifying forged regions in digital photo images that have been acquired. Our re-examination of some of these recently successful experiments shows that variations in image clarity in the experimental datasets were correlated with authenticity, and may have acted as a confounding factor, artificially improving the results. To determine the extent of this factor's influence on previous results. We demonstrate that a feature derived from Hidden-Markov-Tree-modeling of the digital photo image forgery using wavelet coefficients has the potential to distinguish copies from originals in the new dataset.

INTRODUCTION

Digital image forgery is the process of manipulating photographic images using image-processing tools like digital photo editing software to produce a digital image as evidence to the court; there is a need to identify the authenticity of the image. Digital Image forgery can be classified as the forgery with copy move and without copy move. In case of copy move type, some part of the image is cut and pasted somewhere in the image so that there are no manipulations like rotation, scaling etc. In the other case, due to the above-mentioned types, the data becomes highly correlated. The advent of the modern digital technology has not only brought about the prominent use of digital images in our daily activities but also the ease of creating image forgery

using public accessible and user-friendly image processing tools such as Photoshop. Hence the need for image authenticity assurance and detection of image forgery such as photomontage becomes increasingly acute as digital images take role as news photographs, legal evidence and digital financial document.

A comparative study of the existing algorithms helps to investigate new methods. It opens up new avenues of research. It also helps the real world to overcome the problems being faced due to photomontage and forgery.



Fig. 1: Lenin and Trotsky (left) and the result of photographic tampering (right), Trotsky is missing.

The literature survey has revealed that a substantial amount of work has been done in the field of digital image forgery and forensic science. Various algorithms and mathematical models were developed for detecting digital image forgery and various digital image forgery prevention methodologies, tools and techniques.

Manipulation of early photographic images was not an easy task, requiring a high level of technical expertise and specialized equipment. Alterations had to be made to the negatives, thus, if access could be obtained to the negatives, the authenticity or otherwise of the image could be determined by visual examination.

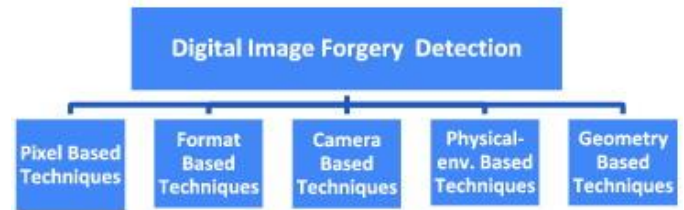
Tampering with photographic images dates back almost to the time when permanent photographic images were first created. One of the earliest instigators of photographic image tampering was Vladimir Ilyich Lenin,

who, for political reasons, instructed that certain individual be removed from photographs (Figure 1)

With the advancement of the digital image processing software and editing tools, a digital image can be easily manipulated. The detection of image manipulation is very important because an image can be used as legal evidence, in forensics investigations, and in many other fields. The pixel-based image forgery detection aims to verify the authenticity of digital images without any prior knowledge of the original image. There are many ways for tampering an image such as splicing or copy-move, resampling an image (resize, rotate, stretch), addition and removal of any object from the image. In this paper we have discussed various pixel-based techniques for image forgery detection, mainly copy-move and splicing techniques.

Passive image forgery detection techniques roughly can be divided into five categories [Citation4] as shown in Figure 1. Pixel-based techniques detect statistical anomalies introduced at the pixel level; format-based techniques leverage the statistical correlations introduced by a specific lossy compression scheme; camera-based techniques exploit artifacts introduced by the camera lens, sensor, or on-chip post-processing; physical environment-based techniques explicitly model and detect anomalies in the three-dimensional interaction between physical objects, light, and the camera; and geometry-based techniques make measurements of objects in the world and their positions relative to the camera.

Figure 1: Digital image forgery detection techniques.



Pixel-based image forgery detection: Pixel-based techniques emphasize on the pixels of the digital image. These techniques are roughly categorized into four types. We are focusing only two types of techniques copy-move and splicing in this paper. This is one of the most common forgery detection techniques. [Figure 2](#) shows categorization of pixel-based forgery detection techniques.

Figure 2: Pixel-based image forgery detection. PCA: principal component analysis; DCT: discrete cosine transform; DWT: discrete wavelet transform; SVD: singular value decomposition; SIFT: scale invariant feature transform; SURF: speeded up robust features. Fridrich et al. [Citation13] proposed a method for detecting copy-move image forgery in 2003. In this method, the image is divided into overlapping blocks (16×16) for feature extraction. Authors have used DCT coefficients for feature extraction. Then, the DCT coefficients of blocks are lexicographically sorted. After lexicographical sorting, similar blocks are detected and forged regions are found. In this paper authors perform robust retouching operations in the image. But authors have not performed any other robustness test.

Popescu et al. [Citation14] proposed a technique for detecting duplicate image regions in 2004. In this paper, authors applied PCA on small fixed-size image

blocks (16×16 , 32×32). They computed the eigenvalues and eigenvectors of each block. After applying lexicographical sorting, the duplicate regions are automatically detected. This algorithm is an efficient and robust technique for detecting a tampered region automatically. The advantage of this algorithm is the ability to detect duplicate region even if the image is compressed or noisy. Kang and Wei [Citation8] proposed the use of SVD to identify the tampered regions in a digital image in 2008. In this paper Authors used SVD for extracting feature vector and dimension reduction. Lexicographical sorting is applied on rows & column vectors and similar blocks are identified to detect forged regions. This algorithm is robust and efficient. Lin et al. [Citation15] proposed a fast copy-move forgery detection technique in 2009. In this paper Authors used PCA for finding features vectors and dimension reduction then Radix sort is applied on feature vectors to detect forgery. This algorithm is efficient and works well in noisy and compressed images. Huang et al. [Citation9] proposed the detection of copy-move forgery in digital images using SIFT algorithm in 2009. In this paper, authors introduced SIFT algorithm using feature matching. The algorithm provides good results even when image is noisy or compressed. Li et al. [Citation10] proposed a sorted neighbourhood approach for detecting duplicate region based on DWT and SVD in 2007. In this paper, authors used DWT and decomposed into four sub-bands. SVD was used in low-frequency sub-bands to reduce dimension representation. Then, they applied lexicographical sorting on singular value vector and the forged region is detected. They tested grey-scale and colour images for detecting duplicate region. This algorithm is robust. Luo et al. [Citation16] proposed a robust detection of region duplication in digital images in 2006. In this

paper, authors divide an image into overlapping blocks and then apply the similarity matching on these blocks. The similarity matching identifies the duplicate regions in the image. This method also works in the JPEG compression, Gaussian blurring, and additive noise.

