



A University of Sussex PhD thesis

Available online via Sussex Research Online:

<http://sro.sussex.ac.uk/>

This thesis is protected by copyright which belongs to the author.

This thesis cannot be reproduced or quoted extensively from without first obtaining permission in writing from the Author

The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the Author

When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given

Please visit Sussex Research Online for more information and further details

Video Big Data: An Agile Architecture for Systematic Exploration and Analytics

Submitted for the Degree of Doctor of Philosophy,
Department of Engineering and Design,
University of Sussex, December 2016

Soladoye Oyebowale Ajiboye

Dedication

Dedicated to the memory of my father and mentor, late Snr. Apostle Titus Olukolade Ajiboye and, to the joy of life with Olufunmilayo, Riversofjoy, and David.

Acknowledgement

I am grateful to many people who have been instrumental to the success of my research, leading to the completion of this thesis. First, I thank God for everything.

Special thanks to my supervisors Dr. Philip Birch and Dr. Rupert Young, for their advice, support and guidance since the first day I showed up at their doors – I remember asking Dr. Birch: “so what do I need to start” He answered: “*just turn up with a **notepad**, and a **pen** - you’ll need it*” ...and ***I truly needed it***. Your dedication and positive attitude towards your students is commendable and worth emulating – thanks to both of you.

Special thanks to Professor Chris Chatwin, whom I consider as my ‘ultimate’ supervisor due to the massive support he awarded me. His doors were always open to discuss, advise, and guide - I will always cherish your kindness and selflessness.

The journey would be more difficult without the seasoned insights and comments of the annual reviewers – Dr. William Wang, Professor Peter Cheng and the staff at the school office, in particular Mr. Luke Scott. It also has been a pleasant experience meeting fellow researchers - Lasisi Lawal, Ini Ituen, Babatunde Olawale, and my office mate Auday Al-Mayyahi.

I owe a great deal of thanks to Professor John Carroll, Professor Jackie Cassell, and Dr. Natalia Beloff for ‘initiating’ me into the research world. Particularly Professor Carroll for the unprecedented mentorship towards my professional progression.

Most importantly, I could not have completed this research without the support, and encouragement of my family and friends. My late father and mother, who inspired me to “always be my best” - they taught me that family comes first. My wife Olufunmilayo, and our 2 sons – Riversofjoy and David, who are always there for me during the eventful days and the sleepless nights while I complete this work – **thank you**.

University of Sussex

Soladoye Oyebowale Ajiboye

Submitted for the degree of Doctor of Philosophy

Video Big Data: an Agile Architecture for Systematic Exploration and Analytics

Summary

Video is currently at the forefront of most business and natural environments. In surveillance, it is the most important technology as surveillance systems reveal information and patterns for solving many security problems including crime prevention. This research investigates technologies that currently drive video surveillance systems with a view to optimization and automated decision support.

The investigation reveals some features and properties that can be optimised to improve performance and derive further benefits from surveillance systems. These aspects include system-wide architecture, meta-data generation, meta-data persistence, object identification, object tagging, object tracking, search and querying sub-systems. The current less-than-optimum performance is attributable to many factors, which include massive volume, variety, and velocity (the speed at which streaming video transmit to storage) of video data in surveillance systems.

Research contributions are 2-fold. First, we propose a system-wide architecture for designing and implementing surveillance systems, based on the authors' system architecture for generating meta-data. Secondly, we design a simulation model of a multi-view surveillance system from which the researchers generate simulated video streams in large volumes. From each video sequence in the model, the authors extract meta-data and apply a novel algorithm for predicting the location of identifiable objects across a well-connected camera cluster.

This research provide evidence that independent surveillance systems (for example, security cameras) can be unified across a geographical location such as a smart city, where each network is administratively owned and managed independently. Our investigation involved 2 experiments - first, the implementation of a web-based solution where we developed a directory service for managing, cataloguing, and persisting metadata generated by the surveillance networks. The second experiment focused on the set up, configuration and the architecture of the surveillance system. These experiments involved the investigation and demonstration of 3 loosely coupled service-oriented APIs – these services provided the capability to generate the query-able metadata.

The results of our investigations provided answers to our research questions - the main question being “to what degree of accuracy can we predict the location of an object in a connected surveillance network”. Our experiment also provided evidence in support of our hypothesis – “it is feasible to ‘explore’ unified surveillance data generated from independent surveillance networks”.

Table of Contents

| | |
|--|------|
| Table of Contents..... | v |
| Table of Figures | x |
| List of Tables..... | xii |
| List of Publications..... | xiii |
| Acronyms | xiv |
| 1. Introduction..... | 1 |
| 1.1. Overview | 2 |
| 1.2. Research Background and Relevance..... | 4 |
| 1.3. The Research Gap..... | 6 |
| 1.3.1. Areas Needing Further Research | 7 |
| 1.4. Research Context and Scope | 8 |
| 1.5. Research Questions..... | 9 |
| 1.6. Overall Research Objectives and Goals..... | 9 |
| 1.7. Research Contributions and Achievements | 10 |
| 1.8. Thesis-Specific Terminologies..... | 12 |
| 1.9. Thesis Structure | 13 |
| 2. Literature Review..... | 15 |
| 2.1. Overview | 16 |
| 2.2. The Evolution of Video Surveillance | 16 |
| 2.3. Enterprise Video Surveillance Architecture..... | 18 |
| 2.3.1. Metadata Architecture | 19 |
| 2.3.2. Video Metadata Standards..... | 21 |
| 2.4. Self Awareness and Autonomous Systems | 22 |
| 2.4.1. Self-Awareness in Camera Networks | 25 |
| 2.4.2. Self-Awareness in Name Server | 26 |
| 2.5. Topology Learning and Auto-Interactivity..... | 27 |
| 2.5.1. Graph Theory Definitions and Concepts | 27 |
| 2.5.2. Network Centrality..... | 31 |
| 2.5.3. Closeness Centrality | 32 |
| 2.5.4. Degree centrality:..... | 33 |
| 2.5.5. Eigenvector centrality: | 33 |

| | | |
|--------|---|----|
| 2.6. | Resource Directory and Service Discovery | 35 |
| 2.6.1. | Centralised Client/Server Architecture | 36 |
| 2.6.2. | Peer-to-Peer Architecture | 38 |
| 2.7. | Object Detection and Identification..... | 40 |
| 2.7.1. | Object Identification and Classification..... | 41 |
| 2.7.2. | Autonomous Multi-Camera Coordination..... | 41 |
| 2.7.3. | Head Count and Distance Estimation..... | 43 |
| 2.7.4. | Visual RGB and Depth Data Approaches | 45 |
| 2.8. | Person Re-Identification..... | 46 |
| 2.9. | Storage Solution for Surveillance Systems | 49 |
| 2.10. | Web Services..... | 50 |
| 2.11. | Conclusion of Literature Review | 53 |
| 3. | The Fused Video Surveillance System Architecture..... | 56 |
| 3.1. | Overview | 57 |
| 3.2. | Current Systems | 58 |
| 3.3. | Design Goals | 61 |
| 3.4. | Design Considerations | 63 |
| 3.5. | The Overall System Architecture of the FVSA | 64 |
| 3.5.1. | Overview of the FVSA..... | 65 |
| 3.5.2. | The Camera | 65 |
| 3.5.3. | Intelligent Network Video Recorders (i-NVR)..... | 66 |
| 3.5.4. | Analytics Server..... | 67 |
| 3.5.5. | Video Storage..... | 67 |
| 3.5.6. | Web services | 68 |
| 3.5.7. | Directory Server | 69 |
| 3.5.8. | The User..... | 69 |
| 3.6. | Hierarchies, System Scope and Visibility..... | 69 |
| 3.6.1. | Authorisation and Resource Visibility | 71 |
| 3.7. | Application..... | 72 |
| 3.8. | Chapter Conclusion | 73 |
| 4. | The Experiment - Design and Implementation Strategy | 75 |
| 4.1. | Overview | 76 |
| 4.2. | Design Strategy | 77 |
| 4.3. | The CamNet Configuration | 80 |

| | | |
|--------|---|-----|
| 4.3.1. | The Camera | 80 |
| 4.3.2. | The Camera Object..... | 81 |
| 4.3.3. | The Camera Set up | 82 |
| 4.3.4. | Camera Configuration | 83 |
| 4.3.5. | Camera Registration with the MDS..... | 84 |
| 4.3.6. | The Metadata..... | 85 |
| 4.3.7. | The Metadata Server (MDS) | 86 |
| 4.3.8. | MDS Configuration | 87 |
| 4.3.9. | MDS Registration with the CRD | 89 |
| 4.4. | The City Resource Directory Server (CRD) | 91 |
| 4.5. | CRD Set up and Configuration | 91 |
| 4.6. | System Hierarchies and Scopes | 92 |
| 4.7. | The User..... | 94 |
| 4.7.1. | Security: Roles, Responsibilities and Permissions | 96 |
| 4.8. | Chapter Conclusion | 97 |
| 5. | An Implementation of the CamNet | 98 |
| 5.1. | Overview | 99 |
| 5.2. | Simulation Strategy..... | 100 |
| 5.2.1. | Experiment Set up..... | 102 |
| 5.2.2. | Assumptions and Considerations..... | 102 |
| 5.2.3. | Person and Journey Simulation | 103 |
| 5.3. | The Simulation Model of a CamNet | 105 |
| 5.4. | Unsupervised Topology Learning..... | 106 |
| 5.4.1. | Person Re-Identification | 107 |
| 5.4.2. | Algorithms 5A: Re-ID ranking | 108 |
| 5.4.3. | Formulation of Tracking | 109 |
| 5.4.4. | The Fused Video Surveillance Network (FVSN) | 110 |
| 5.4.5. | The Routing Protocol and System Query..... | 111 |
| 5.5. | Problem Formulation | 113 |
| 5.6. | Predicting the Location of a Person | 114 |
| 5.6.1. | Algorithms 5B: Predicting the Location of a Person..... | 114 |
| 5.7. | Results | 115 |
| 5.7.1. | The Impact of travel period on person re-id..... | 116 |
| 5.7.2. | The Impact of Camera Density..... | 117 |

| | |
|--|-----|
| 5.7.3. The Impact of Crowd Density..... | 118 |
| 5.7.4. Uniqueness of Persons | 118 |
| 5.8. Accuracy of Predicting a Person's Location | 119 |
| 5.9. Validation and Testing..... | 120 |
| 5.9.1. Comparison of Predicted Location with God's Eye | 121 |
| 5.9.2. Comparison with the state of the art..... | 122 |
| 5.10. Chapter Conclusion | 123 |
| 6. A Scalable Resource Directory for the Globally Unified Video Surveillance Network | 125 |
| 6.1. Overview | 126 |
| 6.2. System Architecture | 127 |
| 6.2.1. Global System Overview..... | 127 |
| 6.2.2. The City Resource Directory Server (CRD) | 128 |
| 6.2.3. Implementation of the DHT..... | 129 |
| 6.2.4. The Gateway Resource Directory Server (GWRD) | 131 |
| 6.2.5. Overview of the Country Surveillance System | 132 |
| 6.3. The System Overview of a City | 133 |
| 6.4. CamNet Registration | 134 |
| 6.4.1. Registration Process | 134 |
| 6.4.2. Algorithms 6A: CamNet to CRD Registration..... | 134 |
| 6.4.3. Algorithms 6B: CamNet to GWRD Registration | 135 |
| 6.5. Experimentation and Evaluation | 136 |
| 6.5.1. Experiment Setup..... | 136 |
| 6.5.2. Simulation..... | 138 |
| 6.6. Results | 139 |
| 6.6.1. Flat and Error-Free Surveillance Directory | 139 |
| 6.6.2. The Effects of Failed Nodes on the Flat Architecture | 140 |
| 6.6.3. The Effect of GWRD on System Failure | 142 |
| 6.6.4. The CoSS Topology with Self Management | 143 |
| 6.7. Chapter Conclusion | 144 |
| 7. Conclusion and Future Directions..... | 146 |
| 7.1. Overview | 147 |
| 7.2. Contributions of the Thesis | 147 |
| 7.3. Limitations and Challenges..... | 149 |

| | |
|--|-----|
| 7.4. Ethics and Privacy | 150 |
| 7.5. Future Work..... | 150 |
| 7.5.1. Standardisation and Terminologies | 150 |
| 7.5.2. The Scope of the Solution | 151 |
| 7.5.3. Testing | 151 |
| 7.5.4. Security and Accessibility | 152 |
| 7.6. Final Notes | 152 |
| 8. References | 154 |
| Appendix A: BPMN Notations Used in this thesis | 176 |
| Appendix B: Database Schemas | 177 |
| Appendix C: SOAP XML Message (Metadata Object) | 178 |

Table of Figures

| | |
|---|----|
| Figure 1: A smart video surveillance suite – proposed in [22] | 6 |
| Figure 2: The research’s smart video surveillance architecture (CamNet & CiSS)..... | 12 |
| Figure 3: Landmarks in technological advancement of video surveillance architecture. | 17 |
| Figure 4: IBM Smart Surveillance Analytics (SSA) [21] | 19 |
| Figure 5: Metadata knowledge-adding service architecture [17]..... | 20 |
| Figure 6: Continuous cycle of the 5 key elements of self-reconfiguration in smart camera network..... | 26 |
| Figure 7: A basic weighted digraph..... | 29 |
| Figure 8: Example of a computing network graph - solid links as downloads, and uploads dotted | 31 |
| Figure 9: Directed structural diagram for people estimation [107]..... | 45 |
| Figure 10: Components of Kinect, an indoor 3D mapping system [111] | 46 |
| Figure 11: Overview of Re-ID process..... | 48 |
| Figure 12: Process flow for streaming video in: (a) current systems (b) the FVSA. | 60 |
| Figure 13: Topology of the video surveillance system in a city - A conceptual police view..... | 62 |
| Figure 14: The high level conceptual model of the FVSA..... | 64 |
| Figure 15: Global and local scope of the MDS | 71 |
| Figure 16 Layered architecture of the FVSA showing its relevance to other IoT compatible architecture [135] [136] [132]..... | 73 |
| Figure 17: A Simplified high-level architecture of the research experiment. | 78 |
| Figure 18: Architecture of the CamNet..... | 80 |
| Figure 19: Camera registration form. | 83 |
| Figure 20: Sequence diagram of the camera registration process | 84 |
| Figure 21: Sample SOAP XML object generated in this experiment. | 88 |
| Figure 22: Webform for configuring the MDS. | 90 |
| Figure 23: Architecture of the CRD | 91 |

| | |
|--|-----|
| Figure 24: Webform for configuring the CRD | 92 |
| Figure 25: The surveillance network in a country is organized into scopes emulating the hierarchies in the country's geographic hierarchies. | 96 |
| Figure 26: The simulation grid in Matlab | 101 |
| Figure 27: The database table that represents objects travelling across the network..... | 104 |
| Figure 28: Matlab plot of a CamNet comprising 12 cameras. | 106 |
| Figure 29: Re-ID process in this research..... | 108 |
| Figure 30: The computed FVSN based on Table 15(b)..... | 111 |
| Figure 31: Period varying match rate..... | 117 |
| Figure 32: matching rate under varying camera density. | 117 |
| Figure 33: matching rate under varying crowd density. | 118 |
| Figure 34: impact of person uniqueness on match rate | 119 |
| Figure 35: Comparison of predicted location of a person with their real location..... | 121 |
| Figure 36: The high-level view of the globally connected surveillance system. | 127 |
| Figure 37: Architecture of the CRD | 129 |
| Figure 38: A system overview of the CoSS..... | 132 |
| Figure 39: City connectivity through DHTs. | 133 |
| Figure 40: The system overview of the CiSS | 133 |
| Figure 41: The Setup showing the research's arrangement of the CRDs in a country..... | 138 |
| Figure 42: CamNet registration success rate remains at 100% despite variation in the number of CRDs and the number of registrations..... | 140 |
| Figure 43: System failures due to failed nodes..... | 141 |
| Figure 44: Success and failure with the client-server CoSS..... | 142 |
| Figure 45: Post optimisation result showing the percentage success rate of the CamNet registration attempts with a 10% unreachable CRDs | 143 |

List of Tables

| | |
|--|-----|
| Table 1: research questions | 9 |
| Table 2: Comparing concepts in computational self-awareness and human self-awareness [39] | 24 |
| Table 3: Cross examining the dimensions of system adaptation [48]. | 25 |
| Table 4: The routing table in node a, derived from the network depicted in Figure 8..... | 30 |
| Table 5: Person Re-identification methods [112] | 47 |
| Table 6: REST and SOAP web services compared | 53 |
| Table 7: Visibility and authorization of system services (in Figure 15). | 72 |
| Table 8: Database properties of the Camera..... | 81 |
| Table 9: Properties of the metadata | 85 |
| Table 10: Database properties of the MDS..... | 87 |
| Table 11: User table properties | 95 |
| Table 12: User role table properties | 95 |
| Table 13: System requirement – experiment 1 | 102 |
| Table 14: Experimental parameters – The CamNet | 105 |
| Table 15: (a) Adjacency matrix generated by the objects identified across the graph in Figure 28. (b) Sparse form of the matrix in (15a)..... | 110 |
| Table 16: CamNet’s simulation parameters..... | 113 |
| Table 17: The degree of accuracy achievable over time..... | 119 |
| Table 18: FVSA Ranking Compared with the state of the art – available at the SSIG website. | 122 |
| Table 19: An example DHT table on a CRD | 130 |

List of Publications

- Ajiboye, Sola O. & Birch, Philip M. & Chatwin, Christopher R. & Young, Rupert. "Hierarchical video surveillance architecture: a chassis for video big data analytics and exploration", Proceedings of SPIE Vol. 9407, 94070K (2015).
- Ajiboye, Sola O. & Chatwin, Chris & Birch, Phil M & Young, Rupert. (2017). "A Directory Service for City Video Surveillance Systems". Information Technologies, Systems and Networks Conference, 17-18th October 2017, Chisinau, Moldova
- Ajiboye, Sola O. & Birch, Philip M. & Chatwin, Christopher R. & Young, Rupert. "A Scalable Resource Directory for the Globally Unified Video Surveillance Network". Submitted.
- Ajiboye, Sola O. & Birch, Philip M. & Chatwin, Christopher R. & Young, Rupert. "Predicting the Location of a Person in Non-Overlapping Camera Surveillance Networks", in preparation.

Acronyms

| | |
|--------|--|
| ARDA | Advanced Research and Development Activity |
| BPMN | Business Process Model and Notation |
| CamNet | Camera Network - an independent surveillance system, which belong to the same owner. |
| CiSS | City Surveillance System – a network comprising the CamNets in the same city |
| CoSS | Country Surveillance System – a network of CRDs within the same country |
| CVR | Cloud Video Recorders |
| FOV | Field of View of a camera |
| FVSA | Fused Video Surveillance Architecture |
| FVSN | Fused Video Surveillance Network |
| GUIDN | Globally Unique Identity Number |
| GWRD | Gateway Resource Directory |
| IaaS | Infrastructure as a Service |
| IARPA | Intelligence Advanced Research Projects Activity |
| IoT | Internet of Things |
| JSON | JavaScript Object Notation |
| MDS | Metadata Server |
| MOT | Multiple Object Tracking |
| NVR | Network Video Recorders |
| OASIS | Organization for the Advancement of Structured Information Standards |
| OAuth | Open Authorisation - an open protocol to allow secure authorisation in a simple and standard method from web, mobile and desktop applications. |
| Re-ID | Person Re-Identification across a camera network |
| REST | Representational State Transfer |
| RGB-D | Red, Green, Blue plus Depth |
| SA, SE | Self-Awareness, Self-Expression |
| SOA | Service Oriented Architecture |
| SOAP | Simple Object Access Protocol |
| TA | Topology-aware Algorithm |
| VACE | Video Analysis Content Extraction |
| VEML | Video Event Markup Language |
| VERL | Video Event Representation Language |

1. Introduction

Learn from yesterday, live for today, hope for tomorrow. The important thing is not to stop questioning.

- Albert Einstein

1.1. Overview

The increasing expectancy of the enablement of the IoT is generating a new paradigm to designing networks of smart objects. That is, any device capable of connecting to the Internet such as computers, phones, sensors including surveillance cameras, actuators, smart homes, and smart transport systems. The number of deployed smart objects is fast increasing. It was estimated at 8.7 billion in 2012 and has been projected to be in excess of 25 billion by the year 2020 [1]. The IoT paradigm supports the notion of connectivity among the billion smart objects on the Internet to support their awareness of one another - it also provides capability for knowledge and information exchange. This presents an opportunity to engage, integrate, reuse, acquire, and to provide services/resources to any smart object with system privileges, irrespective of their geographical location [2] [3].

In other words, this conceptually implies that the smart objects around the world can be conceptually described as a globally unified system. This claim would be more socio-economically attractive, if each device on the Internet has a unique identity, and can be tracked down to a local membership of public hierarchies similar to the geopolitical categories as in: the street, city, country, and the continent – that is, an identifiable member of the global digital community (or a **digital citizen**). This is the fundamental motivation of this research - to investigate the feasibility of a 'globally' unified platform that supports the systematic access to any 'known' video surveillance network with a view to identification, processing, and analytics of the surveillance information.

Video-based systems and services have been one of the most implemented in the last decade since video is these days at the forefront of most life domains including security, entertainment, media, communication, commerce/e-Commerce, domestic, education, manufacturing, technology, and healthcare [4] [5]. In fact, cameras and their networks are increasingly playing vital roles in intelligent, ubiquitous and smart systems where they are being applied to detect, acquire, process and report events in various life support situations. Intelligent camera networks have been introduced to support environments and systems such as autopilot cars, unmanned air vehicles, surveillance, health monitoring, environmental monitoring, and smart cities.

For example, the Microsoft Kinect-like camera technology has been proposed for health monitoring for the fall-prone patients such as the elderly. Such camera is capable of extracting the 3-D body joint coordinates at low cost and without the need for body markers since it captures the RGB properties and the depth information of the image [6] [7]. These video data are collectively being generated at massive rates such that they are fast becoming a significant proportion of the global data [8] [9] [10]. To bring this to perspective, as at 2010, Youtube claimed that about 2 billion videos were watched daily and up to 24 hours of video content were uploaded every minute [11] [12]. More so, in 2012, Cisco systems reported that in 2012, video encoded data was about 53% of all data across global IP networks [13]. And by 2015, video data is about 55% of the total mobile traffic [14] - ***Video data is 'Big data'***.

The massive sets of generated data (including video data) are widely called 'big data' and have been loosely defined as "*datasets whose size is beyond the ability of typical database software tools to capture, store, manage, and analyze*" [15]. Although the

exact quantification and definition for 'big data' is still evolving, all major players in the industry share similar views about the immense benefits and challenges it would unravel.

1.2. Research Background and Relevance

There is increasing reliance on video surveillance systems for systematic derivation, analysis and interpretation of the data needed for business intelligence, predicting, planning, evaluating and implementing public safety and services. This is evident from the massive number of surveillance cameras deployed across public locations. In Britain alone, the British Security Industry Association (BSIA) estimated there are over 4.2 million surveillance video cameras in public domain [16] and it was reported that the size of video data generated by military unmanned aerial vehicles is in the petabyte range [17].

Under the data protection act [18], the owners of video surveillance systems have a responsibility to **protect** the identity and privacy of the people recorded by their camera – especially if the camera is likely to record events that are of personal or sensitive nature such as one installed in residential areas. The owners also have a sense of ownership hence the need to **secure** the surveillance system since the data hold information about their personal (or business) operations and security. Such information in the wrong hands could threaten their lifestyle, or damage their reputation, and lead to loss of their competitive advantage, in the case of business owners.