Zhang et al. [Citation17] proposed a new approach for detecting copy-move forgery detection in digital images in 2008. Authors used DWT and divided low-frequency band into four non-overlapping sub-images and phase correlation is adopted to compute the spatial offset between the copy-move regions. Then, they applied pixel matching for detecting the forged region. This algorithm works well in the highly compressed image. This is a very effective algorithm with lower computational time compared with other algorithms.

Kang et al. [Citation18] proposed copy-move forgery detection in digital image in 2010. Authors divided the image into sub-blocks and used improved SVD. Then, similarity matching is performed on the lexicographically sorted SV vectors and the forged region in the images is detected.

Ghorbani et al. [Citation11] proposed DWT-DCT (QCD)-based copy-move image forgery detection in 2011. Authors used DWT and resolved the image into sub-bands and then performed DCT-QCD (quantization coefficient decomposition) in row vectors to reduce vector length. After lexicographically sorting the row vectors, shift vector is computed. Finally, the shift vector is compared with threshold and the forged region is highlighted. Lin et al. [Citation7] proposed an integrated technique for splicing and copy-move image forgery detection in 2011. First, the authors

converted an image into the YCbCr colour space. For splicing detection, the image is divided into sub-blocks and DCT is used for feature extraction. For copy-move detection, SURF is used. The algorithm works well in both splicing and copy-move image forgery detection. Qu et al. [Citation6] proposed a technique to detect digital image splicing with visual cues in 2009. The authors used a detection window and divided it into nine sub-blocks. VAM (visual attention model) is used to identify a fixation point and then feature extraction for extracting the spliced region in the image. Lin et al. [Citation19] proposed a fast, automatic, and fine-grained tampered JPEG image detection technique using DCT coefficient analysis in 2009. Authors have used DCT coefficient and Bayesian approach for detecting a forged block.

CONCLUSION

In this paper various approaches of pixel-based image forgery detection have been reviewed and discussed. All the methods and approaches discussed in this paper are able to detect forgery. But some algorithms are not effective in terms of detecting actual forged region. On the other hand some algorithms have a very high time complexity. So, there is a need to develop an efficient and accurate image forgery detection algorithm.

M.RAFIYA (18FH1A0510)

III -II YEAR (CSE)

Deploying Wireless Sensor Network on Active Volcano

Augmenting heavy and power-hungry data collection equipment with lighter, smaller wireless sensor network nodes leads to faster, larger deployments. Arrays comprising dozens of wireless sensor nodes are now possible, allowing scientific studies that aren't feasible with traditional instrumentation. Designing sensor networks to support volcanic studies requires addressing the high data rates and high data fidelity these studies demand. The authors sensor-network application for volcanic data collection relies on triggered event detection and reliable data retrieval to meet bandwidth and data-quality demands.

Introduction

Today's typical volcanic data-collection station consists of a group of bulky, heavy, power-hungry components that are difficult to move and require car batteries for power. Remote deployments often require vehicle or helicopter assistance for equipment installation and maintenance. Local storage is also a limiting factor stations typically log data to a Compact Flash card or hard drive, which researchers must periodically retrieve, requiring them to regularly return to each station.

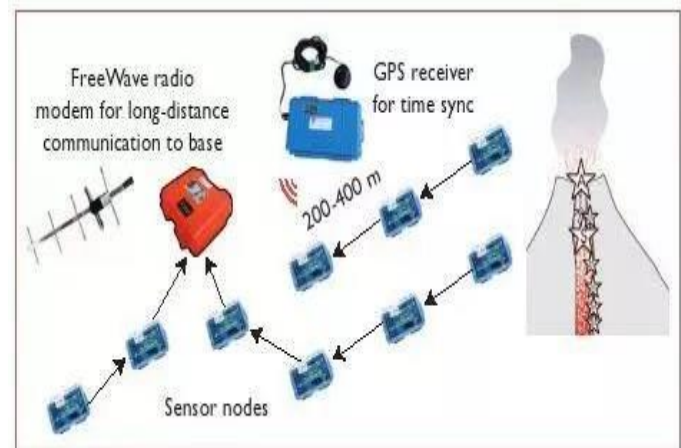
The geophysics community has well established tools and techniques it uses to process signals extracted by volcanic data-collection networks. These analytical methods require that our wireless sensor networks provide data of extremely high fidelity a single missed or corrupted sample can invalidate an entire record. Small differences in sampling rates between two nodes can also frustrate analysis, so samples must be accurately time stamped to allow comparisons between nodes and between networks.

An important feature of volcanic signals is that much of the data analysis focuses on discrete events, such as eruptions, earthquakes, or tremor activity. Although volcanoes differ significantly in the nature of their activity, during the

deployment, many interesting signals spanned less than 60 seconds and occurred several dozen times per day. This let us design the network to capture time-limited events, rather than continuous signals.

Sensor-Network Application Design

Given wireless sensor network nodes current capabilities, we set out to design a data-collection network that would meet the scientific requirements we outlined in the previous section. Before describing our design in detail, let's take a high-level view of our sensor node hardware and overview the networks operation. Figure 1 shows our sensor network architecture.



sensor network architecture

Figure 1. The volcano monitoring sensor-network architecture. The network consists of 16 sensor nodes, each with a microphone and seismometer, collecting seismic and acoustic data on volcanic activity. Nodes relay data via a multichip network to a gateway node connected to a long-distance Free Wave modem, providing radio connectivity with a laptop at the observatory. A GPS receiver is used along with a multichip time-synchronization protocol to establish a network-wide time base.

Network Hardware

The sensor network comprised 16 stations equipped with seismic and acoustic sensors. Each station consisted of a Moteiv Tome Sky wireless sensor network node an 8-dBi

2.4-GHz external omnidirectional antenna, a seismometer, a microphone, and a custom hardware interface board. Each of 14 nodes are fitted with a Geospacer Industrial GS-11 geophone a single-axis seismometer with a corner frequency of 4.5 Hz oriented vertically. The two remaining nodes with triaxial Geospacer Industries GS-1 seismometers with corner frequencies of 1 Hz, yielding separate signals in each of the three axes.

The Tome Sky is a descendant of the University of California, Berkeley's Mica mote,• sensor node. It features a Texas Instruments MSP430 microcontroller, 48 Kbytes of program memory, 10 Kbytes of static RAM, 1 Mbyte of external flash memory, and a 2.4-GHz Chicon CC2420 IEEE 802.15.4 radio. The Tome Sky was designed to run TinyOS,³ and all software development used this environment. The Tome Sky is chosen because the MSP430 microprocessor provides several configurable ports that easily support external devices, and the large amount of flash memory was useful for buffering collected data, as we describe later.

A custom hardware board is built to integrate the Tome Sky with the seism acoustic sensors. The board features up to four Texas Instruments AD7710 analog-to-digital converters (ADCs), providing resolution of up to 24 bits per channel.