To protect and secure the surveillance data from external and/or unwanted access, the data must be processed and preserved with access control measures. This is usually achieved in corporate environments, by physically locking the storage location of the surveillance data, and employing a security officer, who manually inspects the video data for events and/or trends. Additionally, the owners of video surveillance system could authorise other personnel to access the surveillance data only *on-an-as-needed-basis*, such that everyone else is denied access. Incidentally, only 1.5% of the cameras in Britain is state-owned, and accessible to the public safety departments [16].

With some effort, as we proposed in this research, the current level of advancement in video technology could support surveillance system owners to grant controlled access to public safety departments. Since in the first instance, Internet Protocol cameras are already available and affordable [19], it is therefore reasonable to expect that a tangible proportion of the current and future surveillance cameras are capable of capturing digital data. Invariably data from digital cameras could be accessed over communication networks such as the Internet.

In fact, research has proposed the implementation of centralised repositories for video-encoded data originating from multiple sources. An example of such research is by Giovanni Borga et al. in 2011 [20]. They proposed a solution for integrating data arriving from various locations, on a real time basis. They identified sources of video data that included satellite imaging systems, telecommunications and territorial innovative monitoring networks, sensor networks, Internet and mobile monitoring devices.

Another advancement that supports our claim is the smart surveillance architecture implemented in [21], which is capable querying video metadata. Among other achievements, which we discussed further in chapter 2, the solution involved approaches to achieving real-time alerts, events statistics, events detection, object tracking, colour classification, and face tracking. However, the deployment is targeted at specific environments such as retails, and security. The operation of the solution is depicted in Figure 1, which shows that the video data is analysed to generate metadata on detected events. The metadata is persisted in a database, from which analytics servers source data to achieve human activated operations – the work is published in [21] and [22].

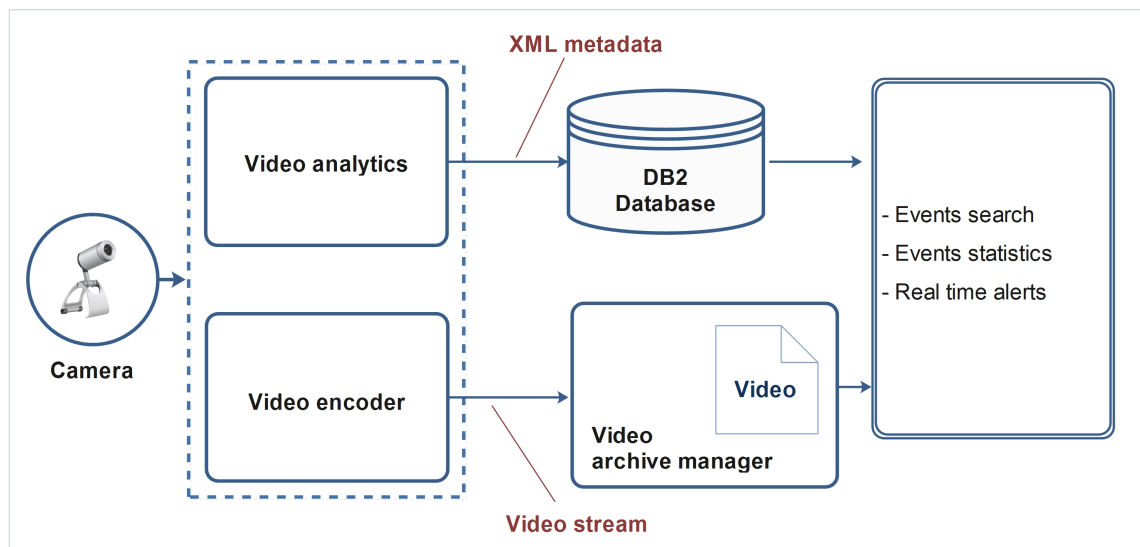


Figure 1: A smart video surveillance suite – proposed in [22]

1.3. The Research Gap

This research conducted a systematic literature review, which is presented in chapter 2, to establish the state of advancement in video surveillance technologies. The review provided evidence that surveillance systems (or cameras) are available in large

numbers in the public domain. They are however currently not utilised to their full potential because the relevant technology is still advancing towards maturity. In essence, the review revealed that video surveillance is an active research area but there are still open problems, which calls for improvement.

1.3.1. Areas Needing Further Research

Despite research and technology advancement, it was noted that most video surveillance systems have not been optimised to achieve the following concepts that are of interest in this research:

- Capability to interact with other 'trusted' surveillance systems in the city.
- Capability to generate and persist metadata for public-level analytics.
- Capability to apply a level of authorisation and authentication on the surveillance system to prevent fraudulent and/or unwanted access.
- Capability to function as an IoT component, such as the smart city component.
- Capability to detect and persist surveillance alerts until reviewed by a human.
- Capability to generate reports, statistical information, and patterns of events.
- Capability to provide limited access to query the surveillance data without allowing full access to the video stream.
- Capability to capture digital data (by the cameras). In many old buildings, the installed cameras are not capable of recording digital data - they are only capable of recording analogue video.

Considering the requirements mentioned above, and based on the knowledge that only a few (if any) of the currently deployed surveillance systems is accessible to the public safety departments, this research explores the notion that:

1. The accuracy of the results obtained from surveillance infrastructure would improve greatly if privately owned surveillance systems contribute to the data used by the public safety departments.
2. Independent owners of surveillance networks would exchange surveillance data with the city, if the data shared will not break the requirements of data protection. That is, they can share only the metadata from their surveillance network without sharing the real video streams.

1.4. Research Context and Scope

The context of this thesis is within the video data generated for surveillance purposes. It focuses chiefly on three features of surveillance systems: (i) unification of independent surveillance systems – metadata, (ii) public-level querying of surveillance data, (iii) event prediction based on pattern recognition of past events. This research does not involve other usage/applications of video technologies such as social networking, health, telecommunications, and entertainment – although the researchers are aware the concepts of this research adaptable for such.

1.5. Research Questions

Following the technological shortcomings that were highlighted in section 1.3 and the evidence of research shortage in chapter 2, the research questions in Table 1 were identified.

Table 1: research questions

| # | Research Question | Developmental Questions |
|-----|---|---|
| RQ1 | Is it technically achievable to analyse and explore a video surveillance system without the full system access to the surveillance network cameras and data? | Can we design surveillance systems with a view to exchanging information across independent networks? |
| RQ2 | Can we query surveillance networks that belong to multiple independent administrative owners? | What system parameters are needed to query a massive surveillance data that have been sourced from multiple networks? |
| RQ3 | How can we develop a globally connected surveillance system that unifies the several independent surveillance networks/cameras while preserving the full system privileges of the network owners? | (a) What are the requirements to unifying independent digital systems and what are the security implications? (b) Can the architecture represent the geo/political structure of the world? |
| RQ4 | To what degree of accuracy can we systematically and analytically predict the location of an object such as a person, across a well-connected camera cluster in a smart city? | How can we suggest the destination of a person in a city, based on places they have been located? |

1.6. Overall Research Objectives and Goals

The overall objectives of this research are in the proposal of scientifically proven and demonstrable solutions to the research questions that were identified in section 1.5.

The following objectives are covered in this thesis:

3. To identify the current state of technology in video surveillance systems and automated surveillance system in general. We achieve this objective by

conducting a literature search and systematic review of the technologies involved in video surveillance systems.

4. To propose and validate system architecture that is capable of processing data from multiple video surveillance networks, as a unified system.
5. To propose a solution for an enterprise video surveillance network.
6. To test and validate our hypothesis that it – “is possible to ‘open’ the data generated by an individually owned surveillance system to the safety department for querying”.

Intelligent video big data architecture such as the proposed FVSA could improve the benefit derived from combination of cameras and their data stores. It suggest the presentation of the surveillance systems in a country as a national surveillance grid, which is composed of independent networks such as a building surveillance systems, street surveillance systems, city surveillance systems. Access to a specific level of the system will depend on user permissions and roles. To achieve this, each subsystem such as a building system would be an independent system but capable of interacting with other subsystems through a defined interface.

1.7. Research Contributions and Achievements

This research involves the set up, configuration and implementation, testing and validation of simulation and modelling of the proposed solution for achieving the research objectives that were identified in section 1.6. The unique contribution of this research includes the following:

- An implementation and demonstration of some of the research algorithms.

- A demonstration of the proposed surveillance network that is capable of managing metadata generated from an enterprise surveillance system. This was achieved in chapter 5 in an experiment, which systematically tracked the location of objects at various locations on the topology-aware network.
- A demonstration of the operations of a city Resource Directory (CRD) that is capable of administering all ‘known’ video surveillance networks (or cameras) within a smart city. This was also demonstrated in chapter 6.
- A demonstration of the design approach to develop a global directory server - the collection of multiple CRDs, as presented in chapter 6.
- A simulation model of the ‘**lean**’ version of the multi-view surveillance system from which video metadata were generated in large volumes.

In practice, the FVSA comprises of a hardware framework that is supported by a multi-layer abstraction software interface. It presents video surveillance systems as an adapted computational grid of intelligent services, which are integration-enabled to communicate with other compatible systems in the Internet of Things (IoT) – this is described in chapter 3. But as mentioned above, this research involved a simulation experiment involving the implement of a lean but demonstrable model of the FVSA.

Figure 2 depicts the high-level architecture in the experiment, showing the flow of metadata from the independent CamNets to the city’s CRD. The components presented in the CamNet shows the high-level process/information flow within the CamNet. It shows that the metadata server (MDS) persists the generated metadata locally but also sends it to the CRD, where the unified metadata is used for city-level

analytics. The detailed design considerations, operations and configuration strategies employed in the experiments are detailed in chapters 4 and 5.

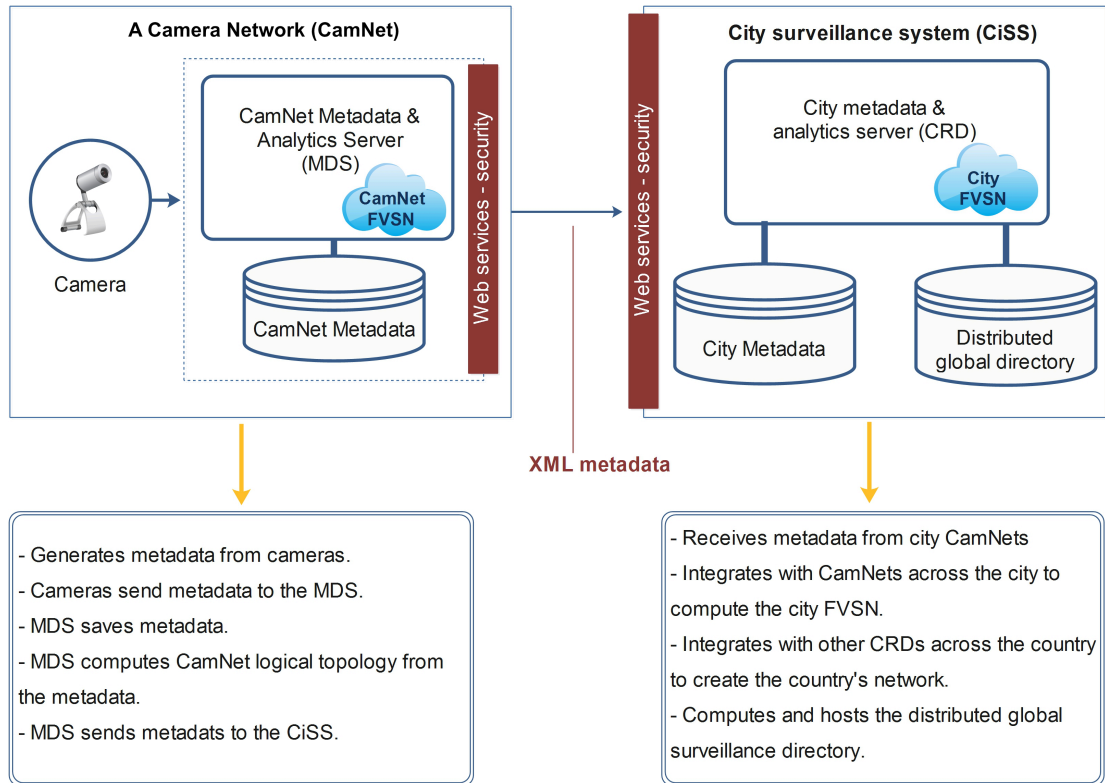


Figure 2: The research's smart video surveillance architecture (CamNet & CiSS)

1.8. Thesis-Specific Terminologies

Throughout this thesis, various acronyms and terminologies were used to identify system components, processes and applications. We refer to a camera or a 'group of cameras', used to capture surveillance data by an individual or organization as a '**camera network**' or **CamNet** for short. The rest of this thesis will refer to both a camera and a 'camera network' as a '**CamNet**' – a CamNet comprise of one or more cameras. For example, a one-camera CamNet is depicted in Figure 2.

We refer to a collection of CamNets in the same city as the City Surveillance System (**CiSS**), where all the CamNets in a city are registered in the City Resource Directory (**CRD**). Furthermore, the network of all the CRDs in a country is referred to as the Country Surveillance System (**CoSS**) – we refer to the resource directory that manages all the CRDs in a country as the Country Gateway Resource Directory (**GWRD**). The GWRD is a CRD configured as the server for all the CRDs in a country.

The metadata generated on the CamNet is saved within the CamNet. However, the CamNet also sends the metadata to the CiSS. Both the CamNet and CiSS are capable of computing a logical network based on the metadata. The logical network is a graph where the camera whose object is identified becomes the network node. If the same object is re-identified, then there is a link between the two nodes (or cameras). The logical network is referred to as the Fused Video Surveillance Network (**FVSN**).

1.9. Thesis Structure

The rest of this document is organised as follows - chapter 2 is a literature review of the existing work in the area of video surveillance systems, focusing on system architecture, metadata generation and future trend in video surveillance. Chapter 3 is a discussion of the system architecture in this research – it is where the components of the experimental network, metadata subsystem, storage infrastructure and their connectivity were discussed. Chapter 4 presents the internal configuration and implementation strategy of the experiments. These include the simulation model of the CamNet, the city resource directories, and the corresponding TA algorithms.

Chapter 5 is an experiment to investigate the appropriate internal configuration of the city CamNets and how they collectively 'interface' with each other, to form the citywide network surveillance directory server, the CRD. At the end of this chapter, it was noted that the ability to explore the CRD can provide evidence to support investigations and business decisions. Ultimately it can reduce both financial and time costs that are associated with processing video surveillance signals by human analysts.

Chapter 6 involve the unification of a collection of the CRDs in a country to produce the country's surveillance system - a technological view of the global surveillance system. This chapter investigated, presented and described the hierarchies, responsibilities, scopes, and system boundaries in the global surveillance system. Chapter 7 is the last chapter and conclusion of this thesis – it includes a discussion of the relevance of this research including technical and security implications of a global surveillance systems, ethical and privacy concerns. Also in this last chapter, we discussed the strength of our approach, the overall direction of this research, future work and the envisaged direction of video surveillance systems.

2. Literature Review

*Knowledge is power only if man knows what facts not to
bother with*

- Robert Staughton Lynd

2.1. Overview

This chapter reviews and examines the current level of innovation and the future trends of technological advancement in video big data, which influence aspects of video surveillance systems such as video surveillance systems/architectures, video metadata generation/extraction, video object detection/identification, video object tagging and video object tracking. The review articulates questions, findings and projects requiring further work with a view to evaluating key elements of video surveillance systems requiring optimisation and potential 'call to action' as basis for the author's research.

2.2. The Evolution of Video Surveillance

An important aspect of a video surveillance system is the architecture, which comprise the interactions of the system components, such as cameras, monitoring screens, video data storage, and analytics of the video data. As depicted in Figure 3, the approach for configuring these components has advanced from the initial one-to-one camera-to-screen connection into better system where multiple cameras are connected to a Network Video Recorders (NVR). NVRs are devices that connect multiple cameras, serving as single output hub for all connected cameras. Lately the architecture has been presented as an enterprise software solution such as the IBM SSA described in section 1.2).

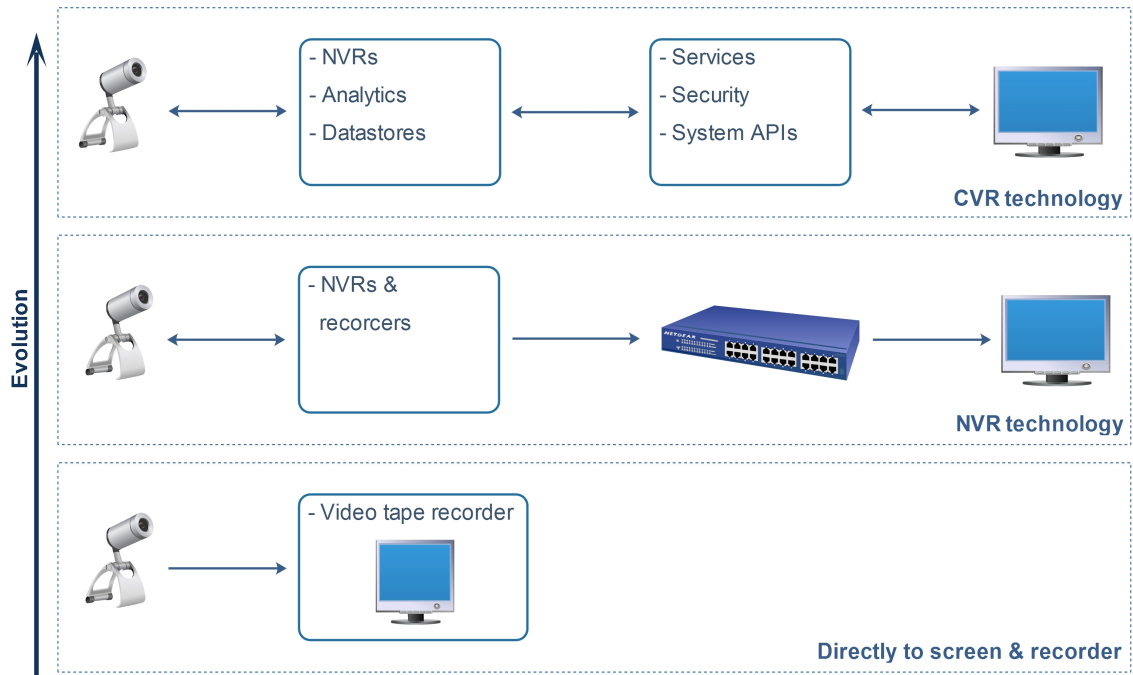


Figure 3: Landmarks in technological advancement of video surveillance architecture.

It was noted in the literature that the global impetus in video surveillance has been towards cloud computing [23] especially since the cost of cloud computing and Cloud Video Recorders (CVR) is dropping. New interest, challenges and opportunities are emerging towards optimising video surveillance architectures [24]. Owing to its capability to record video across communication networks, CVR appears as an appropriate subcomponent for designing reliable architectures for cloud-based video surveillance systems.

An implementation approach for the CVR was proposed in [24], where the authors suggested the use of Infrastructure as a Service (IaaS) - *a cloud computing paradigm in which services are deployed on outsourced virtual infrastructure as opposed to internally managed infrastructure*. The work achieved a scalable and adaptable solution, using high compression algorithm, and H.264 codec for reducing video data size before transmission. The presented evidence shows that the solution is capable of

delivering operations and management of video surveillance across a geographic area such as a city.

2.3. Enterprise Video Surveillance Architecture

Publications in the literature suggested that a research-based project is being introduced to the industry by the IBM, called the IBM Smart Surveillance Analytics (SSA) [22] [21]. The system overall system architecture, which was depicted in Figure 1 above, was described as *“an open framework for event integration and correlation, highly specialized searches based on multiple object attributes, and advanced real-time alerts”* [21]. It was reported that the SSA provides capability and an approach to generating metadata for achieving both real-time analytics and post-event investigation.

A real-time analytics involves small spatial areas plus specific event detection such as identification of threat, alert generation and object tracking. While post-event investigation involves large spatial area such as a city for the purpose of investigation and historical pattern discovery [25]. Figure 4, the block diagram of the components in the architecture, shows that the SSA employs a software stack (that is, SSE, MILS, Middleware etc.), mostly written in C++ framework, to achieve the solutions, including event detection in metadata and real time alert generation. Publication suggested that this solution has been deployed in the retail environments for achieving business intelligence and intrusion alert [26].

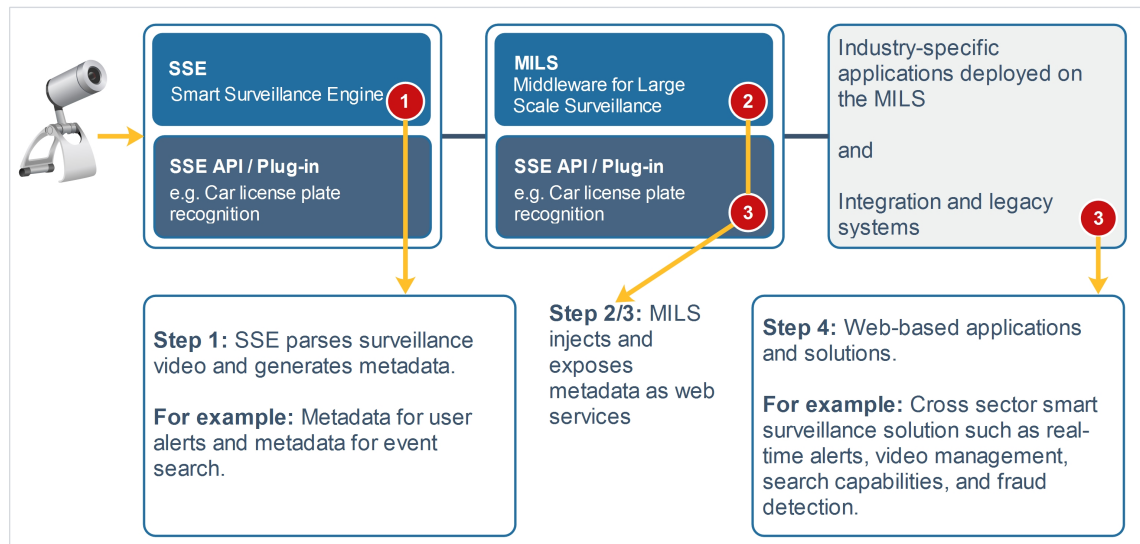


Figure 4: IBM Smart Surveillance Analytics (SSA) [21]

2.3.1. Metadata Architecture

A video metadata specification elements are classified into 3 main categories - which are mandatory elements, optional elements and extension elements [27]. Examples of mandatory elements are file name, keyword, coverage, format and life cycle. Optional elements include version, size, location, and capture time while examples of extension elements are frame count, resolution, sampling, colour, duration and speed.

Neely et al. presented a video metadata approach that has been promoted by the ARDA/DTO/IARPA – the approach was name-tagged ‘Video Analysis Content Extraction program’ (VACE) [17] [28] [29]. Their initial effort was focused on generating ‘annotated temporal log’ from the raw video signals using their metadata extraction solution. Then they classified the log into symbolic representations of events, activities, relationships, and other important attributes, from which they implement a query algorithm named Video Event Representation Language (VERL) and a video markup

algorithm called Video Event Markup Language (VEML) [30]. Both algorithms are metadata representation proposals from the VACE program mentioned in [17] [31].

The approach was an extension of the VEML metadata in which they convert elements of VEML metadata into analytical input for their own novel system called the Analogical Reasoning System (ARS). It sequentially attempts to match behaviours described in the metadata to previously known and analysed behavioural cases, and reason about missing components of the scenes. In Figure 5 below, the author presented an adapted knowledge-adding service, which functions in cloud architecture for the ARS. This service extracts event metadata from connected video sources and publishes them back to the cloud for future analysis, using inherent data storage system.

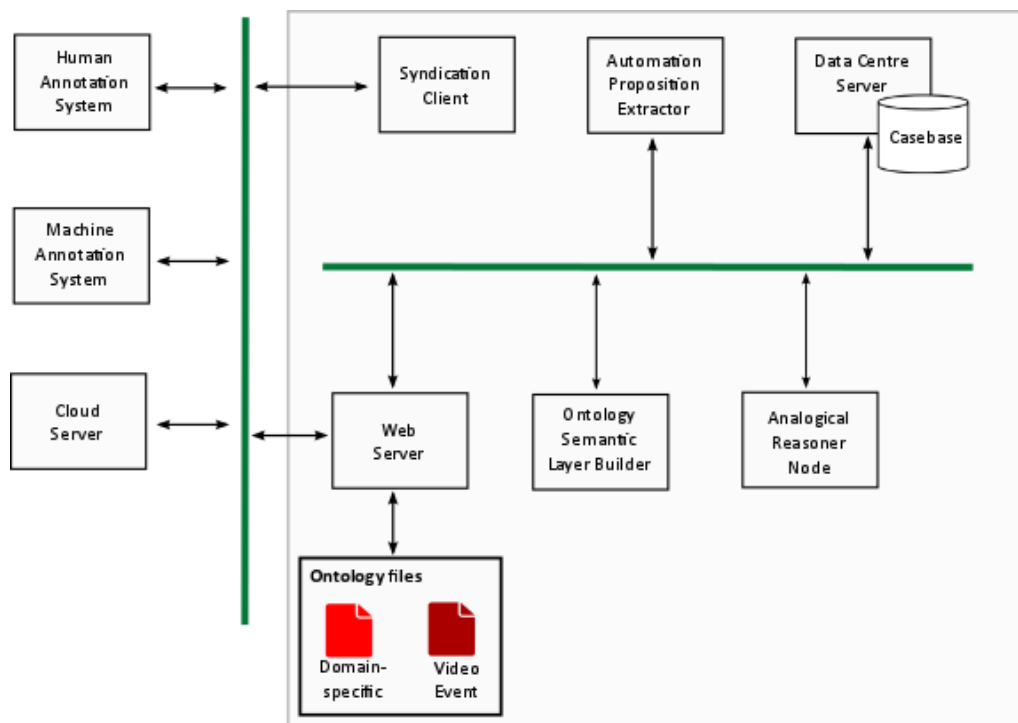


Figure 5: Metadata knowledge-adding service architecture [17]

Notable works were discovered in the literature for generating metadata and analysing the internal processing of surveillance systems. The proposal in [32] focused on the internal transactions in a video surveillance system including remote play, request and response flow. Several works involved the systematic approaches to designing, deploying and implementing automated and event-based metadata from video surveillance systems including ontology and validation of events systems [27] [30] [31] [33] [34]. Metadata persists abstracted structures and content that users can query to retrieve meaningful information such as event detection and object tracking. Metadata can be queried independently of the video images - this can technically solve the problem of data protection.

This research is established on the reality of video metadata based on the theory that surveillance systems can share their metadata. The shared data can solely provide means for matching or comparing interesting events but the video data will not be shared to conceal the identity of people in the video frames. This makes the data useful beyond the political and economic boundaries of the system owners and simultaneously protecting the privacy of the people in the video. A similar concept has been implemented in health informatics where patients' personal health records are de-identified and released for research. The de-identified data could be re-identified in the future for comparative analytics – the process is termed pseudonymisation [35].

2.3.2. Video Metadata Standards

Notable video metadata standards include the Dublin Core Metadata Innovation (DCMI), the ISO/IEC Moving Picture Experts Group's standard (MPEG-7), and the IEEE

LTSC LOM [27] - among which MPEG is most recently updated. The MPEG called for proposals globally in 2008 towards attaining an interoperability solution in the identification of video data. The outcome is a video signature tools that serve as a basis for amendment to the MPEG-7 standards, which is also known as the ISO/IEC 15938 Multimedia Content Description Interface [11] [36].

The work in [36] showed that the amendment offers a set of tools that support unique descriptors on video data that could ultimately detect duplicate and/or derived media content. This will be particularly useful for saving data storage spaces in large video networks and databases such as Youtube [11]. The ability to prevent duplicate data is relevant to this research, as it will enhance the storage of the unified video data that originated across a city sources.

2.4. Self Awareness and Autonomous Systems

The early IBM vision for autonomic computing implies that a self-managing system should be self-configuring, self-healing, self-optimizing, and self-protecting and should exhibit self-aware-ness, self-situation, self-monitoring, self-expression, and self-adjustment [37] [38]. The vision was birthed from the autonomic operation of the human body's nervous system, which was found to oversee the operations of the heart rate and the body temperature without external intervention [37]. To achieve this biologically inspired goal, the proposed architecture configures a node as the autonomic manager, which serves as the nervous system for the other non-autonomic nodes on the network.

The autonomic manager identifies the non-autonomic elements under its influence and monitors both its external environment and executes actions based on the received information. The non-autonomic elements are perceived to be any computer or resources used within the computer system. In other words, the autonomic managers are perceived to relieve the humans of some of the computer administrative tasks. The approach employed in autonomic computing, with the concept of an autonomic manager is not compliant with our approach in this work since we aim at a completely decentralized solution.

Since the emergence of autonomic computing, research efforts have been directed towards actualising and establishing ***autonomous systems***. That is, systems that exhibit a level of self-awareness (SA) and self-expression (SE) to the extent they can achieve their objectives without human intervention [39] [40] [41] [42] [43] [44]. It was noted that SA and SE cuts across multiple disciplines and dimensions including Psychology, Artificial Intelligence & Robotics, Organic Computing, Systems Engineering and Autonomic computing. In particular, [39] gave a good account of how computational self-awareness corresponds to human self-awareness. The work suggested that a healthy and reliable self-aware system is capable of obtaining self-aware knowledge within own internal sensors. Equally, the system can simultaneously capture self-aware knowledge from external phenomena such as integrated servers and the environment.

The work in [39] further compared a proposed framework for computational self-awareness levels with the human self-awareness levels in [45] and [46]. The comparison is summarised in Table 2, showing that each level in the authors' framework maps directly to a corresponding level in the human self-aware framework.

Table 2: Comparing concepts in computational self-awareness and human self-awareness [39]

| Level in [39] | Definition in [39] | Level in [46] | Definition in [46] |
|-----------------------|---|--------------------|---|
| Stimulus awareness | System can respond to processes and events. | Ecological self | Self-perception in relation to the physical environment. |
| Interaction awareness | System can learn the patterns of events. | Interpersonal self | Active knowledge of own emotional involvement in human communication. |
| Time awareness | System can obtain knowledge pertaining to both historical and future phenomena. | Extended self | Actions based on personal memories and expectations. |
| Goal awareness | System can obtain knowledge based on current status and configurations. | Private self | Identifies that some experiences are unique to one. |
| Meta-self awareness | System can review own awareness levels. | Conceptual self | Draws knowledge about own responsibilities based on conceptual roles. |

SA and SE are two key properties, that if jointly present in a system, signifies the system is capable of achieving a level of system autonomy. In the context of computational systems, SA is used to describe the ability of a system to acquire and maintain knowledge about ‘self’ and own context, in relation to achieving expected objective(s) while SE refers to the ability of the system to automate actions and attain adaptive behaviour, following SA [47] [48] [49].

The study by [48] identified 2 inter-related dimensions of adaptive systems (or autonomous systems). First is ‘**where**’ the adaptation takes place – this is further separated into 2 levels: (i) individual level, and (ii) the collective level. Second, they identified ‘**what**’ the possible adaptation mechanisms are – this is further established at 2 levels: (i) self-adaptation, and (ii) self-awareness. Table 3 is a summary of the relationships of the 2 dimensions, based on the study in [48].

Table 3: Cross examining the dimensions of system adaptation [48].

| | Self-Adaptation | Self-Expression |
|------------------------------|--|---|
| Individual adaptation | Agent architectures | Dynamic transformation of internal architectural attributes. |
| Collective adaptation | Organisation patters Swarm intelligence Market-based mechanisms Normative systems | Topology aware systems: Dynamic transformation of system topology and control regime. |

2.4.1. *Self-Awareness in Camera Networks*

The techniques for self-awareness, self-expression and self-management are becoming an established practice in camera networks. These concepts have been implemented to design topology aware camera networks [42] [50] [51] [52] [33] [28]. It was established in [50] that 5 system parameters are fundamental to achieving self-configuration in camera networks: (i) camera, (ii) network, (iii) environment, (iv) task, and (v) performance. The first 3 parameters were used to achieve the cameras' physical network, connectivity and configuration strategy, and the collective field of views (FOVs) of the cameras on the network – these involve the terrestrial and spatial environments covered.

Further, [50] suggested that the task parameter describes the ability of the cameras to communicate and interact with other cameras through task-induced operational decisions. Examples of these decisions are: the detection and tracking of moving objects across the camera network, resource monitoring, and task-load balancing among the cameras on the network. Lastly, performance parameters involve the management of system metrics and an active attempt to achieve a healthy state that fosters reliability and robustness. Ultimately, the activities that actualise self-

reconfiguring in cameras are presented in a cyclic operation as depicted in Figure 6. It shows that the self-configuration process starts at the active vision level and also ends with the same process.

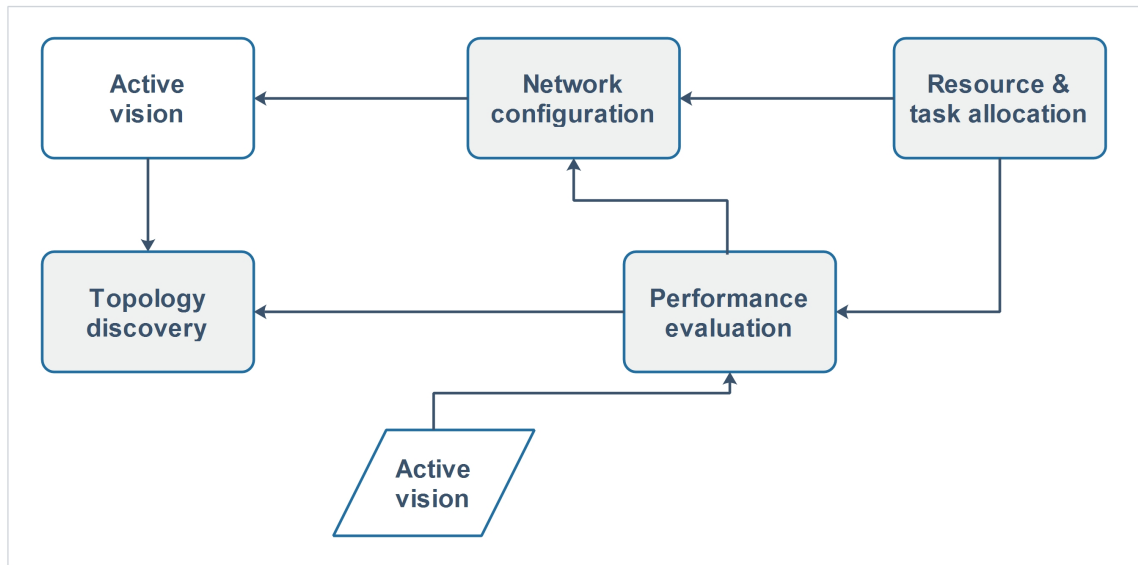


Figure 6: Continuous cycle of the 5 key elements of self-reconfiguration in smart camera network.

2.4.2. Self-Awareness in Name Server

The techniques that have been considered for configuring DNS in IPv6 include (i) router advertisement (ii) Dynamic Host Configuration Protocol (DHCPv6), and (iii) the anycast address [53] [54] - the most relevant of these approaches to this research is the router advertisement. This approach supports the automated configuration of the network nodes, discovery of other nodes and the discovery of the applicable DNS servers, using the Neighbour Discovery Protocol in the link layer of the Internet protocol suite.

In order to discover a DNS server, a node sends out a one-way router advertisement message with a lifetime field. If the lifetime has expired and a response has not been received, the sending node times out the request and switches over to another DNS

server. However, if the host already has a configured DNS server, it reverts back to it, assuming it to be the nearest DNS server. In other words, the host configuration does not aim to acquire the most efficient (in terms of distance from self or any other metrics) DNS configuration; it is based on best effort. This is similar to our approach in this research, the resource directories, which is discussed in chapter 6 does not aim to detect the best neighbours, it simply keeps the records of directory servers in other cities within the same country as itself.

2.5. Topology Learning and Auto-Interactivity

The virtualisation of complex distributed networks has increased research interest in topology aware (TA) algorithms – this is evident in the increased number of published work on topology learning and network centrality in this decade [55] [56] [57] [58] [59] [60] [61]. Network virtualisation involve the abstraction of the hardware and software resources into logical assets, with a view to optimising the system portability, usability, management and administration. The concept of network virtualisation is imperative to the success of this research. The rest of this section 2.5, introduces some concepts and definitions in graph theory that influence our approach to achieving TA, node centrality, and system autonomy in this research experiments.

2.5.1. *Graph Theory Definitions and Concepts*

In this section, we provide definitions to some fundamental concepts of graph theory, based on [62] and [63] (and as applicable to this research). A **vertex** (or a **node** or a **thing** in IoT) is a connection point where 2 or more **edges** meet. An **edge** is a link that connects a pair of nodes, data travels from one node to the other through an edge.

Two nodes that are directly connected by an edge are considered as **adjacent**. A **loop** or self-loop exists in a graph if an edge starts and ends at the same vertex.

A **graph** $G = (V, E)$ consists of a set of V vertices and a set of E edges. A graph is a representation of a communications network by vertices and edges connecting pairs of its vertices, such that the intrinsic properties of the network do not contribute to the characteristics of the graph. A **simple** graph contains a finite set of vertices and a finite set of edges such that no loop exists and each pair of connected edge has a maximum of one edge joining them. A **subgraph** of a graph $G = (V, E)$, is a graph $G' = (G', E')$, where $E' \subseteq E$, and $V' \subseteq V$.

A **weighted** graph is one in which weights are assigned to its vertices and/or edges. The weight may be used to denote frequency or size of the events on the graph. A **directed** graph (or **digraph**) has a pair $G = (V, E)$ where E defines the relation on V , which is a finite set of vertices. The digraph identifies the travel direction between each pair of vertices – the edges of digraphs are also referred to as **arcs** or **arrows**. For example, the weighted digraph in Figure 7 represents the devices on a computing network such that weight 5 is assigned to the edge connecting vertices a and b to indicate the capacity or cost (such as the bandwidth or the attenuation) of the edge. Weight 3 is assigned to vertex a to indicate its **degree** - that is, an arithmetic sum of the edges that it links.

A **path** of length k in a graph is a sequence of distinct vertices v_0, v_1, \dots, v_k such that $v_{i-1}v_i$ is an edge for $i = 1, \dots, k$. A **cycle** in a graph is a path, in which the first vertex is the same as the last and no other vertices are repeated. An **acyclic graph** G is one

where no cycle exists, and a **directed acyclic graph, DAG** (or **acyclic digraph**) is a directed graph with no cycle. Let us denote uv as a valid arc in the digraph $Gd = (u, v)$. If Gd is a simple graph, then it follows that u is said to be an **antecedent** of v .

In a **connected** graph, there exists a path between every pair of vertices. A graph $G = (V, E)$ is a **tree** if G is connected and acyclic. **Spanning tree** for a graph G is a tree derived from G with all its vertices. In a **weighted connected** graph, the minimum spanning tree produces a spanning tree of the graph's minimum total weight.

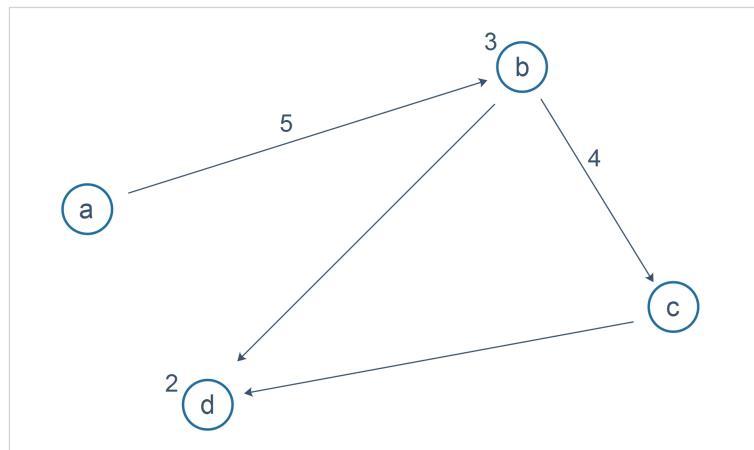


Figure 7: A basic weighted digraph

Equation 1 is an example **adjacency** matrix M , which is usually a $n \times n$ matrix (where n is the number of network nodes) used to represent the relations of the vertices and edges in a graph Gm , with a finite set of vertices and edges, in which the elements of the matrix signify whether each pair of vertex is adjacent or not. For example, Figure 7 is a simple digraph $Gm = (V, E)$ with $V = \{a, b, c, d\}$, and $E = \{ab, bc, bd, cd\}$, where Equation 1 is the adjacency matrix derived from Gm .

A **dynamic routing** procedure (or algorithm) is used to dynamically assign weights to the edges and/or vertices of a network so that they represent the current state of the network based on changing demands on the network. A **protocol** (or a set of rules), is a defined method which the nodes in a network employ to decide *which, how, when* and where to transmit new information. Each node in a network acquires and manages its own routing tables such that the overall network routing calculations are distributed throughout the network.

$$M = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (1)$$

The **routing table** of a node is therefore the computed relations between the *known* vertices and edges of the network, which a node consults to decide the *next* node when forwarding information to a node on the network. For example, based on the depicted configuration of the network in Figure 8, the routing table of node *a* in the network is represented in Table 4. Note that the weight on the edges (in Figure 8) represents bandwidth. The protocol assumed in this example is based on Dijkstra's algorithm [62].

Table 4: The routing table in node a, derived from the network depicted in Figure 8

| Destination node | b | c | d | e |
|------------------|--------------|---------------------|-------|-------|
| Next node | d | d | b | c |
| Bandwidth | 16 | 20 | 24 | 2 |
| Path | (a,d), (d,b) | (a,d), (d,b), (b,c) | (a,b) | (a,e) |

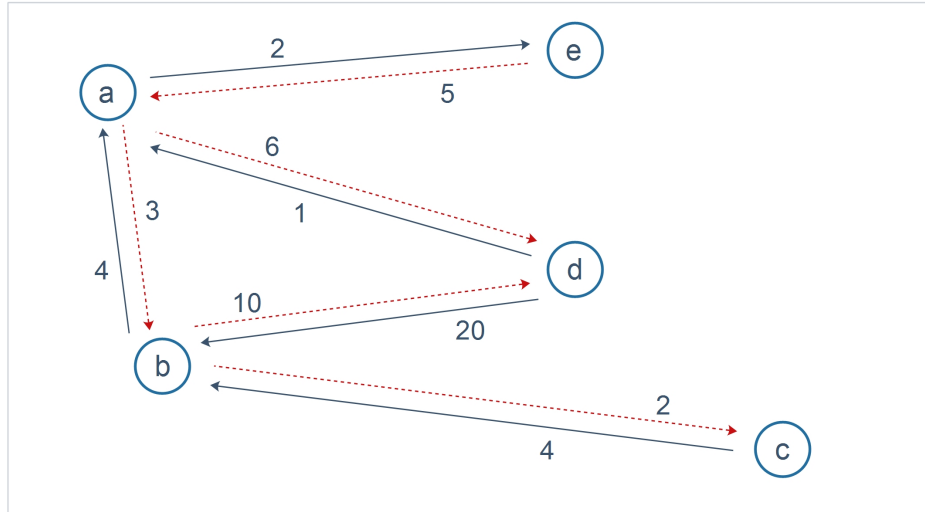


Figure 8: Example of a computing network graph - solid links as downloads, and uploads dotted

2.5.2. Network Centrality

Network **centrality** is a broad concept for assessing and identifying the most ‘important’ vertices (or nodes) in a network. Depending on the measure of interest in the study of centrality, the most important vertices may be considered in terms of the ‘*magnitude of the effect of removing a node*’ (that is, *system science analysis*) or the ‘*magnitude of the influence of a node*’ (that is, *social network analysis*) [55]. It is noted that the study of centrality involved several indicators and parameters but the scope of this thesis does not cover the comprehensive discussion of all the approaches, application and metrics that have been used to implement, compare and assess the efficacy of centrality indicators.

Below, we present basic definitions of some notable centrality methods that support our approach in this research, as presented in the graph theory literature. However, we identify that [55] [64] [65] [66] [67] [68] [69] discussed different approaches to the comparison and the efficacy of centrality measurements. [46] – [52], and [70] [71]

presented approaches in the application of centrality concepts to various projects and environments, and [65] included an extensive proof of stability in closeness, degree and eigenvector centrality indicators.

2.5.3. Closeness Centrality

One of the fundamental indicators of centrality in the literature is the closeness centrality [72] – it is used to rank the influence of a node within the network based on its computed distance to all other nodes in the network [73] [67]. In [67] and [74], the closeness centrality of an undirected and unweighted but connected graph was derived, in which the graph $G = \{V, E\}$ where $V = \{v_1, \dots, v_n\}$ is the set of nodes, and $E = \{e, \dots, e_n\}$ is the set of edges. The closeness centrality C_i of a node i is defined as:

$$C_i \triangleq \frac{N - 1}{\sum_{j \in V} d_{ij}} \quad (2)$$

Where $d_{ij} = d_{ji}$ - the shortest path between i and j . It is observed in (1) above that the larger the value of C_i , the closer (averagely) is node i to all other nodes within G . In Figure 7, for example, node b will compute as the most important node, by closeness – since a maximum of 1 node separates node b and any other node. Conceptually nodes with high closeness can be effective in transmitting data to and from every other node. However, a limitation of this measure is that nodes with very low closeness may cause significantly drop the C_i of nodes they are connected to thereby skewing the behaviour of the connected node [74].

2.5.4. Degree centrality:

Historically noted in [72] as one of the earliest indicators employed in centrality study, this approach is **simplistic** in that, it ranks a node based on the arithmetic sum of its adjacent edges. It follows that the higher the number of adjacent vertices of a node, the higher its degree centrality [65] [75] [72]. For example, in Figure 7, the node b will compute as the most central node, based on the degree approach, since it has 3 adjacent nodes, which is a node higher than any other node on the network.

In this approach, a network may have multiple central nodes at different segments of the network, where clusters appear. It was noted in [76] that the practicality of this concept is limited in that it merely indicates local centrality, it does not indicate the centrality of the full network. The study further provides a simple definition of this approach. For a network with set of node denoted by K (containing $|K| = K$ elements), let the set of edge = L . The set of neighbours of node K is denoted by N_k . It follows that $|N_k|$ refers to the degree of node k .

A limitation of the degree approach is that it does not account for the full network topology since it only considers the centrality of the local network (that is, the sub network), based on the immediate neighbours of a node.

2.5.5. Eigenvector centrality:

The Eigenvector centrality (EVC) measure builds on the concept of degree centrality in that, it thrives on the notion that: *a node is more important within the network if it is*

connected to other important nodes with high degree centrality [76]. The study in [77] provided a definition of EVC as:

$$x_i = \frac{1}{\lambda} \sum_{j \in M(i)} x_j \quad (3)$$

Where $M(i)$ is the set of nodes that are connected to the i th node, and λ is a constant. Eq. 2 computes the value of x_i such that its value depends on the degree of the neighbours (that is, $M(i)$) of node i . If $M(i)$ computes to a high value, then i compute to a high value and vice versa. This can be represented in vector notation as:

$$\vec{x} = \frac{1}{\lambda} A \vec{x} \quad (4)$$

Or equivalently as:

$$A \vec{x} = \lambda \vec{x} \quad (5)$$

With Eq. 4, x represents an eigenvector of the adjacency matrix A whose eigenvalue is the λ and there can be many satisfactory λ . But based on the Perron-Frobenius theorem for non-negative matrices, for non-negative matrices; the highest value of λ that satisfies Eq. 4 yields a non-negative eigenvector, and this value is the eigenvector centrality of the network nodes. The Perron-Frobenius theorem for non-negative matrices states that:

“If the (nontrivial) matrix A has nonnegative entries, then there exists an eigenvector r with nonnegative entries, corresponding to a positive eigenvalue λ . Furthermore, if the matrix A is irreducible (e.g. connected

graph), the eigenvector r has strictly positive entries, is unique and simple (not complex), and the corresponding eigenvalue is the largest eigenvalue of A in absolute value” [77] [78].