The MSP430 microcontroller provides on-board ADCs, but they're unsuitable for our application. First, they provide only 16 bits of resolution, whereas we required at least 20 bits. Second, seism acoustic signals require an aggressive filter centered around 50 Hz. Because implementing such a filter using analog components isn't feasible, it usually approximated digitally, which requires several factors of oversampling. To

perform this filtering, the AD7710 samples at more than 30 kHz, while presenting a programmable output word rate of 100 Hz. The high sample rate and computation that digital filtering requires are best delegated to a specialized device.

A pair of alkaline D cell batteries powered each sensor node our network's remote location made it important to choose batteries maximizing node lifetime while keeping cost and weight low. D cells provided the best combination of low cost and high capacity, and they can power a node for more than a week. Roughly 75 percent of the power each node draws is consumed by the sensor interface board, primarily due to the ADCs high power consumption. The network is monitored and controlled by a laptop base station, located at a makeshift volcano observatory roughly 4 km from the sensor network itself. Free Wave radio modems using 9-dBi directional Yagi antennae were used to establish a long-distance radio link between the sensor network and the observatory.

Typical Network Operation

Each node samples two or four channels of seism acoustic data at 100 Hz, storing the data in local flash memory. Nodes also transmit periodic status messages and perform time synchronization, as described later. When a node detects an interesting event, it routes a message to the base station laptop. If enough nodes report an event within a short time interval, the laptop initiates data collection, which proceeds in a round-robin fashion. The laptop downloads between 30 and 60 seconds of data from each node using a reliable data collection protocol, ensuring that the system retrieves all buffered data from the event. When data collection completes, nodes return to sampling and storing sensor data.

Sensor-Network Device Enclosures and Physical Setup

A single sensor network node, interface board, and battery holder were all housed inside a small weatherproof and watertight Pelican case, as Figure 2 shows.



A two-component station

Figure 2. A two-component station. The blue Pelican case contains the wireless sensor node and hardware interface board. The external antenna is mounted on the PVC pole to reduce ground effects. A microphone is taped to the PVC pole, and a single seismometer is buried nearby.

Environmental connectors are installed through the case, letting cables to be attached to external sensors and antennae without opening the case and disturbing the equipment inside. For working in wet and gritty conditions, these external connectors became a tremendous asset. Installing a station involved covering the Pelican case with rocks to anchor it and shield the contents from direct sunlight.

The antennae are elevated on 1.5-meter lengths of PVC piping to minimize ground effects, which can reduce radio range. We buried the seismometers nearby, but far enough away that they remained undisturbed by any wind-induced shaking of the antenna pole. Typically, we mounted the microphone on the antenna pole and shielded it from the wind and elements with plastic tape. Installation took several minutes per node, and the equipment was sufficiently light and small that an individual could carry six stations in a large pack. The PVC poles were light but bulky and proved the most awkward part of each station to cart around.

Network Location and Topology

We installed our stations in a roughly linear configuration that radiated away from the volcano's vent and produced an aperture of more than three kilometres. We attempted to position the stations as far apart as the radios on each node would allow. Although our antennae could maintain radio links of more than 400 meters, the geography at the deployment site occasionally required installing additional stations to maintain radio connectivity. Other times, we deployed a node expecting it to communicate with an immediate neighbour but later noticed that the node was bypassing its closest companion in favour of a node closer to the base station.

Most nodes communicated with the base station over three or fewer hops, but a few were moving data over as many as six. In addition to the sensor nodes, three Free wave radio modems provided a long-distance, reliable radio link between the sensor network and the observatory laptop. Each Free wave required a car battery for power, recharged by solar panels. A small number of Crossbow MicaZ sensor network node served supporting roles. One interfaced between the network and the Free wave

modem and another was attached to a GPS receiver to provide a global time base.

Design issues of deploying a WSN on the active volcano

Overcoming High Data Rates: Event Detection and Buffering

When designing high-data-rate sensing applications, we must remember an important limitation of current sensor-network nodes: low radio bandwidth. IEEE 802.15.4 radios, such as the Chicon CC2420, have raw data rates of roughly 30 Kbytes per second. However, overheads caused by packet framing, medium access control (MAC), and multichip routing reduce the achievable data rate to less than 10 Kbytes per second, even in a single-hop network. Consequently, nodes can acquire data faster than they can transmit it. Simply logging data to local storage for later retrieval is also infeasible for these applications.

The Tome Skys flash memory fills in roughly 20 minutes when recording two channels of data at 100 Hz. Fortunately, many interesting volcanic events will fit in this buffer. For a typical earthquake or explosion at Retentor, 60 seconds of data from each node is adequate.

Each sensor node stores sampled data in its local flash memory, which we treat as a circular buffer. Each block of data is time stamped using the local node time, which is later mapped to a global network time. Each node runs an event detector on locally sampled data.

Good event-detection algorithms produce high detection rates while maintaining small false-positive rates. The detection algorithms sensitivity links these two metrics a more sensitive detector correctly identifies more events at the expense of producing more false positives. Then

implemented a short-term average/long-term average threshold detector, which computes two exponentially weighted moving averages (EWMAs) with different gain constants. When the ratio between the short-term average and the long-term average exceeds a fixed threshold, the detector fires. The detector threshold lets nodes distinguish between low-amplitude signals, perhaps from distant earthquakes, and high-amplitude signals from nearby volcanic activity. When the event detector on a node fires, it routes a small message to the base-station laptop. If enough nodes report events within a certain time window, the laptop initiates data collection from the entire network (including nodes that didn't report the event).

This global filtering prevents spurious event detections from triggering a data collection cycle. Fetching 60 seconds of data from all 16 nodes in the network takes roughly one hour. Because nodes can only buffer 20 minutes of eruption data locally, each node pauses sampling and reporting events until it has uploaded its data. Given that the latency associated with data collection prevents our network from capturing all events, optimizing the data-collection process is a focus of future work.

Reliable Data Transmission and Time Synchronization

Extracting high-fidelity data from a wireless sensor network is challenging for two primary reasons. First, the radio links are lossy and frequently asymmetrical. Second, the low-cost crystal oscillators on these nodes have low tolerances, causing clock rates to vary across the network. Much prior research has focused on addressing these challenges.

A reliable data-collection protocol was developed, called Fetch, to retrieve buffered data from each node over a multichip

network. Samples are buffered locally in blocks of 256 bytes, then tagged with sequence numbers and time stamps. During transmission, a sensor node fragments each requested block into several chunks, each of which is sent in a single radio message. The base-station laptop retrieves a block by flooding a request to the network using Drip, a variant of the Tinos Trickle6 data-dissemination protocol. The request contains the target node ID, the block sequence number, and a bitmap identifying missing chunks in the block. The target node replies by sending the requested chunks over a multichip path to the base station.

Scientific volcano studies require sampled data to be accurately time stamped; in this case, a global clock accuracy of ten milliseconds was sufficient. The Flooding Time Synchronization Protocol (FTSP) is chosen to establish a global clock across our network. FTSPs published accuracy is very high, and the Tinos code was straightforward to integrate into our application. One of the nodes used a Garmin GPS receiver to map the FTSP global time to GMT. Unfortunately, FTSP occasionally exhibited unexpected behaviour, in which nodes would report inaccurate global times, preventing some data from being correctly time stamped. We are currently developing techniques to correct our data sets time stamps based on the large amount of status messages logged from each node, which provide a mapping from the local clock to the FTSP global time.

Command and Control

A feature missing from most traditional volcanic data-acquisition equipment is real-time network control and monitoring. The long-distance radio link between the observatory and the sensor network lets our laptop monitor and control the networks activity. A Java-based GUI is developed for monitoring the networks behaviour and

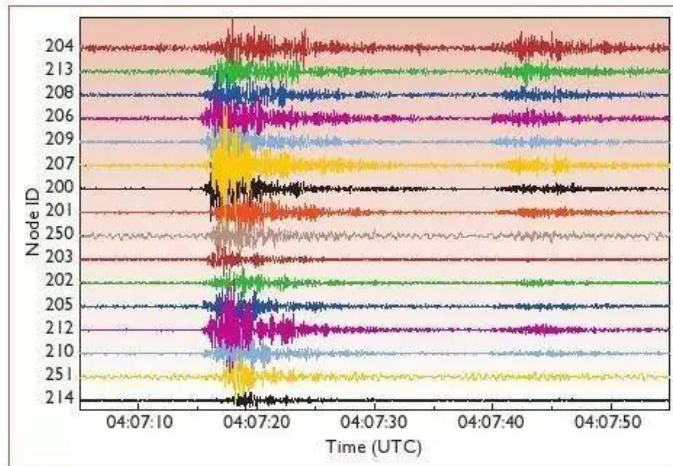
manually setting parameters, such as sampling rates and event-detection thresholds. In addition, the GUI was responsible for controlling data collection following a triggered event, moving significant complexity out of the sensor network. The laptop logged all packets received from the sensor network, facilitating later analysis of the network's operation.

The GUI also displayed a table summarizing network state, based on the periodic status messages that each node transmitted. Each table entry included the node ID; local and global time stamps; various status flags; the amount of locally stored data; depth, parent, and radio link quality in the routing tree; and the nodes temperature and battery voltage. This functionality greatly aided sensor deployment by letting a team member rapidly determine whether a new node had joined the network as well as the quality of its radio connectivity.

Early Results

The sensor network deployed at Vulcan Retentor for more than three weeks, during which time seism acoustic signals from several hundred events were collected. Some early observations during the 19-day deployment, data from the network 61 percent of the time was retrieved. Many short outages occurred because due to the volcanos remote location powering the logging laptop around the clock was often impossible. By far the longest continuous network outage was due to a software component failure, which took the system offline for three days until researchers returned to the deployment site to reprogram nodes manually. Finally, the event-triggered model worked well. During the deployment, the network detected 230 eruptions and other volcanic events, and logged nearly 107 Mbytes of data. Figure 3 shows an example

of a typical earthquake our network recorded.



a volcano tectonic (VT) event
Figure 3. An event captured by our network. The event shown was a volcano tectonic (VT) event and had no interesting acoustic component. The data shown has undergone several rounds of postprocessing, including timing rectification.

Conclusion

By examining the data downloaded from the network, we verified that the local and global event detectors were functioning properly. As we described, we disabled sampling during data collection, implying that the system was unable to record two back-to-back events. In some instances, this meant that a small seismic event would trigger data collection, and we'd miss a large explosion shortly thereafter. We plan to revisit our approach to event detection and data collection to take this into account. Our deployment raises many exciting directions for future work.

ESHA RANI (18FH1A0501)
III-II (CSE)

Imbricate Cryptography

H. ATEEQ AHMED

(ASSISTANT PROFESSOR)

Introduction to Cryptography

Security and privacy are critical for electronic communication and e-business. Network security measures are needed to protect data during its transmission. Cryptography plays a vital role in network security as it allows two parties to exchange sensitive information in a secured manner. The word cryptography means covered writing (covered for crypto and writing for graph). It involves the use of a secret key known only to the participants of the secure communication: If A wants to send a message to B, he encrypts the original message X by the encryption algorithm using the key agreed upon by them. The encrypted message is transmitted through the communication media and the key is transmitted through a secured media like RF cable, fibre, etc. The receiver decrypts the original message from the encrypted message using the same key and the descriptor. A cryptanalyst may try to capture the message and the key. If he fails to do so, the encryption algorithm is successful.

CRYPTOGRAPHY:

When it comes to the security of any important data, the first solution what strikes is encoding the actual data in some form which is private to the user/users only. In technical terms, the simplest solution is Cryptography.

What is cryptography?

Cryptography is the art of achieving security by encoding messages to make them non-understandable to others.

Forming an intelligible message into unintelligible one and then re-transforming that message back to its original form is Cryptography.

There are two types of cryptography:

Asymmetric Cryptography

Symmetric Cryptography

If the sender and the receiver use different keys, it's called asymmetric or multiple-key, public-key encryption.

If the sender and the receiver use the same key, it is called symmetric or single-key, secret-key or conventional encryption.

Encryption: It is the technique of converting original text into coded text using particular key.

Decryption: It is the technique of converting coded text into original text using some key.

Whereas the original text known as plain text and coded text is known as cipher text.

What is Imbricate cryptography?

Imbricate cryptography is a new technique that uses the layered approach designed by us. It is a type of symmetric cryptography in which the key is implanted in the message, so the message cannot be recovered without using the correct key. Here the message and the key are inwardly plaited.

It involves layers of encryption and decryption. Since the key is of variable length of the user's choice, it cannot be found by permutation and combination. Moreover, the output transmitted as a bitmap file perplexes the cracker. Thus, the encrypted file can be sent across the network of interest. Implementation is done by us for the message involving text but the algorithm is extensible to any media. Simplicity, user-orientation and compatibility are the key features of the algorithm.

Notion of encryption: -

The algorithm extends to three layers of encryption, each having its own importance.

Layer-1-

It is called the mapping layer and juggles the cracker by jumbling characters. Here each of the characters is replaced with another one present in the same set. There are two types of sets: repeated characters and non-repeated characters.

English words consist of alphabets, in which the probability of occurrence of some characters such as ,a, ,e, ,i, ,o and ,r is maximum. These characters are called repeated characters. Others are non-repeated characters, i.e., they are repeated occasionally. Each and every character of the source file is mapped with a character present in the same set, thus providing the first layer of crypton. This layer does not include the password or key. Equivalent mapping characters for source file characters are shown in the table. The numbers are also replaced, causing mismatch in numbering also.