The knowledge of the network topology is imperative to our work since the automatic operation, management, and optimization we propose depends on the active learning and adaptation of the network properties such as the nodes interaction, connectivity, and accessibility. Earlier works have proposed approaches to actively learning a network comprising multiple sources and destinations. The most relevant to these work are the works of [79] and [80] – these works showed 2 main approaches.

First is the tomography approach, where the researcher measures the end-to-end activities to infer network topology. For example researchers send probes from a single source in a tree topology and use specific properties of the output probe (such as the number or order) as an input to statistical signal processing technique. The Traceroute-based technique, the second approach relies on nodes across the network to connect the identity of the nodes along the paths and uses the path information to compute a path across the network.

In chapter 5 we demonstrate the use of a traceroute-based approach, supported by the degree centrality of the nodes to predict the path of travel by objects of interest.

2.6. Resource Directory and Service Discovery

A resource directory (RD) is a network service (typically, a software component) that holds information about services and resources hosted on other network objects. It

provides a lookup and discovery facility to the resource or service in response to a network request [2] [81]. The RD does not contribute to the performance of the resource but seeks to provide location and availability information about the resource by modelling the request/response elements, using the concepts of the Internet Protocol (IP) and web properties such as the URI and Internet media type. The activities of the RD will normally rely on the use of a service discovery protocol, which describes the procedures that a request uses to learn the endpoints of the RD.

The importance of resource directories servers (RDS) are already established, with implementation techniques published in various notable works that included proposed standards and several experimental and pilot works [2] [81] [82] [83] [84] [85] [86]. The works found that the anticipated interaction between smart objects in the internet-like structure (which is one of the key concepts of the IoT) could only be efficient, if the registration of new objects, discovery and resolution of the objects are automated. It was suggested that once each smart object acquires a unique identification on the Internet, for example using IPv6, a directory-like ideology could act as a low cost but effective system concept for initiating multidimensional unification among the heterogeneous smart objects and networks in the IoT.

2.6.1. *Centralised Client/Server Architecture*

The architecture for service discovery proposed in [2] is relevant to this research. In their proposition, Cirani et al employed the Constrained Application Protocol (CoAP) [81], which is currently being proposed to the IETF and the IEEE as the standard communication protocol for constrained applications, to provide a mechanism for

service discovery and location. Each CoAP server exposes an interface (called ‘/.well-known/core’) through which a device can send a request to discover available resources. The server replies with a list of resources and an attribute that specifies the data format associated with the resource.

At the application layer, the CoAP is designed to employ the Representation State Transfer (REST) framework for automatically registering and managing smart objects in smart networks. Despite its perceived suitability to distributed systems, the CoAP has been criticized for severe limitations. For example, (i) the CoAP does not provide specifications for a node to announce itself to the RDS, when joining the network for the first time. (ii) The RDS in the CoAP is a centralized node, which makes the concept less appealing to this work. The main limitation of the centralized client/server that makes it unattractive to this research is the potential limitation in scalability of the servers since it was found that continuously increasing in the number of clients may result in system failure [2].

The work in [87] suggested a potential solution to the limitations imposed by the use of the client/server architecture in a website hosting environment, which limits the number of concurrent visitors to a website. The proposed solution involved a 3-tier architecture that provides end user interfaces to publish websites, supports full-text search and indexing, resolves names, manages sessions, and removal of a website membership to a network. Tier-1 is the hosting network, which is essentially the Internet and all the underlying namespace and name resolution schemes.

Tier-2 was introduced to support and group data persistence, while tier-3 is used to administer a novel routing protocol and searching algorithm. However, the solution is not suited to the approach of this research because we empowered selected servers to handle more resources/features, whereas our system only needs the capability of looking up and routing from host to host, the resource directories in our work do not manage system capability and system operations.

2.6.2. Peer-to-Peer Architecture

Peer-to-peer (P2P) framework is designed to scale with the increase in the number of participating nodes, and has become one of the default solutions in the techniques for content distribution and data streaming in distributed systems [88] [87] [89] [90]. This has been demonstrated by the Chord, a P2P protocol [90], which was proposed to achieve 5 system goals:

1. **Load balancing**, in which Chord acts as a distributed hash function, which distributes the key evenly over the nodes.
2. **Decentralization**, the even spread of the keys establishes that no one node is more important than any other node.
3. **Scalability**, the cost of a lookup grows only as the log of the number of nodes, making it desirable even in very large systems.
4. **Availability**, the internal tables are dynamically adjusted to reflect the currently connected nodes – this promotes the knowledge of newly joined nodes and discards disconnected nodes.

5. **Flexible naming convention**, irrespective of real system name of the nodes, Chord uses its own generated keys to map to the nodes, making it possible to keep the name assigned by system administrators but still be a compliant member of the P2P network.

The authors of the P2P network proposed in [88] groups the nodes into grid membership so that each grid assigns distance and other cost information to each member node. This information is then used to generate indexing and search criteria, which are used to assign new nodes to the appropriate grid membership. A more relevant approach to establishing neighbourhood information about the network peers was found in GeoKad [91] [92], a P2P protocol that is optimized to use a distance metric to identify neighbourhood information about other nodes.

The nodes in GeoKad periodically perform lookup on other peers to decide nearness information by making a web service call, using a *get(lat, lon, rad)* signature. This call returns a collection of peers within the radius *rad* with a centre in the location of the specified latitude and longitude (*lat,lon*). This protocol was demonstrated to provide a near accurate result and scales in the order of P2P networks.

The demonstrated approaches in the P2P network are relevant to this research's approach to integrating the CRDs. It is noted that the P2P approach could provide mechanisms that we can build on in this research. For example, the demonstrated features in Chord and the GeoKad DHTs are relevant to the desired characteristics of the network of CRDs in this research.

2.7. Object Detection and Identification

In computer vision, object tracking refers to the process of continuous identification and location of moving object(s) across a sequence of video frames where the video frames could be sourced from single or multiple cameras. The study in [93] suggested that, to effectively track an object, the system must achieve the 4 sub components:

1. Object identification – the process that detects the object to track.
2. Object tracking – this is described in the above paragraph.
3. Object association – a process that builds patterns and/or events using the tracked objects.
4. Analysis – interpretation of the patterns for the final purpose, which could be decision-support, alert-trigger systems, security systems, among others.

Research work have achieved a significant success in near-accurate segmentation of still images [94] [95]. In addition, several research activities have explored the hypothesis of automatic human detection and tracking (in video sequences) with combination of several monocular cues such as colour, texture and disparity to classify the body into segments such as head, torso, upper legs, lower legs, and feet [93] [96].

In outdoor scenes however, unresolved problems still plague the achievement of consistent segmentation and tracking dynamic multi-view video sequences due to complexities such as abrupt object motion, non-rigid shape distortion, object and scene occlusions, variable camera trajectory, and camera motion [97] [98]. For example, threshold-included colour histogram has been tested but it appeared to ignore the adjacency information of pixels, which can be problematic when identifying

multiple colours since it unify similar colour/density irrespective of their location in the image or video frame [98]. Because of these outstanding challenges, a number of work on interaction of colour and depth cues are restricted to indoor scenes [97] [96].

2.7.1. *Object Identification and Classification*

Several approaches have been employed to detect and classify objects in video surveillance researches, with each approach leaning towards a set of criteria that best support the conditions and features of the video and prevalent technology. For example, object classification approaches have been based on the features in objects throughout the video shot (shot-based approach), while others are based on the features of objects in selected video frames (object-based approach) [99] [100].

These employed a combination of established frameworks such as the k-nearest neighbour, hidden Markov models, support vector machine (SVM), finite-state machine, and neural networks to model the classifiers. Whereas, the classification process use various criteria such as foreground or background extraction, movement estimation, image segmentation to detect the objects [101] [102] [103]. These are relatively common techniques and approaches so we will not discuss them further in this article.

2.7.2. *Autonomous Multi-Camera Coordination*

The ability to dynamically manage camera actions such as tracking, coverage optimization and pan-tilt-zoom (PTZ) is important in surveillance, transportation and other activity recognition environments [104] [105] [106]. In [104], it was noted that

images from multiple cameras can be fused to improve multi-camera coordination and tracking. In their multi-camera fusion experiment, a network node is assigned the role of fusion centre. The fusion centre is a node with high performance output and capability, which collects data from cameras within the network cluster (or sub-network) and performs state estimation. The result of the estimation is then shared with other fusion centres on the network to determine state estimates among the clusters.

The fusion centre calculates the correspondence and measurement activities on camera and data properties such as the camera position, bounding boxes and trajectories. Then it applies the outcome, using data association and tracking algorithms such as graph matching and to match and track the objects across the network. In the Graph Matching (GM) concept, for example, objects at entry and exit points of each camera are considered as nodes on a network graph whereas, an edge between a pair of vertices, Z_a and Z_b , is weighted by their similarity, that is $\zeta(Z_a, Z_b)$. Further, the appearance features such as colour is used to measure similarity of target objects across cameras. Whereas a path smoothness function, based on the variance of the feature is used to cater for variation of the object due to different illumination conditions.

The fusion methods showed advantages such as scalability and the reduction of overall network load since it limits the transmission of measurements among neighbouring cameras, there are yet to be considered an efficient multi-object multi-camera tracking approaches. For example, the prevalent track-to-measurement association for multiple targets with higher accuracy incurs higher processing costs than tolerable in a large

network. It is however noted as an active research area with potential for future improvement.

2.7.3. *Head Count and Distance Estimation*

Researchers have proposed approaches to automatically estimate the number of people in a camera network, including approaches to estimate their distances from the cameras [5] [26] [107] [108] [109]. The solution proposed in [107] attempts to estimate the number of people covered by a network of cameras. The method involves 3 steps:

1. Estimation of crowd density in an area, which is then compared with the transformation of the people across the location. The crowd density $d(n)$ is given by:

$$d(n) = \frac{d_{r+1} - d_r}{n_{r+1} - n_r} (n - n_r) + d_r \quad (6)$$

Where r is the r -th region, as in $[n_r, n_{r+1}]$.

2. Calculation of crowd velocity based on optical flow. The optical flow is derived using $OF_u(K)$ and $OF_v(K)$, where u and v are the horizontal and vertical components, and k is the video frame. Velocity is calculated by finding the speed and direction of the optical flow - the speed is obtained by:

$$OF_\rho(k) = \sqrt{OF_u^2(k) + OF_v^2(k)} \quad (7)$$

ρ is an assumed constant for mapping group relations between crowd density and the number of people such that: $\{(d_i, n_i)\}_{i=1 \dots p}$, where d_i is the crowd density of the i -th group; n_i is the number of people.

While the direction is given by:

$$OF_{\theta}(k) = \begin{cases} \arctan \frac{u}{v}, & u > 0, v > 0 \\ \arctan \frac{u}{v} + \pi, & u < 0 \\ \arctan \frac{u}{v} + 2\pi, & u > 0, v < 0 \end{cases} \quad (8)$$

Where θ is derived by sampling the direction with M bins histogram (that is 4 or 8) such that $\theta = \frac{(2M_{max}-1)\pi}{M}$.

3. The prediction of crowd density based on directed structural diagram – this involved the estimation of number of people an area a moment of time before the crowd arrival. Figure 9 is used to describe the approach where node A represents the area of interest, installed with camera C0. Nodes B, C, D and F are installed with cameras C1, C2, C3, and C4 respectively and are distances away from A as in S1, S2, S3, and S4 respectively. It follows that the estimated number of people is given by:

$$W_A^! = W_A + W_B + W_D + W_E - W_F \quad (9)$$

Note that W_A is the current number of people, and this value is assumed to be a known beforehand. $W_B - W_F$ are values of people currently in nodes B – F, which are calculated in real time.

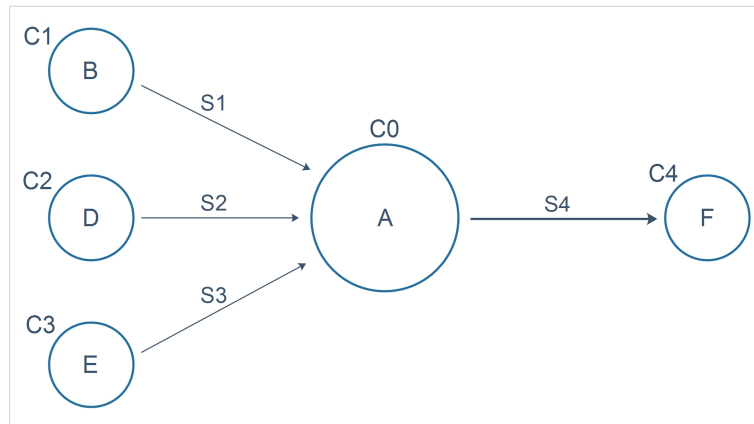


Figure 9: Directed structural diagram for people estimation [107]

2.7.4. Visual RGB and Depth Data Approaches

Since the invention of Microsoft Kinect sensor, which combines visual (RGB) and depth data for motion detection, there has been a rise in applications that combine these 2 approaches [110] [111], with depth data only playing auxiliary role to RGB data - this is chiefly because existing depth sensors are not reliable enough since most video signals are either low resolution or possess inherent unstable regions and holes. The approach relies on complementary operations of 2 separate cameras systems – first a colour camera and second a depth sensing system, which comprise of an infrared (IR) camera and an IR projector. The IR devices jointly generate and process IR speckles on the 3-D scenery to produce the depth data while the colour camera, unable to read IR rays, captures and delivers three basic colour components (RGB) of the video. The basic components of the Kinect are depicted in Figure 10.

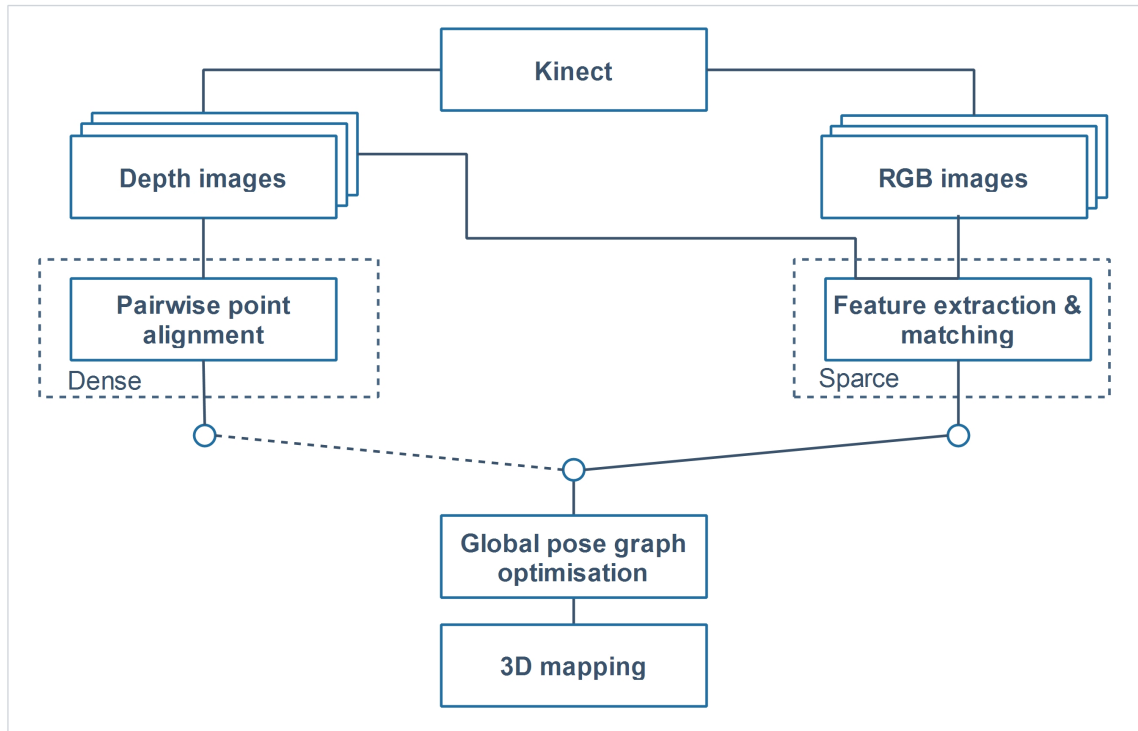


Figure 10: Components of Kinect, an indoor 3D mapping system [111]

2.8. Person Re-Identification

Person re-identification (Re-ID) is a process used to automatically detect, analyse and review the features of people in images captured by cameras, with a view to extract evidence that match the same persons across multiple images [112] [113]. It is noted that several avenues have been explored to achieving person re-id. As summarised in Table 5, the 2 broad methods are – first are the **contextual methods**, the attributes of the cameras such as camera geometry, network topology, space-time cues, and camera calibration are explored to achieve re-id. In the **non-contextual methods**, which appears to be more effective [112], the attributes of the persons are the main focus for re-id. Both active and passive non-contextual methods rely on the extraction and analysis of the visual attributes of the person. Re-id is systematically reviewed with references to the works that proposed them in [112], [114] - if reader is interested.

Table 5: Person Re-identification methods [112]

| | | |
|---------------------------|------------------------|---------------------------|
| 1. Contextual methods | 1.1 Camera geometry | |
| | 1.2 Camera calibration | |
| 2. Non-contextual methods | 2.1 Passive | |
| | 2.1 Active | 2.2.1 Colour calibration |
| | | 2.2.2 Descriptor learning |
| | | 2.2.3 Distance learning |

The most effective approaches involved the extraction of identifying features and the assignment of descriptors to a collection of images, which is generally known as the gallery. The gallery may contain a fixed number of images where the images in the probe are collectively a subset the gallery - this type is the ***closed set re-id***. However, in second approach, the ***open set re-id***, the gallery images are open to changes and the probe images are unknown beforehand.

In both open and closed re-id, each image in the gallery is assigned a system ID, which uniquely identifies the image. The remaining images are then each compared with already ID-ed images in the gallery in a pairwise manner. If a match is found, corresponding mapping is computed to highlight the existence of similarity, and ultimately a confirmation of re-identification of the person(s) in the 2 images.

Re-id is a very challenging ordeal and despite the increased attention given to this research area in both academic and commercial communities, the highest achievable match to date is still just over 53% [115] [112] [113] [116] [117] [118] [119] [120]. The accuracy of the match is however weighted in a ranked fashion such that multiple images can appear to match the same ID-ed image but ranking is used to order the level of closeness to a perfect match.

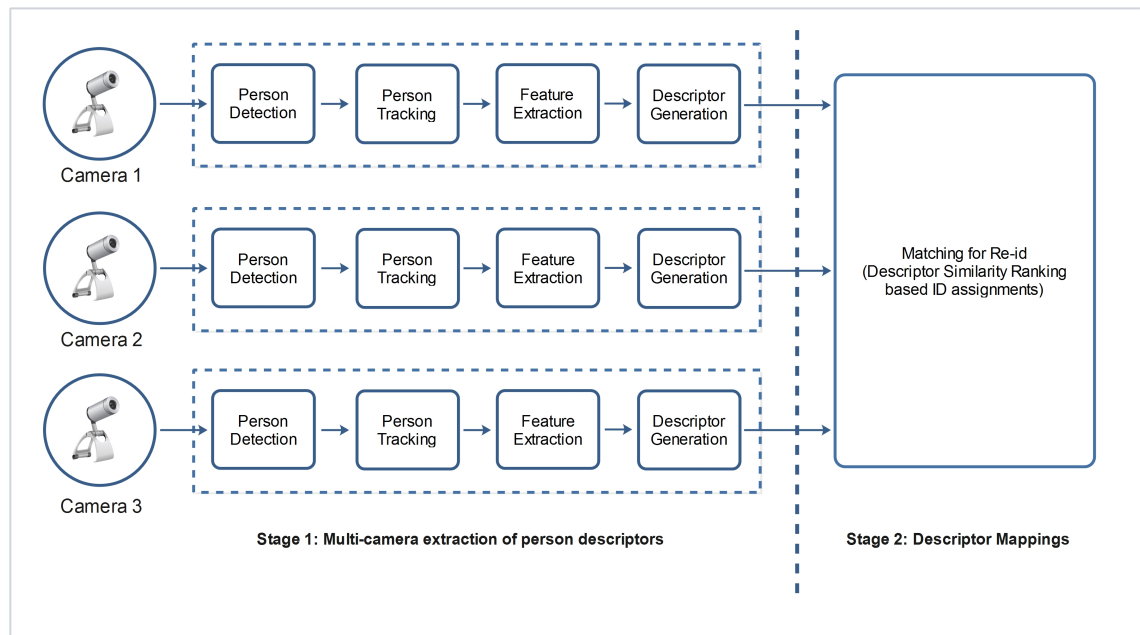


Figure 11: Overview of Re-ID process

Figure 11 is a high-level process flow found in the majority of published works so far – it is observed that research activities are leaning towards the non-contextual methods (seeing that re-id processes are concerned about person feature analysis). The Figure shows the 2 main stages to re-id – the first stage is where images are captured and processed to extract identifying attributes (of persons) as images progress from one sub-stage to the other, while the second stage is tasked to find the match among the identified persons.

Each stage and sub-stage is plagued by complex problems that are still open research challenges. In fact, the problems account for the low confidence level in re-id matching mentioned above. This is because the non-contextual method, which is to date the more effective, relies on visual attributes of people such as their appearance, clothing colour, clothing texture etc. However, these attributes change sporadically and are

unpredictable, especially over a long period of time since people change clothing and lighting conditions differs immensely.

2.9. Storage Solution for Surveillance Systems

Cloud computing is consistently being suggested for video surveillance systems, with capable of compressing the data before storing it [121] [122] [24] [123]. Cloud computing denote a collection of internet-based facilities, which includes software and hardware services (or servers), agents, processes, infrastructure and facilities that have been configured to collectively provide fault-tolerance and high scalability services in a distributed environment, a cloud computing system will usually be providing services for more than a single user – as such, it is usually not physically resident on the local network of its users.

In another work, Dey et al. proposed a solution capable of continuous I/O manipulations, read/write mix, random vs. sequential access with supporting variety of input sources [122]. Others have suggested storing video data in the cloud where growth becomes elastic and affordable [123]. However, while cloud storage is profitable and realistic solutions in most cases for extremely sensitive and/or massive data environments such as defence, cloud storage is not an option. As mentioned earlier, video from several surveillance cameras would consume massive bandwidth and storage resources, and the video data can be highly sensitive. It would appear beneficial to persist video surveillance data within the local network with support for accessibility via a cloud based application layer.

The service of cloud computing suggested for video data is the store – a cloud store comprise of a variety of storage network devices to data storage and access through an application cluster and grid technology, which concurrently persist and retrieve large amount, velocity and variety of data using stacks of network hardware and software technologies [121]. The implementation of cloud storage is expected to increase system availability, reliability, scalability and cost-effectiveness for a data-intensive system. Camera and encoder can be connected to the cloud storage through an access link for real-time persistence of the resulting video signals [121] [124].

2.10. Web Services

A web service is a platform independent software that is designed, implemented, and deployed to support interoperable communication between machines (such as a computer) over the network such as the Internet [125] [126]. Web services are the key components in Service Oriented Architecture (SOA), which is an umbrella term for describing design patterns and architectural approaches that support communications between systems that are independent of vendors, programming language, and the underlying technology. In this thesis, we considered the two most widely-adopted approaches (or protocols) to actualising web services: the SOAP-based approach (Simple Object Access Protocol), and the RESTful approach (Representational State Transfer) [127] [128] [129] [130].