Layer-2-

It is called the core-encoding layer as it exploits the bitwise logics and ASCII format to encode each character. Here each character formed by layer-1 is transmuted to an ASCII character, which is not a usual symbol (alphabet, special character or number). The first character of the message obtained by layer-1 is XORed with negated ASCII character of the first character of the password. This process is carried out for the rest of the message. Since the password is of a small length, it is repeatedly applied to the message.

This can be formulated as follows:

$$\text{Char new} = (\text{Char old}) \wedge (\sim \text{key}[i])$$

Layer-3-

It is called the bitmap-conversion layer as it converts ASCII characters into the equivalent binary value and stores the result as a bitmap file. This is done by just obtaining the binary equivalent of the resultant ASCII characters of layer-2 and writing it into a file that is bitmap in nature.

An example for a specific case-

Let us illustrate our technique by the following sets:

Message M = {hello• };

Key K = {Hai• };

Layer-1: -

Table for Mapping	
Source file characters	Equivalent mapping characters
a/e/i/o/s/t/ {repeated}	o/t/s/i/a/e/
b/c/d/e/f/g/h/j/k/l/m/n/ p/q/r/u/v/w/x/y/z/ {non-repeated}	h/t/b/d/g/c/l/n/j/k/m/u/y/p/z/q/v/w/x/p/
0/1/2/3/4/5/6/7/8/9/ {numerals}	4/6/9/7/0/8/1/3/2/5/
Special characters	Same characters

From the table, we can replace

M1={Loji• };

Layer-2:

M2 = M1 \wedge (\sim K);

M2 = {l \wedge ~h, t \wedge ~a, j \wedge ~i, j \wedge ~h, i \wedge ~a};

M2 = {1032~• };

Note that XOR operation is represented by symbol \wedge , 1s complement operation is represented by symbol \sim , and binary values are 11111011, 11101010, 11111100, 11111101 and 11101111. These binary numbers are put in the character form in the output bitmap file finally. An important criterion entailed here is that there is no one-to-one mapping of message characters and password characters. This can be well understood by looking into the above example.

Observe that for the same character j, the resultant codes are not the same. That is, the first ,j is replaced with 11111100 and the second, j is replaced with, 11111101. This shows that the resultant code is unpredictable even for the same set of characters.

Algorithm for encryption-

1. Get the source file and the password (key) from the user.
2. Choose a mapping character for each

character present in the file using the table.

3. Replace the original character with the mapping character. This is the end of layer-1.

4. Using the password (key) received from the user, encode each character of the message with the successive character of the key.

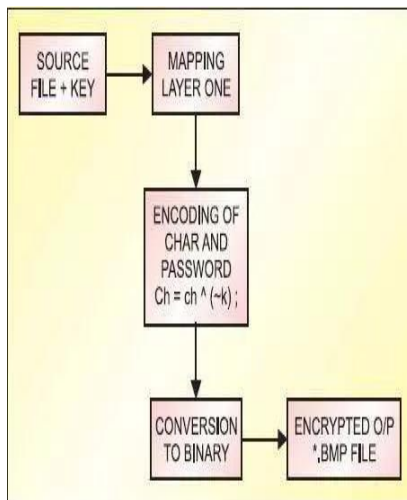
5. The formula for encoding is:

$\text{char new} = (\text{char old}) \text{ XOR } (\sim\text{key}[i]).$

This is the end of layer-2.

6. The resultant character is converted into the binary form. This is the end of layer-3.

7. Write the binary values of the new characters in the output bitmap file.



Imbricate Cryptography – encryption
Notion of decryption-

Decryption is done in the reverse order of encryption. It also has three layers like encryption. Let us go through each layer of the algorithm.

Layer-1-

It is called character-restructuring layer and regroups the bits from the bitmap file to form characters (ASCII). For each 8-bit data found in

the original bitmap file, we find the equivalent ASCII value. Then the character formed by that ASCII is found and noted.

Layer-2-

It is called the core-decoding layer. One of the most fascinating things in XOR logic is that if we apply it twice, the original character can be reproduced. This reveals that the algorithm used in encryption (layer-2) can also be utilized for decryption also. Thus, the same bitwise logic is used here too. Note that only the same key as used in encryption can retrieve the message back.

Layer-3-

It is called the re-mapping layer and works like layer-1 of encryption in the reverse direction. It finds the character in column II of the table and replaces it with the equivalent character present in column I of the table. This completes the decryption process and the output character is written back to the file for decryption.

Note that both the encryption and the decryption processes consist of one layer (layer-1) independent of the key and the other layers are dependent on the key. Thus, now we can know why layer-1 of encryption has not included key.

Algorithm for decryption-

1. Get the bitmap file and the key from the user.

2. Read the binary values from the file and convert back into characters. This is the end of layer-1.

3. From the password (key) received from the user, decode each character with successive character of the key.

4. The formula for encoding is:

$\text{char new} = (\text{char old}) \text{ XOR } (\sim\text{key}[i]);$

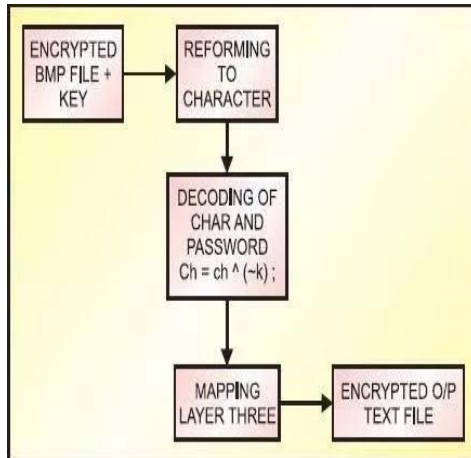
This is the end of layer-2.

5. Choose a mapping character for each

character using the table in the reverse order.

6. Replace the original character with the mapping character. This is the end of layer-3.

7. Write the decrypted character in the output file.



System performance:

Any person who wants to crack this system must:

1. Know that the binary values in the bitmap represent ASCII value of the encrypted character.

2. Read the binary values from the bitmap file and convert them into characters.

3. To break the second layer, find the logic that the key is XORed with the characters. (The key should be known.) But finding the key, which is transmitted over a secured channel, is not possible.

4. Then find the mapping characters to break the first layer. Use of the permutation and combination method for finding the key is impossible. Hence the system performance is good.

Advantages of the system:

1. Confidentiality.

No user can access the message without using the correct key.

2. Simplicity.

The system can be implemented (only for text messaging) through a very simple ,C program given at the end of this article.