Each approach provides a set of features and supports technologies such that, the decision to choose one instead of the other calls for a careful evaluation. A comparison of the two approaches was discovered in the literature [130], the comparison was

carried out from 2 perspectives. First, based on **architectural principles**, and secondly, based on **architectural decisions**. Below is a brief summary of the comparison as it influenced our decision in this research. However, readers are referred to [130] and [131], if interested in further details and comparison of these services.

REST is designed as client-server architecture to be used with the http protocol - the http protocol is the standard transmission protocol for the web. It is based on 3 main principles. 1) **Addressability**, REST acquires a dataset to operate on as a resource that is identified via a URL (a standard means for locating objects over the Internet). 2) **Uniform interface**, resources can only be manipulated using the 4 uniform operations – they are the same in all cases, and they are never required to change – that is, **POST, GET, PUT, and DELETE**. POST is used to transfer a new state onto a resource; GET is used to obtain the current state of a resource; PUT is used to create a new resource, while DELETE is used as its name. 3) **Statelessness**, this means that each message to the web services is self-descriptive, self-contained and does not rely on an existing communication.

The message representation may conform to either XML or JSON format (see Acronyms). At top-level, a rest message is structured into three elements. First is the **endpoint**, which defines the URL of the resource to be operated on. Second is the **header**, which contains information about the data format of the resource. And third, is the **method**, which must be from the uniform interfaces. Service identification of resources is achieved using the standard URI addressing mechanism, which facilitates the encapsulation of information required to uniquely identify and locate a service globally. However, the service description has to be manually compiled in which the

developer writes the code to assemble the URI of the resource and correctly encode/decode the representation of the resource. In terms of quality of service (QoS) and security, the level of guarantee of delivery offered by a RESTful as good as the http offers (which is, best effort) while it supports the security achievable with https, such as data encryption.

SOAP message representation adheres to the XML message architecture and format, which defines the envelope element at the top level. The envelope contains the **header** and **body** elements, while the body may further contain the **fault** element. The header can be extended to transmit routing information (such as addressing schemes), security and QoS information, and configuration information such as security requirements, among others. The SOAP web service employs the WSDL (Web Services Description Language) to define and publish the structure and format of the endpoints including the syntax and how to use the web methods (published web services). The definition in WSDL makes it possible for the compiler to generate the service description that can be processed by the system.

SOAP is transport independent – that is apart from being exchanged over the web (that is, using http), a SOAP message may be exchanged by any other complaint protocol such as TCP, SMTP, and JMS among others. This provides a level of flexibility and adaptability – for example, SOAP services can be used to exchange both request-response and one-way transactions since protocols such as JMS supports asynchronous transmission. In terms of security, SOAP allows extensibility on top of the security offered by the transport protocol - although this is necessary since SOAP

non-secure protocols such as http. This makes it possible to define QoS parameters that guarantee successful communication.

As mentioned, the above summary is based on the comprehensive comparison published in [130]. Among several other comparison criteria discussed in the literature, Table 6 is a snapshot of the summary based on both architectural principles and architectural decisions perspectives.

Table 6: REST and SOAP web services compared

| Comparison concepts | REST | SOAP |
|---|----------------|-------------|
| Transport protocol (HTTP, JMS, SMTP, MQ) | HTML | All |
| Message format (XML, JSON, YAML, MIME) | All | XML |
| Service description (textual, WADL, WSDL) | All | WSDL |
| Security (HTTPS, WS-Security) | HTTPS | Both |
| Service Discovery (UDDI, Do-it-yourself) | Do-it-yourself | Both |
| Transactions (WS-AT, WS-BA, WS-CAF, Do-it-yourself) | Do-it-yourself | All |
| Service Identification (URI, WS-Addressing) | URI | Both |
| Resource relationship (Do-it-yourself) | Supported | N/A |

2.11. Conclusion of Literature Review

In this literature review, this research discovered significant advances towards the automation of video surveillance systems and architecture. We present literatures and evidence of works that support enterprise-scale video surveillance architecture, self-awareness and autonomy in video surveillance technologies. Our findings include technologies that automatically coordinate multi-people tracking by multi-cameras,

crowd estimation, distance estimation, and automation of video storage facilities. An automatic architecture and framework was noted, which provided capability to query metadata generated from a surveillance network and equally achieve both real time and post event analytics on the surveillance data.

Despite these and advances and achievements, there is a great deal more to do to optimise video surveillance systems to improve the benefits. These works provide evidence and various approaches that contribute to the motivation and direction of this research. Based on the available information in the reviewed literature, to our knowledge, no one has proposed an approach for unifying and exploring independent video surveillance systems at the public level.

That said we discover that the IBM SSA was designed to facilitate analytics of both real-time and post event analytics - it is however presented as a standalone systems for a specific business environment. Similarly, we found intelligent approaches to detection, identification and classification of objects in video data - although they are still evolving. Further, we found support for the ability to automatically track multiple objects across multiple cameras was suggested. Standardisation approaches were suggested for metadata generation and implementation in several works providing evidence and support for analytics of video surveillance systems.

The works that were reviewed in this chapter are mostly state of the art in video surveillance and they are in the same direction as our research - these indicate that this research conforms to the state of the art and the future of technology in video surveillance. Based on our knowledge from the evidences in the literature (which

includes both academy and commercial community), we found that all the existing researches and literature have considered video surveillance systems as isolated systems, based on the system boundaries of the independent owner. We identified that the isolation of the surveillance systems are a major reason for the limitation (research gap) that we identified in chapter 1. However, this research provided a unique and novel perspective into solving these problems.

Based on the identified research gaps in this literature review, this research identified the opportunities of unifying and integrating the independent systems and explored the feasibility of public-level exploration of the unified video surveillance systems. We proposed the overall system architecture capable of supporting the unification and integration of multiple but independent surveillance systems, based on real world technologies in chapter 3. Then in chapter 4, we gave a detailed account of the main components of the architecture, as implemented in the experiments in this research.

We implement a simulation project, which involved an implementation of a non-contextual re-id method (following our findings in section 2.8). This was achieved in chapter 5 where we also satisfy this research's use case by using the experiment to predict the location of an interesting object that has been detected on the network. Finally we demonstrate the feasibility and efficacy of this research's ambition to propose an approach for unifying independent surveillance networks. This involved an implementation of the globally scoped surveillance system in a country and we demonstrate how a local surveillance system interacts with it.

3. The Fused Video Surveillance System Architecture

In order to change an existing paradigm you do not struggle to try and change the problematic model. You create a new model and make the old one obsolete.

- Richard Buckminster Fuller

3.1. Overview

In chapter 2, we reviewed the literature and identified existing works, and open questions that contribute to our work. It was noted that, in both the research and industrial communities, the current level of technological advancement has achieved support for the following system features and services:

- Web services that can be implemented to unify independent surveillance systems. Albeit, significant work would be required to achieve and demonstrate a reliable security system at the massive scale and volume of the collaboration suggested in this research.
- Cloud based web-enabled storage solutions with capability to run distributed applications.
- Web services that can provide a level of security, which could be implemented to secure communication over machine-to-machine networks.
- Digital cameras that record high quality video data – although it was mentioned earlier that most old systems are non-digital yet.
- The generation of video metadata is actively being researched. Solutions are already emerging in the research community and are currently being introduced to the industry.

Considering the above system capabilities, the desired concepts in this research appear as being supported by the current technologies. Based on the availability of the support, this research proposes a systematic approach to designing a system design

that leverage the capabilities in these features and services to achieve a globally unified surveillance system - the proposed architecture is described in this chapter. This chapter describes the overall architecture that supports our proposal for a surveillance system, which is query-able at the public level.

In section 3.3, we outline our assumptions, goals and design considerations while section 3.4 is a list of our design considerations and some assumptions about the need and usability for the proposed system. Section 3.5 describes the components of the architecture and their relationships. In section 3.6, we suggest and provide support for the practicality of our approach in terms of deployment and usability in real life environments. Section 3.7 introduces the integration and application of the FVSA into the IoT paradigm and section 3.8 concludes this chapter - we discussed relevance, strengths and envisaged challenges of our proposition and future direction for video surveillance systems based on this solution.

3.2. Current Systems

Existing research into video metadata has chiefly focused on the generation and accessing of metadata by the administrative owner of the system. Our solution, the FVSA presents video surveillance systems as an adapted computational grid of intelligent services, which is integration-enabled to communicate with other compatible systems in the Internet of Things (IoT).

A notable implementation of a computational grid based on the IoT is smart cities, which is a complex system comprising several unrelated lifeline services such as environmental information systems, smart energy grid, travel information, waste

management, urban planning, smart meters, emergency response, and smart events, which are being integrated across a common framework, (usually by implementing big data technology stack) [132] [122].

However, despite progressive trends of integrating systems across industries, as in smart city, video surveillance systems are deployed and administered as standalone systems. Video data originates from each surveillance camera in large volumes without means to aggregately explore the embedded information. This is mainly because of complexities that are technical, financial, socio-cultural, security and ethically inclined, such as the following:

- **Data protection** – owners of video surveillance systems have a sense of responsibility to protect the privacy of the people captured in their data.
- **Data ownership** – fear of loss of full ownership and/or control over data if shared outside their own network facilities.
- **Heavy cost and investment** - surveillance systems were usually installed into the building structure; replacement may disrupt many other services, the financial cost can seem unrealistic or unreasonable.
- **System incompatibility** – based on manufacturer/vendor configuration and video encoding, video from each camera has a format that does not necessarily make it readily compatible with video from another camera.
- **Cost of computing bandwidth** – continuous and consecutive transmission of video by several cameras across the network, where many video frames may not contain interesting events.

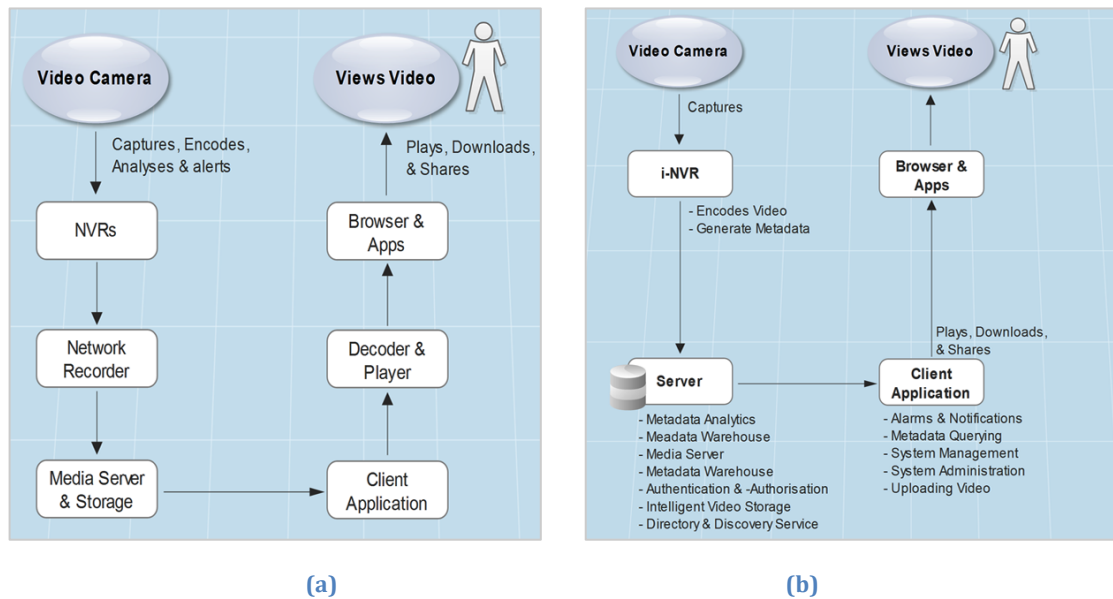


Figure 12: Process flow for streaming video in: (a) current systems (b) the FVSA.

This chapter presents and describes how we resolve the complexities described above. It is noted that current video surveillance architectures have been successful in the sense that they deter vandalism and provide a level of security to their administrative owners/managers [32]. Figure 12a is a typical process flow in video surveillance systems. It shows that anyone with access to the computer screen or TV can view data from any camera on the network.

In the current systems, a typical business model places a security officer in front of multiple screens where the officer attentively monitors video streams from the cameras in order to detect, investigate and raise alarms in the event of unwanted or unexpected scenes. Some of these systems provide the capability to watch real-time video from any camera on the network – permission to view the data is normally assumed since only authorised officers have physical access to the CCTV rooms. In recent years, as mentioned above in chapter 2, some of these systems are configurable to trigger alarms by sending email or SMS in the event of unwanted or unexpected

events. With the proposal of this research (as in Figure 12b), a security office is able to stream video from any device running the system portal, provided they have the system privileges to do so.

3.3. Design Goals

The FVSA aims to analyse the events from video metadata as they are generated from cameras on the network. It provides authentication and authorisation to ensure that only permitted users can access the system, where each user only has access as appropriate for his/her role. For example, while a security officer in a train station has been granted permission to view all surveillance data including real-time video, a police officer, may only have access to alerts that are triggered from the same train station.

Similarly, a permitted police officer is conceptually aware of all video surveillance systems in town (through the directory server, which we describe in chapter 5) and can seek permission to query them. Figure 13 is a map of the areas surrounding University of Sussex, UK - as seen by a city police officer (for example) using the FVSA. The map shows the FVSA deployed at four locations: a university campus, a stadium, Southern Water, and a train station. A city police officer has selected to view full details of the element of the Sussex FVSA system. An overview of the FVSA is provided in section 4 below.

As noted earlier, surveillance data is the property and responsibility of the system owner. However, with appropriate security and privacy enforcement in place, and a

safety officer (such as a police officer) can be granted limited permission (time-limited or access-limited) to stream video data, which can help towards an investigation.

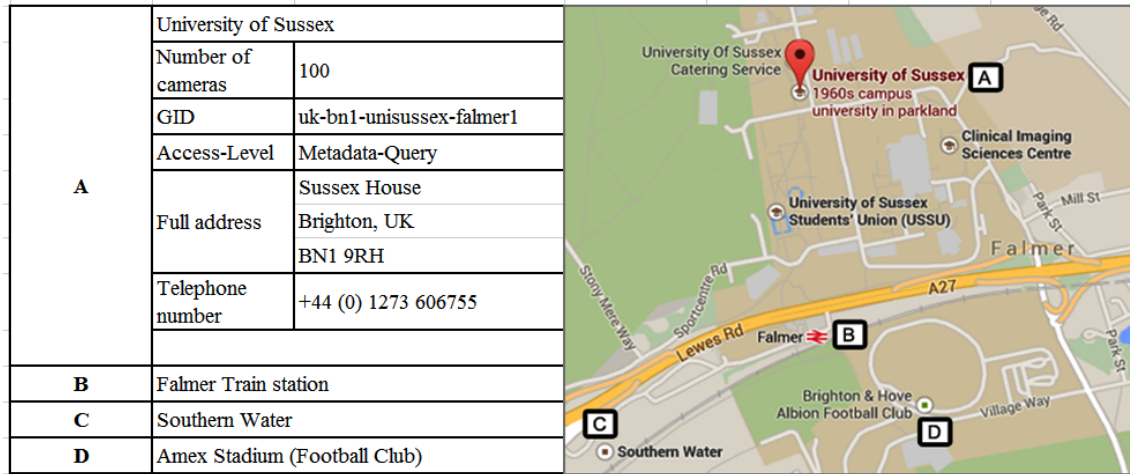


Figure 13: Topology of the video surveillance system in a city - A conceptual police view

Our goals revolve around the need to optimise the video surveillance systems as technology advances towards aggregated analytics in the sense of the IoT, smart city, and hierarchical communications - we explain this further in section 3.7. Summarily, a video surveillance system based on our proposal will achieve the following requirements:

- To reduce the cost of investigation – the police currently appeals for evidence from the public when investigating incidents. The FVSA can make data readily available for such investigations, so police can automatically query any ‘open’ video surveillance systems to build up evidence.
- To unify the data mining interface of independent video surveillance systems through a robust API.

- Surveillance system can interoperate in existing computational grid systems, such as in a smart city or Cisco Service-Oriented Network Architecture (SONA) [133].
- Potential integration point for further sources of surveillance data such as satellite images, social media, which can provide useful information.
- To increase the accuracy of results obtained by public safety departments while the owners of independent surveillance system still protects their 'real' video data.
- Autonomous and continuous identification, tracking and investigation of objects from any camera on the network. And to generate statistical information for informed decision-making
- Apply a level of authorisation and authentication on the data to prevent fraudulent access. A user of the system (in this research) are required to provide a valid username/password before access is granted – the level of access is determined by a set of pre-determined system permissions.
- Perform high data compression on the video data so they are cheaper to store for a reasonable length of time.

3.4. Design Considerations

The suggested architecture of this proposal is based on considerations and assumptions as follows:

- Public safety departments are interested in using video from privately owned surveillance systems.

- We assume that current video systems can be preserved while the new architecture is implemented. However, a new video surveillance system will benefit immensely from this new structure.
- We assume that owners or managers of CCTV systems will find our proposal more profitable and more beneficial.
- We assume cameras are unintelligent recording devices - so all processing is achieved within the i-NVR.

3.5. The Overall System Architecture of the FVSA

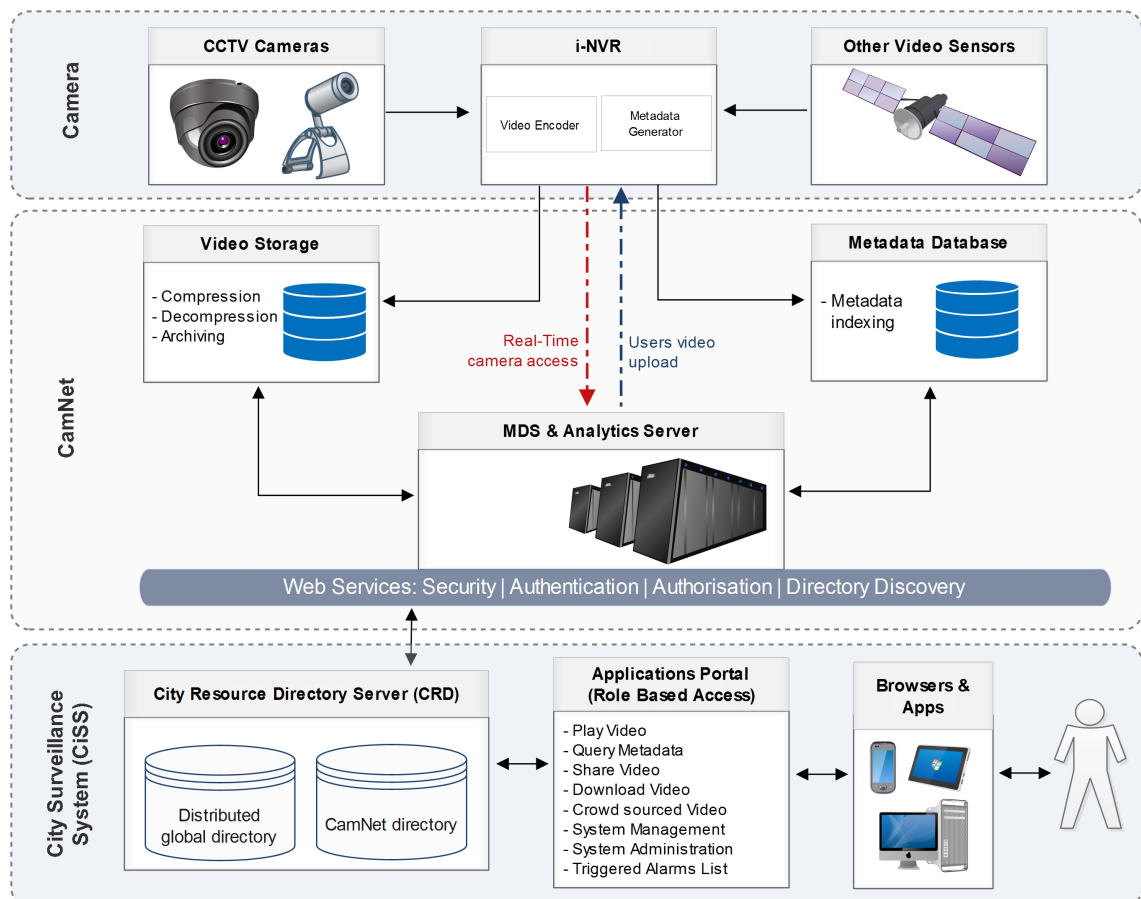


Figure 14: The high level conceptual model of the FVSA.

3.5.1. *Overview of the FVSA*

Figure 14 depicts the high-level architecture of the FVSA - it presents the system services in the modular view. In practice, some of the modules depicted are merged – for example, the web services, metadata server (excluding storage), and queue services are all installed on the analytics server. The depicted architecture features the following system modules (i) cameras, (ii) intelligent Network Video Recorders (i-NVR), (iii) a queue service, (iv) a metadata server (MDS), (v) a metadata warehouse, (vi) an analytics server (vii) web services (viii) a global directory server (ix) user computer system.

As discussed in chapter 1, a CamNet is made up of the collection of surveillance camera(s) that jointly belong to the same administrative owner. The CamNet also includes all other devices, components (both software and hardware), integrated services, and operations on the CamNet. The operation of the individual module is described below. It is worth noting that the researchers envisage a flexible set up of the framework described above, depending on the number of installed cameras and budget.

3.5.2. *The Camera*

In an enterprise CamNet, such as one presented in Figure 14, a surveillance camera requires minimum analytics processing power since metadata generation is performed externally (at the i-NVR as discussed above). In practice however, the camera may host the video encoding software and also the metadata generation software. In essence, the CamNet may have only 1 camera or several thousand cameras. An example of a 1-

camera CamNet is the residential home with 1 CCTV camera, and an example of multi-camera CamNet is the collection of all the CCTV cameras in a university campus. A 1-camera CamNet can perform the combined operations of a simple camera plus the functions of an i-NVR – in this case, the camera records and also generates metadata.

In a large system however, all video processing can be achieved at the i-NVR, while unintelligent cameras can be used to capture data. A system administrator can configure several cameras onto the same CamNet even if the cameras are deployed in different physical locations, as in different streets of the same city. For an organisation with branches across various cities and/or countries, the FVSA can be leveraged to administer all the CCTV systems from all locations. This can be achieved by setting up the i-NVR hierarchically as described in section 3.6. The CamNet is further discussed and experimented with, in chapter 4.

3.5.3. *Intelligent Network Video Recorders (i-NVR)*

In addition to functioning as the hub that physically connects several surveillance cameras (as mentioned in section 2.2) - in this research, the i-NVR supports the cameras connected to it. The i-NVR encodes the video files and generates metadata before sending both to their respective storage solutions as depicted above in Figure 14. Where it was shown that metadata are sent to the metadata server, while the video data are sent to the video storage solution. In practice, the video storage solution could be local or cloud-based.

3.5.4. Analytics Server

The analytics server is a stack of services that performs several administrative and management operations of the network. It hosts the MDS, which is responsible for generating the logical topology of the network. We explained in chapter 4, that the TA algorithm depends solely on the object tracking and re-identification and does not represent the physical topology of the physical network. The analytics server hosts the web services and other unification services such as event generation, running queries and device discovery. It also hosts database engines/solution for storing and managing the metadata.

3.5.5. Video Storage

The intelligent video storage is empowered to dynamically compress, decompress, and archive video data. It compresses data before persisting it for as long as configured but it can decompress and transmit a specified block of video on request. When the configured time lapses, the storage solution deletes old videos to provide space for more recent data.

Metadata contains information that was extracted from the video frames including camera identity, captured objects, and system owner. Data exploration and analytics are carried out on the metadata, so accuracy of results and reports depends on the quality of the metadata. To send metadata to the CRD, the Metadata Server (MDS) must be included in any implementation of this architecture irrespective of the network size. It indexes and stores the metadata, and it's responsible for the following operations:

- Knowledge of all the cameras on the network (it receives data from them).
- Metadata is the main integrated resource in this architecture – all surveillance querying/investigation is carried out on the metadata through the API.
- It acts as network identifier for integrating with the other CamNets in the city, through the CRD.

3.5.6. Web services

Considering the potentially huge cost implications, and also system limitations (such as attenuation and compatibility issues), it would appear impossible to physically connect all the surveillance systems across a city, such as suggested in this research. However, the implementation of web service interfaces (APIs) is key to achieving the loose coupling proposed to unify them. As discussed in chapter 2, Web services provide means to publish specific features or functionality, and provide security to decide ‘who can’ and ‘how to’ access them.

In practice, since web services are platform independent, the web service can be either SOAP based or RESTful provided behavioural, performance, and security requirements are achieved. However, the experiments in this research implements SOAP-based web services. This research implements the SOAP-based services since its features better match to our requirements - in particular, the out-of-the-box authentication provides a good use case. The web service layer supports the achievement of integration and unification between the various components in Figure 14. These include system security in the sense of authentication, authorisation, trust, session management and system audit for establishing how data is being accessed.

3.5.7. Directory Server

This directory service is used to discover, validate and organise a unique identity for all deployed instances of video surveillance systems that register with it. The service is responsible for cataloguing available systems details, and contact details. The high-level functionality of this service is described in the next section. In practice, security firms and public safety departments such as the police will own and administer these services, and surveillance system owners can configure their systems as private (data will not be shared with any directory service) or public, where the system registers with the directory service. This module of the FVSA is the basis for chapter 5 and 6 below.

3.5.8. The User

This comprises of the user portal and support for multiple devices such as a desktop computer, tablet, mobile phones and remote sensing devices such as satellite cameras, road traffic cameras, and mobile devices used by public safety officers. The portal provides an interface for capturing data from different devices and for requesting and responding to user actions such as uploading data, playing video and querying the metadata.

3.6. Hierarchies, System Scope and Visibility

A network architecture based on a flat design, which is one where all routing devices have full knowledge of the network, can only grow to a limited size – where the limitation is dictated by the capacity of the routers' memory size, processing power and transmission speeds. In order to build large networks, where both inter-network

and intra-network routing can scale efficiently, there is a need for hierarchical design [134]. A hierarchical network is partitioned into areas (or sub-networks) where each routing device only has full knowledge of its own local area. For each sub-network, there is an inter-network router, which has knowledge of neighbouring sub-networks. In practice, sub-networks are usually based on network ownership, geographical area covered or overall size of the network. Examples sub-networks are based on floor sub-networks, departmental networks, overall company networks, and city networks. Although these partitions are usually political and ownership defined, they enhance scalability, performance, security and efficiency of the bigger network.

The FVSA depicts video surveillance systems as a hierarchical system, where subsystem boundaries are based on administrative ownership and geographical location. Additionally, metadata servers (MDS) handle routing activities as discussed below. They are configurable as intra-system (local scope) or inter-system (global scope). An MDS in the local scope has full knowledge of the topological details of all the cameras in the system but does not have any knowledge about any external camera. However, in the global scope, an MDS provides connectivity to external surveillance systems through the Directory service as described below.

Figure 15 shows that an L-MDS only has knowledge of cameras that are directly connected to itself, and those that connect through an i-NVR and those that are connected to neighbouring L-MDSs. Any G-MDS knows how to contact any other G-MDS that is connected to the directory server, however the level of access or visibility depends on the role of the user. For example, in Figure 15, the various L-MDS in the city mall system represents various independent FVSA systems in the mall, where

different shops own and independently manage their own surveillance system. The mall's authority however provides a G-MDS, which every shop can connect to. The mall authority manages the G-MDS and at the same time, the G-MDS can provide connectivity to the city police. With this in place, the mall authority can provide evidence of events without the police physically visiting the mall.

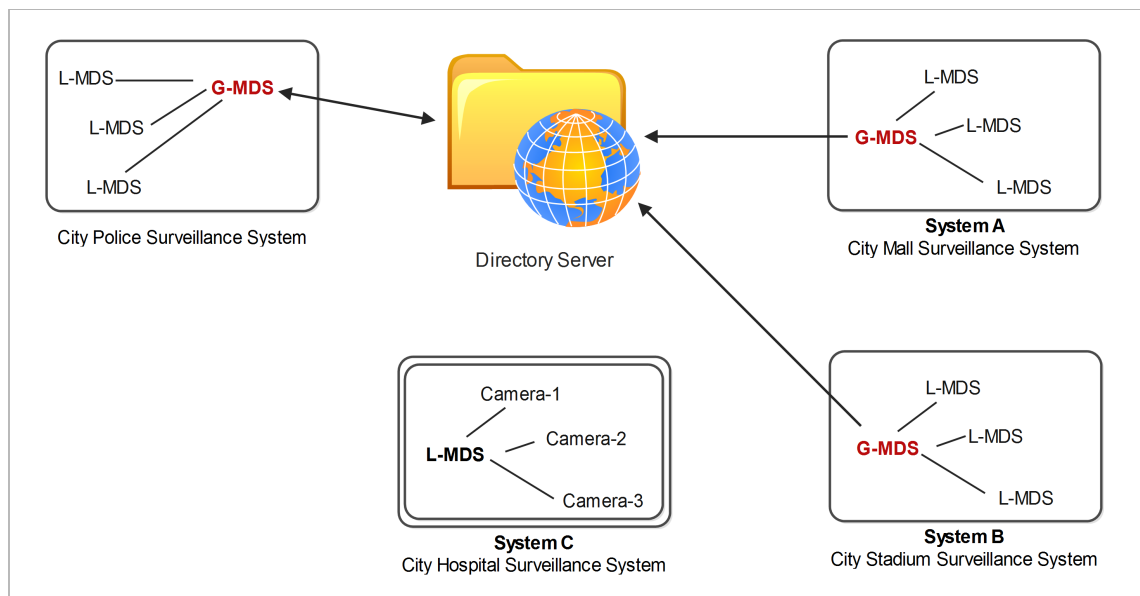


Figure 15: Global and local scope of the MDS

3.6.1. *Authorisation and Resource Visibility*

Any information destined outside the system has to be initiated by a G-MDS, provided the user meets authentication and authorisation requirements. Only the local administrator has full authorisation on all system services. Any user that is not local to the system has to be granted authorisation to use a specific service. For example, by default, a police officer can view a system overview of any connected surveillance system but to play video or query such a system, the system owner must first authorise the access. In Table 7, it is noted that all external users are not allowed

access to the service but public safety officers such as the police may be given authorisation to access some services.

Table 7: Visibility and authorization of system services (in Figure 15).

| Services in system A | An admin of system A | An admin of system B | A city police officer |
|--|-----------------------------|-----------------------------|-----------------------------------|
| Views system overview: cameras, and contact information. | Yes | No | Yes |
| Plays recorded video. | Yes | No | No, unless permitted by system A. |
| Queries System | Yes | No | No, unless permitted by system A. |
| Receives feeds and alerts | Yes | No | No, unless permitted by system A. |
| Configures/updates system or cameras. | Yes | No | No |

3.7. Application

Figure 16 shows the surveillance system in a smart city network - it is noted that each FVSA layer is relevant to a layer in other grid computing platforms, such as smart cities. The layers (or hierarchies) in this view of the architecture fall into either hardware domain (physical and network layers) or software domain (services and application layers). The physical layer comprises all the devices that capture video such as cameras. The network layer includes all network and switching devices such as the routers, MDS, and mobile antennas. The services layer comprises of network-based data solutions and service APIs such as cloud storage. The application layer comprises of client applications and services through which users interact with the system such as video player and query browser.

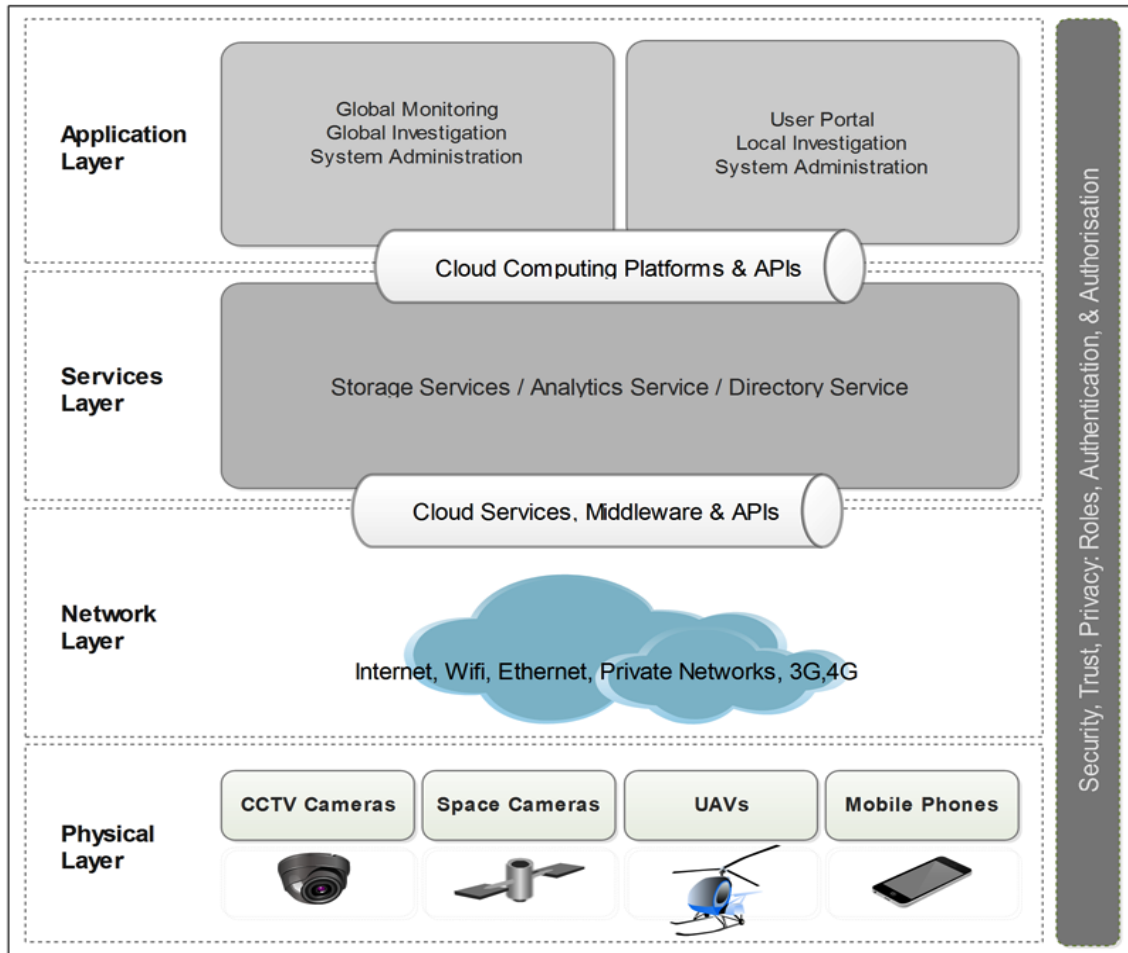


Figure 16 Layered architecture of the FVSA showing its relevance to other IoT compatible architecture [135] [136] [132]

3.8. Chapter Conclusion

Based on the author's industrial experience and the literature review, which was presented in chapter 2, this chapter introduced the overall framework of our system, which encapsulates the system as discussed in this thesis. The suggested framework in this chapter is an overview of our solution for the problems described in the introduction of this thesis – it equally highlights major areas that are still work in progress. The solutions proposed by the FVSA include unification of independent surveillance systems. As described in section 3.5, each implementation of the FVSA is independent while several instances can integrate to form a larger system (or a unified system), such as a city's surveillance system. The same section also introduced the

directory server, which is the integration catalogue for unifying the systems. This in place, section 3.6 introduced how authorised public safety officers can ‘browse’ all connected surveillance systems within their jurisdiction, with latent ability to review alerts and video from any camera. In section 3.7, we demonstrate FVSA’s compatibility with other hierarchical network solutions such as a smart city.

Ultimately, we suggest a hierarchical design and a high-level configuration for video surveillance devices and services, making it possible to approach video networks in layers such as internal system (local) or external system (global). Hierarchical design is an approach engineers employ to abstract complex multifaceted requirements into granular manageable subsystems. The framework of our solution is compatible with the hierarchical structure of computer networks and emerging technologies – the IoT. The next chapter presents an approach to designing and configuring a simulation project that demonstrates the design presented in this chapter.

4. The Experiment - Design and Implementation Strategy

If we knew what it was we were doing, it would not be called research, would it? To raise new questions, new possibilities, to regard old problems from a new angle, requires creative imagination and marks real advance in science.

- Albert Einstein

4.1. Overview

In chapter 3, we presented the proposed system architecture that supports our claim to unifying independent video surveillance systems. We explained that the proposed architecture supports and agrees with the current state of the art, we also explained how our proposal fits and supports the emerging technologies, in terms of the IoT. The modules in the design capture, encode, analyse, and persist video data in a systematic approach that supports our proposal.

To demonstrate the feasibility of the research proposal in chapter 3, we completed 2 simulation experiments – one in chapters 5, and the other in chapter 6. Here in chapter 4, we introduce and account for the internal configuration and design strategy for each component in the experiments. We show how the modules interact to support the hierarchical structure of the world. That is, homes and *street levels*; the CRDs represent the *cities*, while a collection of the CRDs represents a *country*.

We discuss how components are represented in the experiments, and how each module contributes to the overall operation of the FVSA. In essence, this chapter provided answers to two research questions asked in section 1.5. First the developmental question in **RQ1: “Can we design surveillance systems with a view to exchanging information across independent networks?”**. Second, part b of the developmental question in **RQ2: “Can the architecture represent the geo/political structure of the world?”**. This chapter provides an approach to answering these questions, showing the design strategies, configuration of the components, and representation of the data in our experiments in chapters 5 and 6.

4.2. Design Strategy

As mentioned in section 1.8, in this research, a **CamNet** is a surveillance camera or a group of surveillance cameras that belong to the same owner, at the same location. A City Surveillance System (**CiSS**) is a collection of CamNets in the same city, where all the CamNets in a city are registered in the City Resource Directory (**CRD**). A network of all the CRDs in a country is referred to as the Country Surveillance System (**CoSS**), and CRD that is assigned to serve as the server to the CRDs in the country is referred to as the Gateway Resource Directory (**GWRD**).

Both the CamNet and CiSS are capable of computing a logical network based on their metadata. The logical network is a graph where the camera whose object is identified becomes the network node. If the same object is re-identified, then there is a link between the two nodes (or cameras). The logical network is referred to as the Fused Video Surveillance Network (**FVSN**).

Figure 17 is the high-level block diagram (the sequence of the processes - numbered) of the design in the simulation experiments. It shows that the camera, which generates the metadata, starts the process. The metadata is sent to the MDS, which processes and saves the metadata in the CamNet database (see section 4.3.1). Then the MDS sends the metadata to the city surveillance system (CiSS), where it is saved and computed into the CamNet FVSN. ***The FVSN is the answer to the research question in RQ1*** since queries can be employed to explore and analyse the FVSN - as in chapter 5.

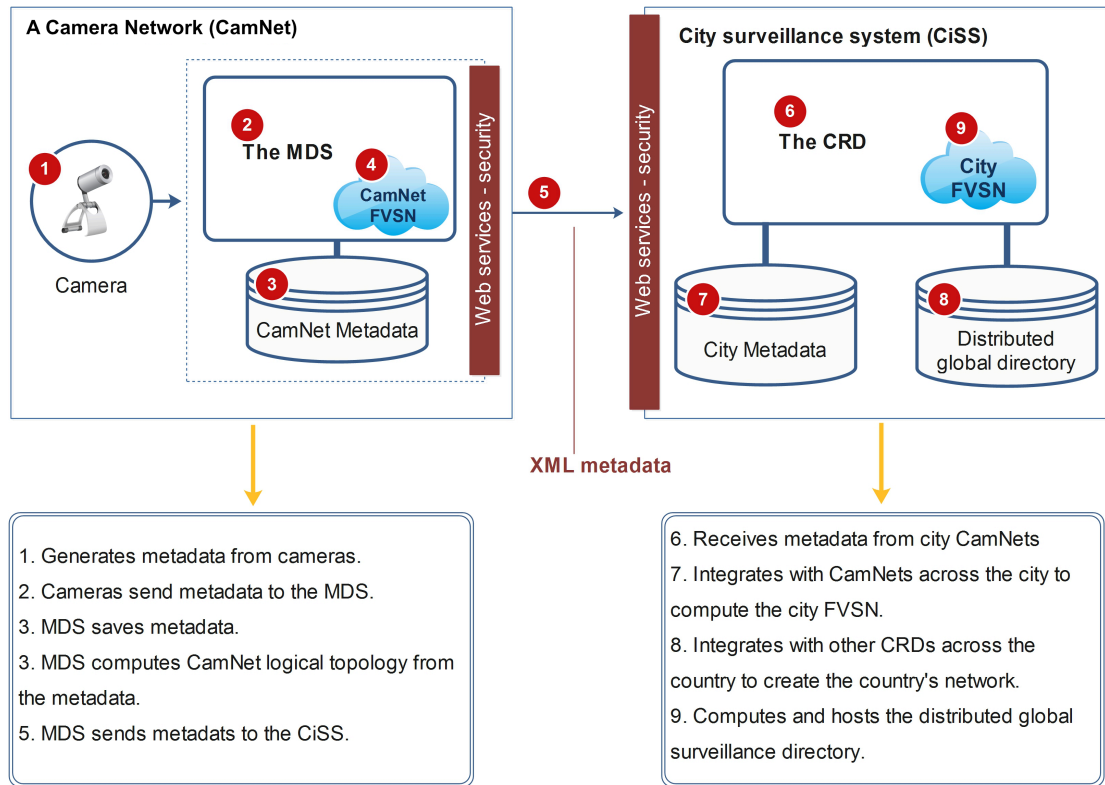


Figure 17: A Simplified high-level architecture of the research experiment.

In the simulation experiments, each component (that is, the camera, the CRD etc.) is designed and represented with database fields, and then the database fields is abstracted into a reusable object – we used PHP and Matlab in 2 separate projects. For example, each camera on the network has a corresponding database table where all its features and properties are configured as database fields. Whenever a particular camera is needed, its fields are selected and instantiated as an object of the camera class, as described in section 4.3.

To achieve a loosely coupled integration between the devices – that is, between cameras and an MDS, then between an MDS and a CRD, the MDS and the CRD were designed each interface of the MDS and CRD as a web service, which provided APIs (or endpoints) through which the services are consumed. Based on our review in chapter

2, the web service could be either SOAP-based or RESTful, provided behavioural and security requirements are achieved. The experiments in this research implement the SOAP-based approach because, its features discussed below makes it more practicable and suited than the RESTful approach:

- **Out of the box support** - SOAP provides standardised approaches to some non-functional features that were desired in the research such as the implementation of trust and authorisation between components, devices and system users. For example, it provides a directive to enforcing integrity and confidentiality on messages using XML signature and encryption to provide end-to-end security [131].
- **Security** - It readily supports the concept of security, in that a successful SOAP call must be authenticated with a unique username/password combination. This is particularly important to our design since we require authentication for users and devices when integrating each. Rest does not enforce this notion of authentication.
- **Standardisation** - It relies on the generic transport protocol and readily supports the industry-standards SSL, which is easily implemented across the devices implemented in this research.
- **Compatibility** - The components in the experiments (such as the camera and the MDS) are abstracted and encapsulated as xml objects. Since SOAP messages are formatted as XML-based, it readily supports our approach to virtualising our experimental network.

4.3. The CamNet Configuration

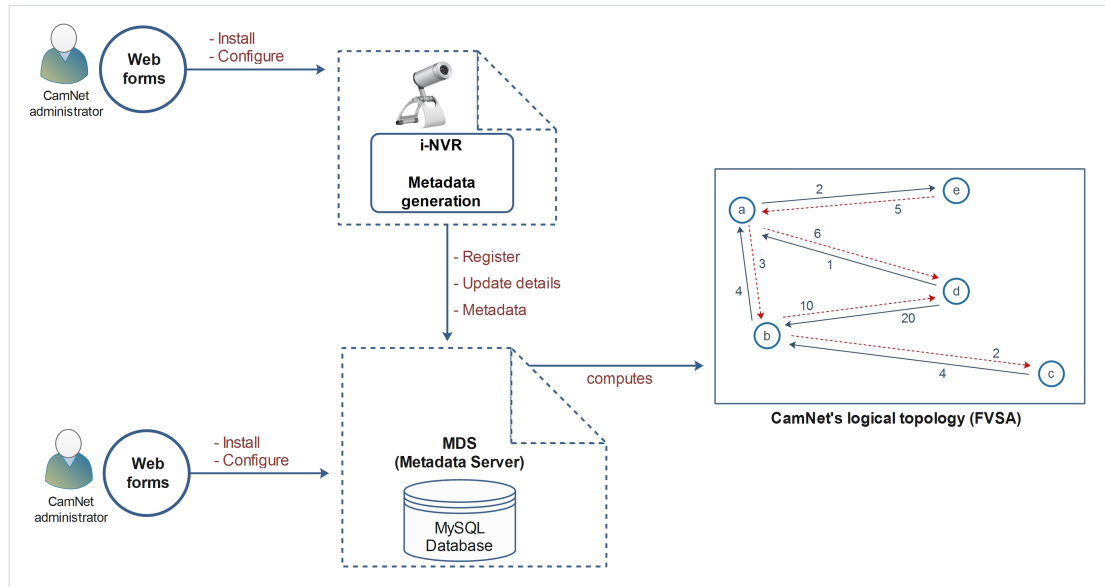


Figure 18: Architecture of the CamNet

The block diagram in Figure 18 depicts each component of the CamNet as implemented in the experiments. The web forms are used to install, manage, and configure the devices – that is, the cameras, and the MDS. Additionally, it is used to administer and manage the CamNet’s FVSN. The experiment simulates camera objects that generate metadata, which are saved in MySQL database. Once metadata is saved in the database, the MDS periodically computes the FVSN, which is based on the pattern in which objects were identified across cameras that generated the metadata. The detailed design and generation approach for the FVSN are discussed in chapter 5.

4.3.1. The Camera

Based on our implementation, the camera is responsible for capturing and generating the metadata that it sends to the MDS. The cameras in this experiment only generate and persist metadata since the availability of metadata satisfies our research

objectives. A camera is not aware of the topology of the CamNet and is not aware of any other camera - it's visibility and knowledge of existence is limited to itself and the MDS that it sends metadata to. As suggested in the last section, the camera is implemented by abstracting the database representation of the camera properties into a programmable object.

4.3.2. *The Camera Object*

The approach for encapsulating the camera object is crucial to the realisation of integrating the metadata from multiple CamNets. The camera object is created as an abstraction of the database properties of the camera, which are described in Table 8 – the left column is the name of the property while the right column describes it. The complete database definition and the data types are presented in Appendix C. The system administrator manually inputs the latitude and longitude. Although it is noted that, in practice, a better approach would be the installation of radio device that automatically detects location of the camera. The 'mds_url' field holds information about the location (URI) of the MDS. This field is used to acquire the URL of the MDS, which is discussed in section 4.6.