3. Security.

The system is secure because the key is sent through a secret medium and the message cannot be recovered without the key.

4. Protection.

It is provided by the key as it controls the access to the message.

5. Incorporated key.

Many cryptography techniques use the key for only access control. Our system integrates the key with the message, so the message can be separated from the key only if the correct key is produced.

Imbricate cryptography involves layers of encryption and decryption. Since the key is of variable length of the user's choice, it cannot be found by permutation and combination. Moreover, the output transmitted as a bitmap file perplexes the cracker.

Thus, the encrypted file can be sent across the network of interest.

Applications:

Identification and Authentication:

Identification and authentication are two widely used applications of imbricate cryptography. Identification is the process of verifying someone's or something's identity. Authentication merely determines whether that person or entity is authorized for whatever is in question. For this purpose, Digital signatures are used.

Certification:

It's a scheme by which trusted agents such as certifying authorities vouch for unknown agents, such as users. The trusted agents issue vouchers called certificates which each have some inherent meaning. Certification technology was developed to make identification and authentication possible on a large scale.

Personal Use:

Privacy is perhaps the most obvious application of imbricate cryptography. Privacy is the state or quality of being secluded from the view and or presence of others. Imbricate cryptography can be used to implement privacy simply by encrypting the information intended to remain

private. In order for someone to read this private data, one must first decrypt it. Note that sometimes information is not supposed to be accessed by anyone, and in these cases, the information may be stored in such a way that reversing the process is virtually impossible.

Passwords:

Passwords are not typically kept on a host or server in plaintext, but are generally encrypted using some sort of hash scheme. In the Windows NT case, all passwords are hashed using the MD4 algorithm, resulting in a 128-bit (16-byte) hash value.

K.VAMSHI (19FH1A0540)

II-II CSE

Snake Robot – The future of Agile motion

Crawling movement as a motive mode seen in nature of some animals such as snakes possesses a specific syntactic and dynamic analysis. A snake robot or serpentine robot designed by inspiration from nature and snakes crawling motion is regarded as a crawling robot. In this article, a snake robot with spiral motion model will be analysed. The purpose of this analysis is to calculate the vertical and tangential forces along snake's body and to determine the parameters affecting these forces. A snake-like device that could slide, glide and slither could open up many applications in exploration, hazardous environments inspection, and medical interventions.

Introduction

Biological snakes are pervasive across the planet; their diverse locomotion modes and Physiology make them supremely adapted for the wide variety of terrains, environments, and climates that they inhabit. A snake-like device that could slide, glide and slither could open up many applications in exploration, hazardous environments, inspection and medical interventions.

One of the fundamental issues is understanding their locomotion. A wheel turns the vehicle moves. A leg pushes the vehicle moves. How a snake moves is not so evident. A worthwhile snake robot has the ability to wriggle into confined areas and traverse terrain that would pose problems for traditional wheeled or legged robots. The design and implementation of a snake robot is the confluence of several technologies: actuation, form and structure, electronics, control, sensing, etcetera.

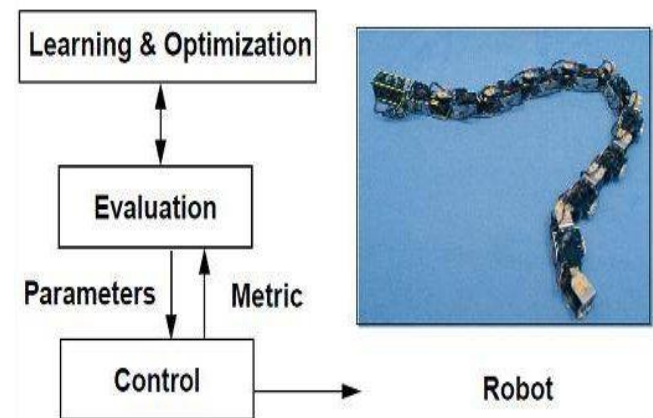
Suggested Read:

Humanoid Robot

Anti-HIV using Nano Robots Why Serpentine Locomotion?

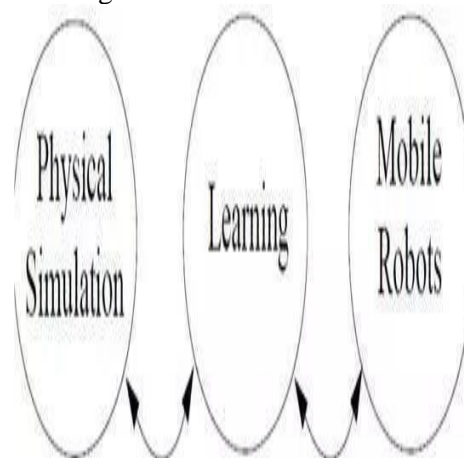
For centuries, people have created a menagerie of machines whose appearance and movement have mirrored animals to an astonishing degree. The general motivations for serpentine

locomotors are environments where traditional machines are precluded due to size or shape. For example, environments include tight spaces, long narrow interior traverses, and travel over loose materials and terrains. Wheels offer smooth and efficient locomotion but often require modifications to terrains for best use. Integration is complicated, even intractable if individual areas are not thought of in the whole.



Flowchart for Snake Robot design Configuration and Design

The challenge of configuration is determining the form of a robot. The challenge of actuation is determining the technology that drives the mechanism. The questions are sometimes mundane but essential to answer: How long should segments be? What angle should they subtend? Are there actuation techniques that can provide smoother curves? Determining both the result and implications of each decision is a challenge.



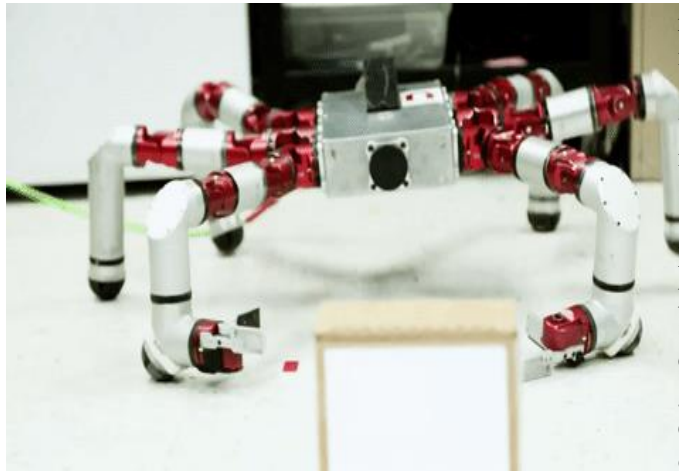
Configuration and Design for snake robot

Infrastructure and Electronics

Supplying and routing power and signals in complex robots is often underestimated as a design task. Serpentine robots must be compact and small to accrue the advantages shown in the previous section. Small size burdens the tasks of wire routing and actuation support.

Control and Sensing

Finally, the greatest challenge: how to learn to control such a device? A larger issue is determining the process, method and framework to achieve this.



Carnegie Mellon's snake robot PC-TechCrunch Advantages of Snake Robots

Stability: Unless a serpentine robot purposefully slithers off a cliff, it can't fall over. In contrast, stability is of great concern to wheeled and legged machines in rough terrain; they can fall over. Terrain contacts in vehicles form a constellation of points on the terrain; if the centre of gravity moves beyond the bounds of the convex polygon formed by these points, it tips over. In a serpentine robot, the potential energy remains low in most situations; therefore, there are few concerns for stability and no need for the support polygons formed by wheel or leg contact points.

Terrain ability: Terrain ability is the ability of a vehicle to traverse rough terrain. Terrain roughness is often measured by scale of features, power spectral density, distribution of obstacles

such as rocks and geographic forms or even its fractal dimension. A serpentine mechanism holds the promise of climbing heights many times its own girth; this feature can enable passage through terrain that would encumber or defeat similarly scaled wheeled and legged machines.

Traction: Traction is the force that can be applied to propel a vehicle. Traction is usually limited to the product of the vehicle weight and the coefficient of friction. The distribution of the snake mass over such a large area, in comparison to mass equivalent legged or wheeled vehicles, results in forces that can be below the thresholds of the plastic deformation of the soil. In comparison, load concentration resulting from most wheels or leg designs results in soil work. Because of the large contact area, serpentine vehicles may result in little or no soil work. Limbless locomotion may prove superior in marginal or soft terrains where ploughing and shearing actions restrict wheel mobility.

Efficiency: Snakes achieve efficiencies and performance equivalent to biomechanisms of similar scale and mass. Reasons include reduced costs associated with less lifting of the centre of gravity as compared to legged animals, elimination of the acceleration or deceleration of limbs, and low cost for body support. The answer is that energy losses in snakes include greater frictional losses into the ground, lateral accelerations of the body that do not contribute to forward motion, and the cost of body support for partial body elevations during movement.

Size: Depending on the mechanism design, the small frontal area of snake mechanisms allows penetration of smaller cross-sectional areas than mass-equivalent legged or wheeled vehicles. If the volume of a snake, a cylindrical form, is kept the same and the diameter is reduced by half, the length becomes four times greater. Cross-sectional area for mechanisms of similar density and mass may result in very long vehicles.

Redundancy: Candidate configurations for serpentine robots may employ many simple motion actuators in sequence. During operation,

the loss of short segments would still permit mobility and manoeuvrability

Disadvantages of Snake Robots

Payload: Much locomotion has to do with work; the transport of materials from one place to another. There is no integral platform for attaching payloads.

Degrees of Freedom: To subtend the various curves needed for locomotion requires a larger number of actuators than most wheeled or legged vehicles. The number of DOFs in vehicles can range from two up to eighteen and even more for some walkers. A large number of DOFs may introduce reliability problems; numbers of units have a higher chance of having any unit fail.

Applications

Exploration: In unpredictable environments, there are zones of uncertainty and footing is insecure or unknown. A snake-like device can distribute its mass over a large area for support so that even if footing gives way, self-support between secure points enables continued operation. Such environments include planetary surfaces and extreme terrains with loose rubble and inclines near the slope of repose.

Inspection: Many inspection techniques in industry and medicine rely on fixed-base mechanisms such as borescopes, videoscopes, and fiberscopes. These devices are primarily used to inspect cavities that cannot be seen directly by the eye. Inspection applications include airline engine maintenance, quality control in manufacturing, and process monitoring and inspection in utilities and chemical plants. Simple direct-view borescopes have proven useful, but articulated self-advancing devices forming and following complex paths could open many more applications.

Medical: Snake-like devices have received attention as a potential medical technology. Minimally-invasive surgery reduces or eliminates the need to cut open large sections of skin and tissue. It is currently estimated that 35% of the 21,000,000 surgeries performed each year could be done with minimally invasive techniques]. There could be dramatic reductions

in hospital stays, patient suffering, and costs. Laparoscopic devices, which are rigid tools inserted into the abdominal wall, and Endoscopic devices are used in these types of surgical procedures.

Hazardous Environments: Human activity is precluded in many areas where there are extremes of radiation, temperature, chemical toxicity, pressure or structural weakness. However, it is often necessary to explore and survey these areas to ensure safety and ascertain status. A variety of small tracked or wheeled machines have been constructed for such applications, but these have limitations in their ability to traverse and manoeuvre through hazardous terrain.

A serpentine mechanism could fare better due to the advantages cited earlier. Other dangerous areas include those following disasters such as earthquakes, Explosions, cave-ins, hurricanes, fires etcetera. The search for survivors and removal of material is often thwarted by loose rubble that might be penetrable by a snake. Outfitted with sensors such as ammonia or pyroelectric IR detectors, a snake-like mechanism would enable sensing of humans in the rubble. These are applications that would eliminate life-endangering alternatives such as using heavy construction equipment to move loose material from accident sites.

Conclusion

The basic purpose of the article was to introduce a new and upcoming subsection of hyper-redundant robots which are finding various uses in all fields. The future of snake robots with the amount of research and development being done is very bright. This article is trying to enlist the various types of snake robots emerging and the innumerable ways of achieving redundancy.

A.FASI AHMED(19FH1A0530)
II-II CSE

Cloud Computing vs. Distributed Computing: Know the Differences

A.E RAJU (ASSISTANT PROFESSOR)

In the last several decades, there have been tremendous improvements in computers and computer network technologies. With the emergence of the Internet, computers and their networking have exhibited tremendous development, such as today's theme – distributed computing and cloud computing.

The terms Distributed Systems and Cloud Computing Systems relate to distinct entities, although the principle behind both is the same. To better grasp the ideas for each of them, a strong understanding of the distributed systems and knowledge of how they vary from the centralized computer is essential.

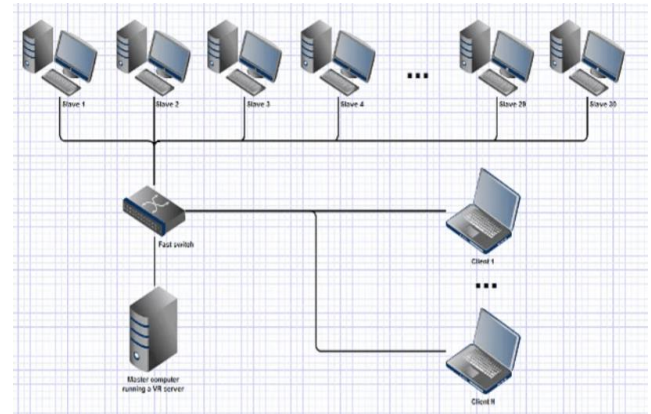
You may observe the use of cloud computing in most organizations nowadays, either directly or indirectly. For example, if we use Google or Amazon's services, we immediately access the resources housed in Google or Amazon's Cloud environment. Twitter is another instance of our tweets in the Twitter cloud. Faster data processing and computer networking may be seen as necessary for these new computing technologies to develop. Read more about the architecture of cloud computing.