Table 8: Database properties of the Camera.

| Column name | Description |
|--------------------|--|
| id | Automatically generate system id (auto-increments). |
| camera_id | A unique camera identity number. Similar to IMEI in phones. |
| lp_address | The network IP address |
| camnet_camera_id | An identifier assigned by the MDS upon registration as a CamNet-based camera. |
| camera_name | A human readable name assigned by a system admin. In a large surveillance network, this can contain name of the department, building, floor etc. |
| camera_description | A human readable description assigned by a system |

| | |
|--------------------|---|
| | admin. |
| mds_url | The url of the metadata server of the local CamNet. This is used to communicate with the MDS. |
| username | This is used to authenticate with the CamNet MDS. |
| password | This is used to authenticate with the CamNet MDS. |
| longitude | Longitude of the physical location of the camera. |
| latitude | Latitude of the physical location of the camera. |
| direction | The direction faced by this camera. |
| camera_rojection | The angle of projection in relation to the ground. |
| owner_name | The name of the camera owner. |
| owner_email | Owner's email address. |
| owner_phone | Owner's phone number. |
| privacy | Signifies the highest level of access to the metadata generated by the camera. |
| address1 | Line 1 of the address where camera is installed. |
| address2 | Line 2 of the address where camera is installed. |
| city | City the of the camera's location. |
| region | Region/county of the camera's location. |
| postcode | Zip/postal code of the camera's location. |
| country | The country of the camera |
| metadata_frequency | The frequently of generating metadata – this is only effective, if the privacy field is NOT set to 'private' (see sention 4.3.6). |
| status | This signifies if the camera is currently active or not. |

4.3.3. *The Camera Set up*

A human is required to configure the camera either during the initial installation of the camera or when updating the properties of the camera. We designed a user interface to configure the cameras. Figure 19 is a screenshot of the user interface for this purpose. It shows the properties of the camera that are configurable by the system administrator. These are grouped into 4 categories: Configuration, camera identity, location, and ownership information. These are all represented in the database table presented in Table 8. The country field is missing on the web form since this is detected from the camera's IP address.

CHOOSE A COUNTRY:
UNITED KINGDOM

Fused Video Surveillance Architecture - the FVSA

[View RDs](#) | [Register a RD](#) | [View Camera Networks](#) | [Register a Camera Network](#)

REGISTRATION - CAMERA NETWORK

CONFIGURATION

-- Enable City Data Sharing? --
-- Enable Self Managing? --

IDENTITY INFORMATION

-- Number of Cameras in this Network? --
90.255.49.209
===AUTO_GENERATED===

OWNERSHIP DETAILS

Name
Email
Phone

ADDRESS

United Kingdom
Address 1
Address 2
Address 3
City
Region
Postcode

SUBMIT

Figure 19: Camera registration form.

4.3.4. Camera Configuration

The privacy property of the camera signifies if the camera will send its metadata to the MDS, and/or if will allow access to use the metadata in building the network topology.

A manually configured camera can be assigned of the following 3 privacy levels:

1. **Full access:** The metadata is allowed to be shared with the city and can be by the city for any safety-related activities. For example, a camera placed at the front of a store may be configured to this level of privacy.
2. **Query:** This configuration sends the metadata to the city but the metadata can only contribute to the query.
3. **Private:** The metadata will not be shared at all.

4.3.5. Camera Registration with the MDS

A camera with privacy level set to either 'full' or 'query' will register with the MDS upon a successful installation. The registration process involves the verification of the location and the identity of the camera. The MDS will only accept a camera registration, if the IP address of the camera is within the same local network and if it satisfies the security requirements. A registration message originating outside the network will be dropped without any processing or response. If registration is successful however, the MDS responds with a unique identification number.

The identification number is then used to encapsulate all future communications with the MDS – this is in addition to the username and password of the user. Figure 20 is a sequence diagram of the registration process and how decisions are made to either honour or drop the registration request for a camera. The depicted flows are for a manual registration triggered by a person, such as the system administrator.

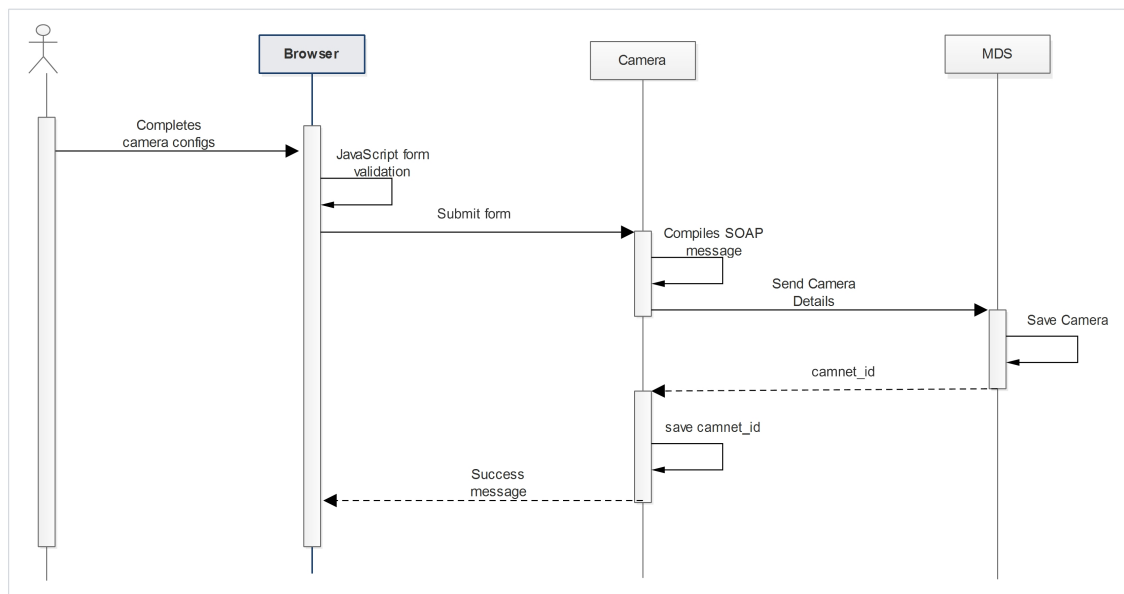


Figure 20: Sequence diagram of the camera registration process

4.3.6. The Metadata

The camera generates and sends metadata to the MDS periodically, based on the interval assigned to its 'metadata_frequency' field (if the privacy is *not* set to **private**). In our experiment, an identifiable object is constructed by referencing a pre-built object in the database. The object is encapsulated and introduced unto the FVSN, at intervals (see chapter 5 for further details). The digital image is then saved in a secure location while a reference to its location and other identifying and ownership information are encapsulated and persisted in the database. Table 9 depicts a flat list of all the properties in the metadata - however, the normalised version of the tables is available in Appendix C. Also we present the SOAP XML in section 4.3.8

Table 9: Properties of the metadata

| Column name | Description |
|-------------------|---|
| id | Automatically generate system id (auto-increments). |
| camnet_id | A unique device identity number. Similar to IMEI in phones. |
| camera_id | ID of the camera that captured the metadata image. |
| camnet_camera_id | A CamNet-wide unique id – this is generated and sent by the MDS upon successful registration with the MDS |
| journey_id | The absolute path of the location of the metadata video frame. |
| object_id | The database id of the object in the objects database. |
| top_colour | The top colour of the captured person – for Re-ID. |
| bottom_colour | The bottom colour of the captured person – for Re-ID. |
| height | The height of the captured person - for Re-ID. |
| texture | The texture of the captured person - for Re-ID. |
| longitude | Longitude of the physical location of the camera. |
| latitude | Latitude of the physical location of the camera. |
| velocity | The velocity of travel by the object in the metadata. |
| direction | The direction faced by this camera. |
| camera_projection | The angle of projection in relation to the ground. |
| name | The name of the camera owner. |
| email | Owner's email address. |
| phone | Owner's phone number. |

| | |
|----------|--|
| privacy | Signifies the highest level of access to the metadata generated by the camera. |
| address | address |
| city | city |
| region | Region/county |
| postcode | Zip/postal code |
| country | The country |

4.3.7. *The Metadata Server (MDS)*

The MDS automatically generates a logical topology, the FVSN of the CamNet as described in detail in section 5.4. ***The FVSN is used to identify, track, query and achieve analytics – as proposed in this research.*** As mentioned in chapter 3, the MDS may be configured to only communicate within the local network (that is, L-MDS). But it can also be configured to serve as the gateway to the CamNet (that is, G-MDS), which communicates with the city CRD to integrate and unify the CamNet with the city surveillance system. When configured as a G-MDS, it integrates with both internal cameras and the CRD. It then receives metadata from the cameras through the exposed web methods while it consumes the web methods published by the CRD.

The G-MDS is responsible for sharing the metadata with the city's CRD. Before sending the metadata with the CRD, the G-MDS first encapsulates the metadata, adding the CamNet's identity information. This way, the CRD unifies with the surveillance system only through the G-MDS – the CRD does not have direct connectivity with the individual cameras on the CamNets – any CRD communication with the CamNet goes through the MDS. The MDS is capable of establishing the camera that generated metadata through the 'camera_id' field of the metadata.

4.3.8. MDS Configuration

Similar to the privacy property in the cameras, the privacy setting of the MDS is used to decide the level of access granted to the metadata – the options are similar to those found in cameras. Table 10 shows the list of configurable properties of the MDS. These properties are used to manage the attributes of the MDS including the identification information, ownership information and location information.

Table 10: Database properties of the MDS.

| Column name | Description |
|-----------------|--|
| id | Automatically generate system id (auto-increments). |
| mds_id | A unique device serial/identity number. Similar to IMEI in phones. |
| lp_address | The network IP address |
| city_id | An identifier assigned by the CRD upon registration as a CamNet in the city. |
| mds_name | A human readable name assigned by a system admin. In a large surveillance network, this can contain name of the department, building, floor etc. |
| mds_description | A human readable description assigned by a system admin. |
| mds_crd_uri | The uri of the CRD in the city. This is obtained during the initial configuration of the MDS. |
| username | This is used to authenticate with the city CRD. |
| password | This is used to authenticate with the city CRD. |
| name | The name of the camera owner. |
| email | Owner's email address. |
| phone | Owner's phone number. |
| privacy | Signifies the highest level of access to the metadata generated by the camera. |
| address | address |
| city | city |
| region | Region/county |
| postcode | Zip/postal code |
| country | The country |
| status | This signifies the camera is active or not. |

The MDS has 4 endpoints (or APIs) are out of the box: (i) **register**, (ii) **update**, and (iii) **metadata**, and **alert**. The **register** endpoint accepts registration requests from the

CamNets, the **update** is used when updating an existing CamNet, the **metadata** endpoint accepts metadata from the camera, and the **alert** method is used to send alerts to the CRD. Based on the above database properties, an MDS object is encapsulated as a SOAP XML document before being sent to the CRD.

Figure 21 is an example SOAP XML message generated by a MDS in the experiment, showing the header, body envelopes alongside other properties that are defined in this research. It shows that the request contains a set of identity details for the requesting CamNet. The identification number was assigned to the MDS by the CRD upon registration and it must be attached to all subsequent communications between the 2 devices. This was implemented and explained further in chapter 6.

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:FVSA" xmlns:ns2="http://xml.apache.org/xml-soap" xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<SOAP-ENV:Body>
<ns1:callResponse>
<callReturn SOAP-ENC:arrayType="ns2:Map[32]" xsi:type="SOAP-ENC:Array">
<item xsi:type="SOAP-ENC:Array">
<item>
<key xsi:type="xsd:string">id</key>
<value xsi:type="xsd:string">1</value>
</item>
<item>
<key xsi:type="xsd:string">camera_id</key>
<value xsi:type="xsd:string">1</value>
</item>
<item>
<key xsi:type="xsd:string">camnet_id</key>
<value xsi:type="xsd:string">1</value>
</item>
...
</item>
</callReturn>
</ns1:callResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Figure 21: Sample SOAP XML object generated in this experiment.

4.3.9. *MDS Registration with the CRD*

The process of registering the G-MDS with the CRD involves the verification of its location and identity – A screenshot from the user interface is shown in Figure 22. The CRD will only respond if the IP address of the camera is within the same city or if the CRD is configured to accept CamNets from the address of the CamNet. If registration is successful, the CRD sends a unique identification number, which the MDS must attach to all subsequent messages to the CRD. During the CamNet configuration, the IP address of the network is retrieved and used to evaluate and verify the location of the network. This is possible since IP addresses are registered to specific geographical locations [137] [3] – if the supplied address is not consistent with the registered IP address, the user will be presented with a warning message.

The user can either accept the suggested address based of the IP address or ignore the warning. If the warning is ignored, the response sent to the CRD signifies that the provided address is inconsistent with the IP address provided by the network. However, during the availability check, as described above, this level of trust is re-investigated, and if the detected location changes but within the city of the IP address, the trust level of the network is adjusted accordingly. However, if the newly configured address is outside the city of the CRD, it initiates an action to transfer the network to a more appropriate CRD, as described in the protocol above. The registration algorithm is presented in section 6.4.

Fused Video Surveillance Architecture - the FVSA

[View RDs](#) |
 [Register a RD](#) |
 [View Camera Networks](#) |
 [Register a Camera Network](#)

REGISTRATION - CAMERA NETWORK

| | |
|--|--|
| <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> CONFIGURATION <div style="display: flex; justify-content: space-between; align-items: center;"> -- Enable City Data Sharing? -- ⬇ </div> <div style="display: flex; justify-content: space-between; align-items: center;"> -- Enable Self Managing? -- ⬇ </div> </div> <div style="border: 1px solid #ccc; padding: 5px;"> ADDRESS <div style="display: flex; justify-content: space-between; align-items: center;"> United Kingdom ⬇ </div> <div style="margin-top: 5px;"> <input style="width: 90%;" type="text" value="Address 1"/> <input style="width: 90%;" type="text" value="Address 2"/> <input style="width: 90%;" type="text" value="Address 3"/> <input style="width: 90%;" type="text" value="City"/> <input style="width: 90%;" type="text" value="Region"/> <input style="width: 90%;" type="text" value="Postcode"/> </div> </div> | <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> IDENTITY INFORMATION <div style="display: flex; justify-content: space-between; align-items: center;"> -- Number of Cameras in this Network? -- ⬇ </div> <div style="margin-top: 5px;"> <input style="width: 90%;" type="text" value="90.255.42.247"/> <input <="" div="" style="width: 90%;" type="text" value='="AUTO_GENERATED=="'/> </div> <div style="border: 1px solid #ccc; padding: 5px;"> OWNERSHIP DETAILS <div style="margin-top: 5px;"> <input style="width: 90%;" type="text" value="Name"/> <input style="width: 90%;" type="text" value="Email"/> <input style="width: 90%;" type="text" value="Phone"/> </div> </div> </div> |
| <div style="background-color: #800000; color: white; padding: 5px 20px; display: inline-block; cursor: pointer;">SUBMIT</div> | |

Figure 22: Webform for configuring the MDS.

The security of the registration process is crucial in ensuring that only appropriate MDS can register with a CRD - that is, the registering CamNet must be within the city of the CRD. In chapter 6, we present a detailed approach, implementation, and the algorithms that actualise these requirements. The main requirements are:

- To ensure that each MDS is unique and corresponds to only 1 CRD at a time.
- Each MDS registration is investigated and verified against locality and ownership criteria before granting access to send metadata.
- The personal details and the identity of the objects in the metadata will never be published on the network - only the pattern of movement, and the result of queries can be published. This is so to protect the privacy of the people in the metadata.

4.4. The City Resource Directory Server (CRD)

Figure 23 is the architecture of the CRD as implemented in this experiment. It shows that a CamNet only receives a response from a CRD (upon a successful request) – a CRD is not designed to make requests to CamNets (hence, the one-sided arrows between CRD and CamNets). However, a CRD communicates as with other CRDs within the same country, peers on P2P network so they have similar web methods and can therefore reciprocate requests/responses (as depicted with double-sided arrows). Any interaction between the CRDs must be achieved via a published endpoint while achieving all necessary QoS, access and security requirements. Its internal architecture is discussed in chapter 6 (section 6.2.2).

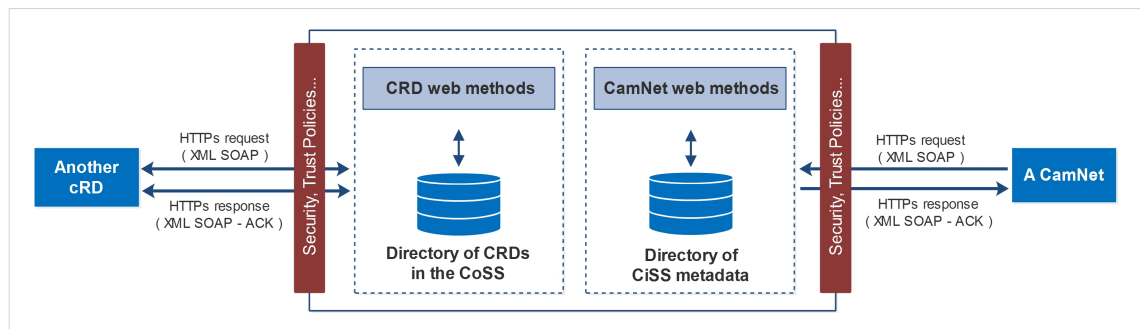


Figure 23: Architecture of the CRD

4.5. CRD Set up and Configuration

The setting up of a CRD can be accessed once the user has access to the user interface. Figure 24 shows the user interface for setting up the CRD, showing the input fields for adding the allowed cities, regions and postcodes among others. The CRD is the main directory server that manages the entire CamNets in the city - it also belongs in the country's surveillance network. It has a database that is used as a look up table service for discovering and receiving metadata from the CamNets in the city. It also has

another database, which it uses to discover other CRDs in the country’s surveillance network.

Upon submitting the registration form (as in Figure 24), the CRD saves the values in its own database. When saving the information, the CRD automatically generates unique identification information, which it uses to communicate with both CamNets and other CRDs – this is referred to as the Globally Unique Identity Number (GUIDN). These values are used in its distributed tables that were discussed in section 6.2.3. The database properties for the CRD are available in Appendix C.

CHOOSE A COUNTRY:
UNITED KINGDOM

Fused Video Surveillance Architecture - the FVSA

[View RDs](#) | [Register a RD](#) | [View Camera Networks](#) | [Register a Camera Network](#)

REGISTRATION - RESOURCE DIRECTORY (RD)

LOCATION OF RD

United Kingdom

City

Region

Postcode

ALLOWED LOCATION(S)

Allowed Cities

Allowed Regions

Allowed Postcodes

SUBMIT

Figure 24: Webform for configuring the CRD

4.6. System Hierarchies and Scopes

One of the objectives of this research is *“to achieve a system design that represents the surveillance system as in buildings, streets, cities, regions, and country”* (as

mentioned in section 1.6). The overall system hierarchy, as depicted in Figure 25 achieves this objective. In addition to the local roles within each layer of the hierarchies, the Figure also represents the user permission as described in section 4.7. There are 3 manageable layers of system hierarchies within the surveillance system in a country – (1) the ***building or street***, which is represented by the ***CamNet***. (2) The ***city***, which is represented by the CRD. (3) ***National***, which comprises the network of all the CRDs in the country. And (4) the ***country***, which is designated by the GWRD of the country – this is discussed in detail in chapter 6.

Furthermore, for any request received by a CRD, there are 3 possible scopes (within the CRD network. 1) ***local***, that is the destination object resides within the smart city of the CRD, 2) ***national***, which signifies that the destination server is outside the city of the CRD but within the same country, and 3) ***international***, which is the case when the destination object is in a different country. The access scope of the destination device is used to decide to either respond with the requested information or to forward the request to another CRD, with a more localized view of the destination object. For example, if a public safety officer in charge of city A requests to search the possible location of a car nationally, the resultant result will be limited to CamNets in city A. However, another public safety officer with the national responsibility will be presented a result that covers the activities of the same car, nationally.

The outcome of this is that an officer in any location in the world may be assigned system permission to query the system of any CamNet anywhere in the world. For example, the public safety officer in the UK may search for cameras where a car of interest has been in France, if the officer has been assigned permission to access the

French network. A combination of techniques are being suggested for improved security of the surveillance networks including the following:

1. The encryption of surveillance data and its transmission over secure protocol such as https - this is demonstrated in chapters 5 and 6.
2. Verification, validation and authorization of devices, requests and users leaving and arriving at each device - these are demonstrated in chapter 6.
3. Limiting communication to a specific segment of the network based on the role of the system user, as depicted in Figure 25.

4.7. The User

This experiment tested 2 roles for both MDS and CRD. First is the **administrator**, and second is the **power user**. On the MDS, a power user is empowered to view, query, and analyse data in the FVSN of the CamNet, while a CamNet administrator has privileges to install, administer and manager the CamNet – that is, all the components and devices on the CamNet. On the CRD, a powered user can view, query, and analyse the data from all non-private metadata from the CamNets in the city. While the city administrator is capable managing and administering all the metadata from the entire city CamNets. The permission of a user is only effective within their local network.

Table 11 shows the fields in the user details' database table while Table 12 contains the fields in the users' role table – the schema for both tables are presented in Appendix C. In the experiment, a user can only login into a system in which they have been granted permission beforehand and it is not possible to register oneself on any

other components. However in practice, permissions are designed as depicted in Figure 25, based on the user's level of visibility and access to system resources.

Table 11: User table properties

| Column name | Description |
|-------------|--|
| user_id | Globally unique camera identity number. Similar to IMEI in phones. |
| role_id | The primary role id of the user on the current network. |
| username | The absolute path of the location of the metadata video frame. |
| firstname | User first name. |
| lastname | User last name. |
| email | User email address |
| phone | User phone number. |

Table 12: User role table properties

| Column name | Description |
|-------------|---|
| role_id | The role id |
| role_name | Descriptive name of the role such as admin. |
| description | Any further description for the role. |

In Figure 25, the first level **CamNets** signify where the users of the CamNets have permissions - the CamNets users have the lowest level of permission since a CamNet user only has access to the immediate network. Second the **Cities level** where city safety officers (for example, the city police) have permission to use the public safety affairs - a city security officer has permission to operate the CRD, and any *permitted* CamNet within the city. Third the **Regions level** is not physically connected by any device, it is a conceptual layer for establishing roles to officers with administrative responsibilities over multiple cities such as a county or a state. And finally, the **Country level** where only a few officers are permitted to operate – this is the country's gateway that sets the national boundary.

4.7.1. *Security: Roles, Responsibilities and Permissions*

In Figure 25, the GWRD is the country's gateway, which also acts as the directory server for the entire CRDs across the country. However, the regional networks have only system roles, based on the membership of cities in the region. Finally CamNets belong to a specific city. The CamNets are all in the city managed by CRD1, while both CRD1 and CRD2 belong to the same region and ultimately the country GWRD knows about both RD1 and RD2. A user must first satisfy security requirements to access the system. Within the CamNet, this is achieved by a username/password combination. However, this is implemented using a services-based approach for the system calls.

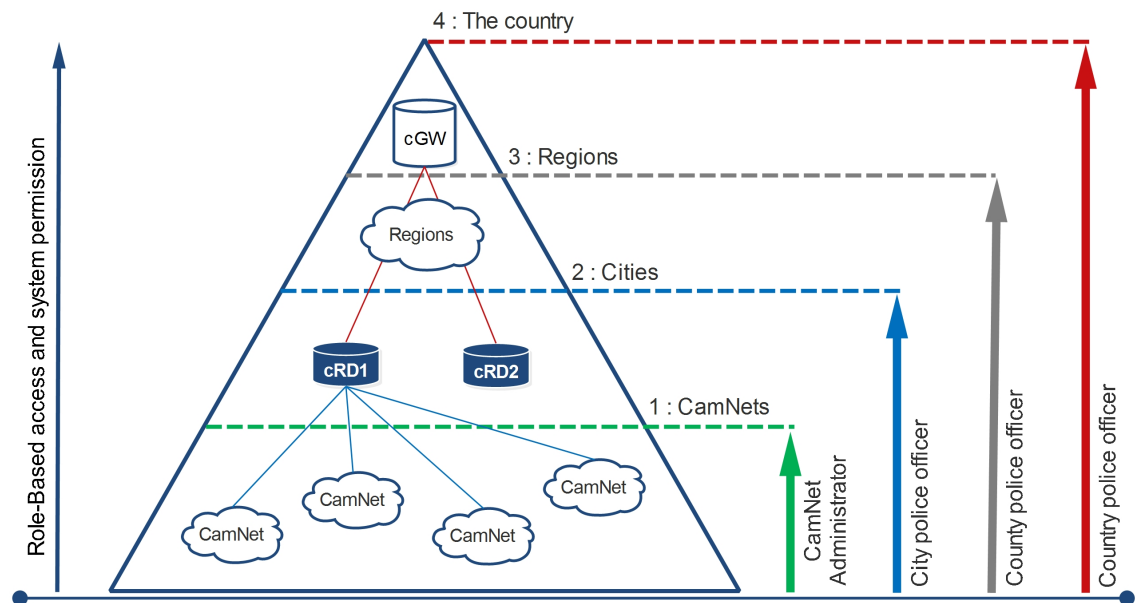


Figure 25: The surveillance network in a country is organized into scopes emulating the hierarchies in the country's geographic hierarchies.