What is Distributed Computing?

When numerous autonomous devices connect via a central network to achieve a shared objective, distributed computing, distributed computing solves a difficulty using distributed autonomous machines and communicating with each other over a network. This is a computer method that can interact and solve one single problem on

numerous machines.

Distributed computing is a much faster way to do computing activities than utilizing one computer. Some distributed computing features divide a single job amongst machines to simultaneously carry out the work, calling remotes and calling remotes for distributed calculations.



Need for Distributed Computing

Centralized computer systems, such as IBM for decades, mainframes have been available in technical calculations. One central computer controls all peripherals in centralized computing and conducts complicated calculations. However, centralized computer systems have been inefficient and costly to handle enormous amounts of transaction data and serve tons of online users simultaneously. This cleared the door to the commercial exploitation of similar technologies in the cloud and distributed computing.

Distributed Computing System Examples

World Wide Web

Hadoop's Distributed File System (HDFS)

ATM

Facebook

Google Indexing Server

Google Web Server

Cloud Network Systems

Google Bots

The system is dispersed for a regular user,

and the system is connected to multiple nodes that execute the assigned computer duties. The system is distributed as a single system. Consider from the user's perspective the Google web server. Users are satisfied that the Google Web Server is a single system where they need to login and look for the needed phrase when submitting a search query.

It is a Distributed Computer technique underneath which Google builds and distributes multiple servers in different geographical areas, providing the search results in seconds or milliseconds.

Benefits of Distributed Computing

Compared to a centralized computer, distributed computing systems provide a superior price/performance rate, as adding microprocessors is economic rather than mainframes.

The processing capacity of distributed computing systems is higher than centralized systems. Distributed computer systems enable increased expansion to add software and computing capacity as and when business demands increase.

What is Cloud Computing?

Cloud computing is a service that is provided to a network computer. For example, 10,000 people might process SETI data on their PCs via a screensaver over a dedicated computing network. And cloud computing might be when one million Apple customers keep all their MP3s on iCloud instead of PCs.

In cloud computing, IT resources and services like servers, storage, databases, networks, analytics, software, etc. are provided over the Internet. It is a computer technology that offers its users/customers host services through the Internet.

Cloud computing delivers services, including hardware, software, and Internet

networking resources. Cloud computer features include pooled computer resources, on-demand service, per-use payments, service providers' services, etc. Read more about Cloud computing and the different types of services in cloud computing here.

It is divided into 4 separate kinds, for example.

Private Cloud

Public Cloud

Hybrid Cloud

Community Cloud

Examples of Cloud computing

YouTube is the most acceptable cloud storage example hosting millions of video files uploaded, streamed, and downloaded.

Picasa and Flickr are hosting millions of digital photos that enable their users to build online photo albums by uploading images to servers of their services.

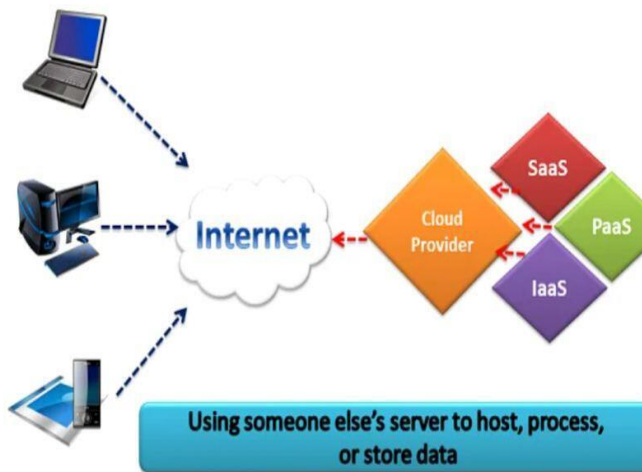
Google Docs is another type of cloud computing that enables users to hook up their server presentations, text documents, and slides. Google Docs allows users to change and publish other people's work for viewing or changing.

Benefits of Cloud computing:

Cloud computing has numerous advantages and advantages, but we are the most relevant:

Cloud computing makes it the most excellent resource for connection via companies' cloud offerings at a reasonable cost for organizations.

In place of conventional or orthodox use of e-mails and file-sharing, companies can make use of their cloud solutions to exchange information with workers.



Distributed Computing Vs. Cloud Computing

DISTRIBUTED COMPUTING vs **CLOUD COMPUTING**

Distributed computing is solving a difficulty using distributed autonomous machines and communicating with each other over a network. In cloud computing, IT resources and services like servers, storage, databases, networks, analytics, software, etc. are provided over the Internet.

In basic distributed computing, a computing method may be used to communicate and solve one single problem by numerous machines. A basic cloud computing technology may provide host services to your users/customers over the Internet.

It is divided into three main types: distributed computing systems, distributed systems, and distributed power systems.

It is divided into 4 kinds such Public, Private, Hybrid, and Community Cloud. It has a varied ranking.

The distributed computers have numerous advantages, such as flexibility, dependability, increased performance, etc.

Cloud computing has numerous advantages, including cost efficiency, flexibility, and reliability, scaling economies, world market access, etc.

Distributed computing helps more quickly

than utilizing a single computer when it takes much time to complete the computing activities. Cloud computing delivers services, including hardware, software, and Internet networking resources.

The objective of distributed computing is to distribute and complete a single job amongst several computers fast through cooperation between machines. Internet Pay-per-Use Computing Services is available on request from the cloud computing service providers.

Some distributed computing features spread a single job amongst the machines to simultaneously advance the work, remote procedure calls, and the remote method invocation for distributed calculations.

Some of the features of cloud computing include pooled computer resources, on-demand service, pay per usage, service providers, etc.

Some cloud computing disadvantages include the possibility of a node failure, and sluggish connectivity might lead to problems. Some cloud computing disadvantages may include less control, particularly public clouds, limitations on existing services, and cloud security.

Conclusion:

The main difference between cloud and distributed computing is that cloud computing provides the software, hardware with complete infrastructure over the internet while distributed computing is dividing tasks into different computing machines that are connected through a network. Are you looking for some help with your assignments? If yes, you can contact My Assignment Lab.

STUDENT ARTS GALLERY



**Art by
P. Sruthi (18FH1A0503)
III -I SEM (CSE DEPT)**



Art by
HARSHITHA (19FH1A0513)
II-I SEM (CSE DEPT)