To access the operations or features within a CamNets or a CRD, the system administrator is required to create a username/password for each user. The administrator also assigns the user role(s), which decides the users' level of access to system resources. If account creation is successful, the password is encrypted using

the industry-standard SHA-256 hashing algorithm¹. Access to system resources are granted based on user role - this is role-based access. The experiments in this research only implements 1 role, which has the highest level of permission.

For web service calls, an API username/token combination is added, which is used to authenticate the device or user depending on the message source. The authentication in the research experiments is based on Zend²'s implementation of the OAuth³.

4.8. Chapter Conclusion

In this chapter, we have discussed the design considerations and approaches for designing the components of the FVSA. We showed how we achieved the installation process, configuration process, and the properties for each component, including the metadata objects. Furthermore we presented the design and implementation details of the system users, including the implementation of role-based permission.

Two research questions were answered - RQ1 and RQ2 (see section 1.5). The design strategies were targeted at creating and managing independent surveillance systems that are capable of interacting with each other, through published endpoints, while exchanging SOAP-based XML metadata. The CamNets are capable of sending metadata to a single repository where the unified metadata is used to compute a logical network from the metadata sourced across the city, which also constitutes the surveillance system of the country.

¹ See - <https://en.wikipedia.org/wiki/SHA-2>

² <https://framework.zend.com/>

³ <https://oauth.net/>

5. An Implementation of the CamNet

Research is what I'm doing when I don't know what I'm doing.

- Wernher von Braun

5.1. Overview

In chapter 4 (the last chapter), we discussed the construction of the components in this chapter's experiments. The chapter provided the basis for the design strategies, and configuration approach for the components, and the representation of the data, which are employed in this chapter to implement and run the simulation projects. It gave an account of how the components of the FVSA are represented in the experiments, and how each module contributes to the overall operation of the FVSA.

The activities in the current chapter involve the development of an experimental project that simulates a CamNet. Using the simulation, we capture and generate metadata across the CamNet based on some objects that were randomly introduced to randomly traverse the CamNet. Using the metadata generated from the simulation, we demonstrate how the FVSN is computed from the metadata, and we explain how an object is detected, and tracked. Ultimately, we provide an approach to querying the metadata in the FVSN.

While achieving the above, we also provide answers to two research questions in this research. First, the primary question in **RQ1: *"Is it technically achievable to analyse and explore a video surveillance system without the full system access to the surveillance network cameras and data?"***. Second, **RQ4: *"To what degree of accuracy can we systematically and analytically predict the location of an object such as a person, across a well-connected camera cluster in a smart city?"***.

5.2. Simulation Strategy

This chapter involve the development and implementation of an experimental project, which simulates the processes and results that are obtainable on a CamNet, as proposed in this research. The design, format, and representation of the components in the simulation are based on the designs presented in chapter 4. As described in chapter 4, each component is an encapsulation of database properties, which are instantiated and encapsulated as reusable objects. The project components are configured to achieve the following system features and capabilities.

1. A 2-dimensional grid was constructed using Matlab, which represents the space covered by the surveillance cameras. All infrastructures and components of the simulation are implemented in relation to the grid. An example grid is shown in Figure 26, which contains 100 x 100 cells, where each cell is identified by its x,y coordinates.
1. Within the dimension of the grid, several cells are identified as nodes with links introduced between them.
2. Also within the dimension of the grid, several cells are randomly identified as cameras.
3. People are then introduced to randomly walk the grid where each object starts on a node and traverses the grid to the pre-assigned end node.
4. People movement is simulated as a relocation to a different cell, based on the shortest path algorithm computed between the start and end node.

5. Whenever a person is located at a cell, which has been marked as a camera, that object is recorded into the metadata table. Other movements that do not land on a camera are ignored.
6. The unsupervised topology-learning algorithm runs on the entries in the metadata database – the protocol uses ranking protocol to decide the accuracy of the re-identification of objects.
7. The ability to query the surveillance system is achieved by deriving a network topology based on the metadata database.
8. It is possible to identify the camera that generates a metadata object through the camera_id field of the metadata.

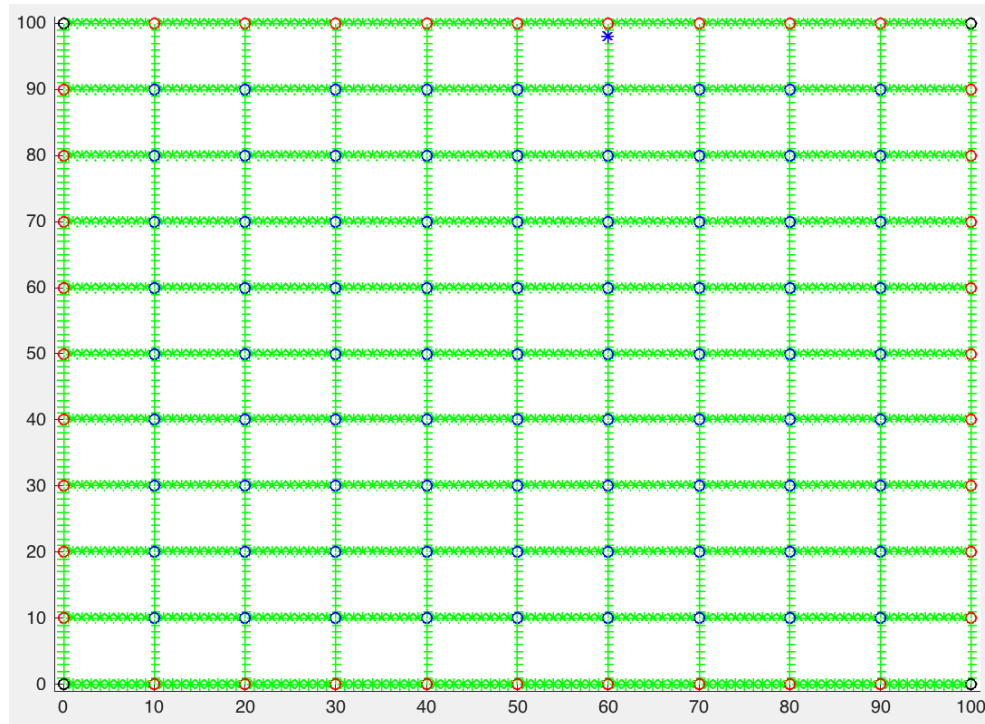


Figure 26: The simulation grid in Matlab

Figure 26 depicts the spatial dimension of the example grid - a grid of 100 x 100 cells, where each cell in the grid represents an area where a person may be located and identified. In the example grid, there are a total of 10000 cells (that is, 100 x 100 cells).

5.2.1. *Experiment Set up*

The versions of the system and application implemented in this research are presented in **Table 13** - Matlab is the main development environment used in this chapter. The Bioinformatics toolbox is used to plot the graph of the network, while the database toolbox is used to connect to the MySQL database, which defines each component and parameter in the experiments.

Table 13: System requirement – experiment 1

| Parameter | Value |
|------------------------|--|
| Simulation environment | Matlab & Simulink |
| Matlab Toolboxes | Bioinformatics Toolbox Computer Vision System Toolbox Curve Fitting Toolbox Database Toolbox Image Acquisition Toolbox Mapping Toolbox Simulink 3D Animation |
| Hardware & OS | Max OS X El Capitan; MacBook Pro 15"; 2.5 Ghz Intel Core i7; 16Gb DDR3 |
| Database | MySQL: 5.5.47 MySQL Community Server |

5.2.2. *Assumptions and Considerations*

- Each person introduced to the surveillance network is pre-assigned the following attributes: top colour, bottom colour, height, and system ID. These are used to achieve re-identification and tracking on the network.
- The identification features of 95% of the people introduced to the network are known beforehand and this value contributes to the re-identification process. However, 5% of the people introduced are false so these people do not contribute any activity to the processes of the system.

- The starting point and end point of each journey is pre-assigned at the beginning of the journey by randomly choosing nodes within the grid. However, the analysis and computation of a journey may start at any point during the journey – since the computation only depends on when a camera first spots the person.
- The algorithm that predicts the location of people on the network assumes that each person travels through the most popular paths (between any 2 locations).
- Each journey starts and ends within the grid – no activity outside the grid is considered in any aspect of the experiment.
- No person completes a loop. Any journey resulting in a loop is ignored when computing the graph - this implies that only hollow matrices are allowed in this simulation.
- For each metadata generated by the cameras, only one person is identified and only this one person is represented in the metadata object.
- The Field on view of the cameras does not overlap so only one camera captures the same person at the same time.
- The values of the following attributes of a traveller are known at the beginning of the journey – colour, height, speed, total time taken to complete the journey and the other features in Figure 27.

5.2.3. *Person and Journey Simulation*

Figure 27 is the database schema that represents the journey object in the database, in which some fields represent the person who completes the journey. These fields are crucial to the success of the person re-identification process, which is discussed in

section 5.3 – they are '**traveller_id**', the system generated ID. The '**top_colour**', which represents the colour of the top half of the person, The '**bottom_colour**', which represents the colour of the bottom half of the person, then the '**height**', and then the '**texture**', which is any other characteristic of the person.

In addition to the identifying properties above, the '**startLocation**', is used to represent the first visited cell. The '**path**', which is a list of all the nodes that the person visited, and the '**finishLocation**', cell where the object last appeared. Then there is the '**distance**', which is computed from the time_taken and velocity, the 'time_taken', which includes the total period for completing the journey. The 'start_timestamp', which is the time journey started.



| fvsa_matlab_journey | |
|---------------------|--------------|
| id | INT(10) |
| traveller_id | INT(10) |
| top_colour | VARCHAR(128) |
| bottom_colour | VARCHAR(128) |
| height | INT(10) |
| texture | VARCHAR(128) |
| travelerColorName | VARCHAR(28) |
| startLocation | VARCHAR(128) |
| finishLocation | VARCHAR(128) |
| path | TEXT |
| distance | DOUBLE(16,1) |
| velocity | DOUBLE(16,1) |
| start_timestamp | VARCHAR(128) |
| time_taken | DOUBLE(16,1) |
| Indexes | |

Figure 27: The database table that represents objects travelling across the network.

5.3. The Simulation Model of a CamNet

Table 14 presents the main parameters in the experiments. A 2-dimensional grid is used to model the spatial dimension occupied by the network of cameras in which the cells simulates features that are of interest in this research. These include the network nodes, where a subset of the nodes represent the cameras, start point, and end points of roads across the network, etc. A 100 x 100 grid was assumed. Within the grid, each object traversing the network has a maximum 10000 chances of being located throughout the simulation (if each cell was visited once).

Table 14: Experimental parameters – The CamNet

| Parameter | Value |
|--|-----------|
| Grid dimension | 100 x 100 |
| No of cameras | 12 |
| No of edges | 52 |
| Total number of simulated journeys | 600 |
| Number of people captured by the cameras | 62 |

Figure 28 is an example digraph output by the simulation based on the parameters in Table 14. The coloured lines (that is, red, green, blue, and black lines) correspond to the top colours of the last person that traversed the path. However, the grey lines represent real paths (roads) linking the location of the cameras. The weights on the grey lines represent the distances between the edges while the weights on the coloured lines represents the popularity of the links (that is, the number of times that a person has been re-identified traversing the link).

Further, the grey lines in Figure 28 represent the graph based on the physical network while the coloured lines represent the paths that may be included in the logical network, which is generated based on the pattern that objects traverse the network. The logical network is the basis for unification and integration in this research. A detailed account of the processes that computes the FVSN is presented in section 5.4.

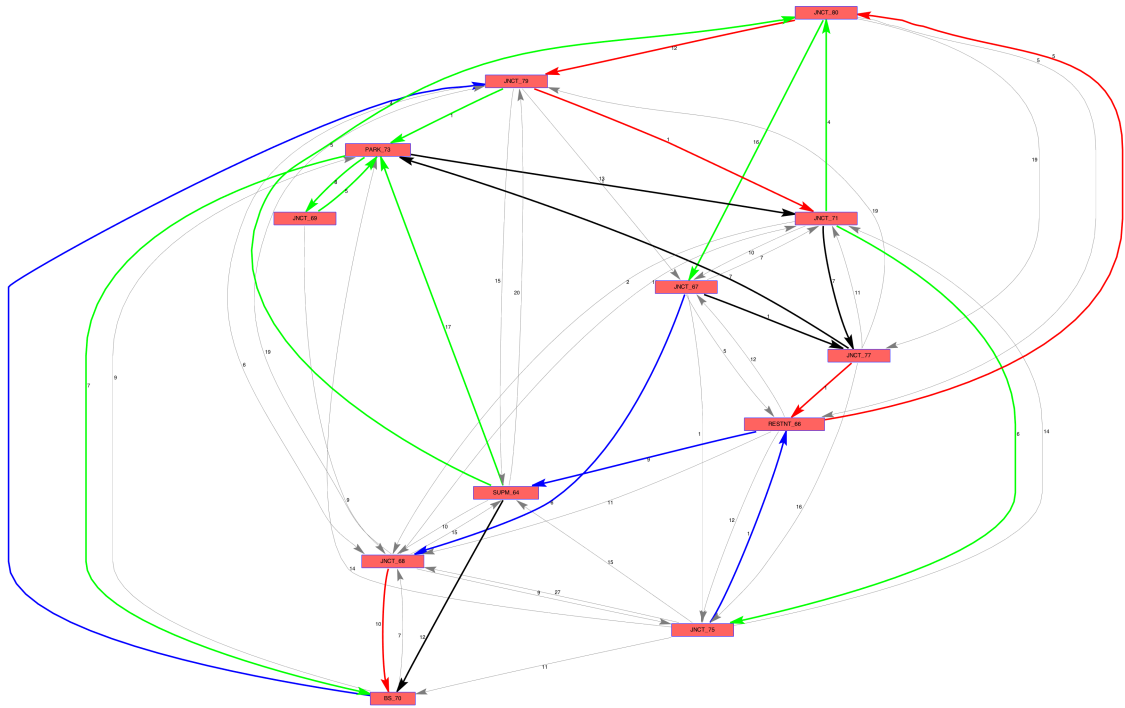


Figure 28: Matlab plot of a CamNet comprising 12 cameras.

5.4. Unsupervised Topology Learning

This section contains information about the experiment's implementation of person identification and re-identification (Re-ID) of people across the network, and the derivation of the logical network, the Fused Video Surveillance Network (FVSN). The FVSN is the principal requirement to achieving the unification goals of this research – it is the query-able network protocol computed tracking people across the surveillance system.

5.4.1. *Person Re-Identification*

To achieve an imperfect matching scheme, which is as close as possible to the current level of research in re-id, we introduce a level of unpredictability and inaccuracy into the matching simulation. Based on the methods employed in similar research [115] [116], we consider the non-contextual methods (colour-based feature extraction) in which we assign and extract 6 descriptors on each simulated person.

We divide a person into upper and bottom horizontal halves so a different colour is assigned to each half of the person's body. The 5 simulated descriptors are – **top_colour** (20%), **bottom_colour** (20%), texture (10%), height (10%), and system ID (30%). The percentage at the front of each descriptor signifies the maximum that the descriptor can account for when matching a person. Basically, if a match is found for all 5 descriptors, then there is a 100% match – that is rank 1. However, a partial match will be used to rank the matches for example, if a match is only found for the top colour of a person - that is, 20% match.

The re-id process in this research is described with the aid of Figure 29. The metadata database represents the **gallery** while newly added metadata object is the **probe**. Once the descriptors are extracted, a search is conducted on the metadata database for a match. If a match is found, the system returns the first 20 results sorted by the highest match - this represents the rank of the images. If a match is found, a link is created between the two cameras that captured each metadata, while the rank denotes the reliability of the link – this information is saved in the FVSN database, which is used to

generate the FVSN topology (see section 5.4.4). The algorithm in section 5.4.2 shows the sequence of processing the match.

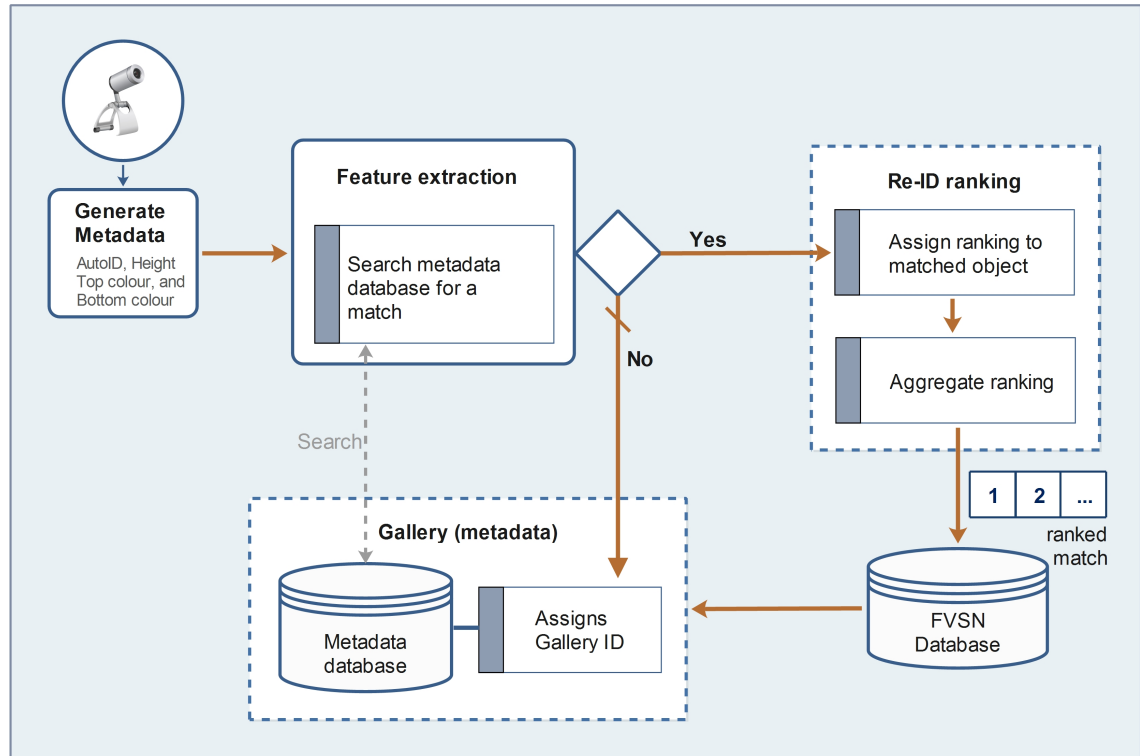


Figure 29: Re-ID process in this research

5.4.2. Algorithms 5A: Re-ID ranking

The algorithm below accepts the gallery image and the probe, and then it compares each descriptor of the 2 objects to establish the proportion of match (if any). Line 14 shows that 2 descriptors are required to match to accept there is a match - otherwise no match is concluded. If there is a match, the highest value of the match is recorded into the FVSN database table, including the ranking of the match.

Constants: top_color (tc) = 20, bottom_color (bc) = 20, texture (tx) = 10, height (hg) = 10, personality (pn) = 10, and id = 30

Action: match and rank gO to pO, based on similarity of their constants

Output: percentage match for pO to gO. **Input:** probe **pO**, gallery **gO**

Begin

1. *Foreach image in gO*
2. *set rank=0; count=0; tc=20; bc=20; tx=10; hg=10; pn=10; id=30;*
3. *if gO.tc == pO.tc then*
4. *rank += 20 AND count += 1*
5. *if gO.bc == pO.bc then*
6. *rank += 20 AND count += 1*
7. *if gO.tx == pO.tx then*
8. *rank += 10 AND count += 1*
9. *if gO.hg == pO.hg then*
10. *rank += 10 AND count += 1*
11. *if gO.id == pO.id then*
12. *rank += 30 AND count += 1*
13. *if gO.tc == pO.tc then*
14. *rank += 20 AND count += 1*
15. *if gO.pn == pO.pn then*
16. *rank += 10 AND count += 1*
17. *if count > 1*
18. *save pO in the FVSN database;*
19. *return rank*
20. *else*
21. *return 0*
22. *save pO into metadata database*
23. *calculate rank – that is, $(rank/count*100) = rank\%$*
24. *save path, rank in the FVSN database*
25. *End foreach*

End

5.4.3. Formulation of Tracking

For each person that appears on a cell designated as a camera, the journey details and person's attributes are recorded into the metadata database table. As the cameras capture people, the pattern of travel is used to generate a logical network (that is, the FVSN), using the frequency of re-identification to add direction and weight to the

edges of the FVSN. Note the frequency is not based on unique persons - the weights on the FVSN links are a sum of all the detected journeys. For example, based on Figure 28, the adjacency matrix in Table 15 is computed. Element **(a,b)** of the matrix shows that people were re-identified 5 times between cameras **a** and **b**.

Table 15: (a) Adjacency matrix generated by the objects identified across the graph in Figure 28. (b) Sparse form of the matrix in (15a)

| (15a) | | | | | | | | | | | | | (15b) | |
|-------|----|---|---|----|---|----|---|----|---|---|----|---|--------------|--------|
| | a | b | c | d | e | f | g | h | i | j | k | l | Ordered Pair | Weight |
| a | 0 | 5 | 0 | 7 | 0 | 0 | 0 | 0 | 0 | 0 | 13 | 0 | a,b | 5 |
| b | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | a,d | 7 |
| c | 0 | 0 | 0 | 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | a,k | 13 |
| d | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | b,a | 8 |
| e | 17 | 0 | 0 | 12 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | c,d | 10 |
| f | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | d,f | 4 |
| g | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | e,a | 17 |
| h | 0 | 0 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | e,d | 12 |
| i | 0 | 0 | 0 | 0 | 0 | 12 | 0 | 16 | 0 | 0 | 0 | 0 | e,i | 5 |
| j | 0 | 0 | 0 | 0 | 9 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | f,a | 1 |
| k | 0 | 0 | 0 | 0 | 0 | 0 | 6 | 0 | 4 | 0 | 0 | 7 | f,k | 1 |
| l | 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | g,i | 1 |

| Ordered Pair | Weight |
|--------------|--------|
| h,c | 6 |
| i,f | 12 |
| i,h | 16 |
| j,e | 9 |
| j,i | 5 |
| k,g | 6 |
| k,i | 4 |
| k,l | 7 |
| l,a | 7 |
| l,j | 1 |

To compute the FVSN, a sparse form of the matrix is generated, which Matlab uses to plot a graph of the FVSN. For example, Table 15(b) is the sparse form of the adjacency matrix in Table 15(a) - it includes only the non-zero elements of the adjacency matrix.

5.4.4. *The Fused Video Surveillance Network (FVSN)*

Based on the matrix presented in Table 15(a) and the re-id process, the FVSN topology is computed as in Figure 30 in which the FVSN is a subgraph of the full network (Figure

28 in this case). The FVSN is the network used to achieve 2 goals of this research – first, the unification of independent networks and second, the ability to query a surveillance network.

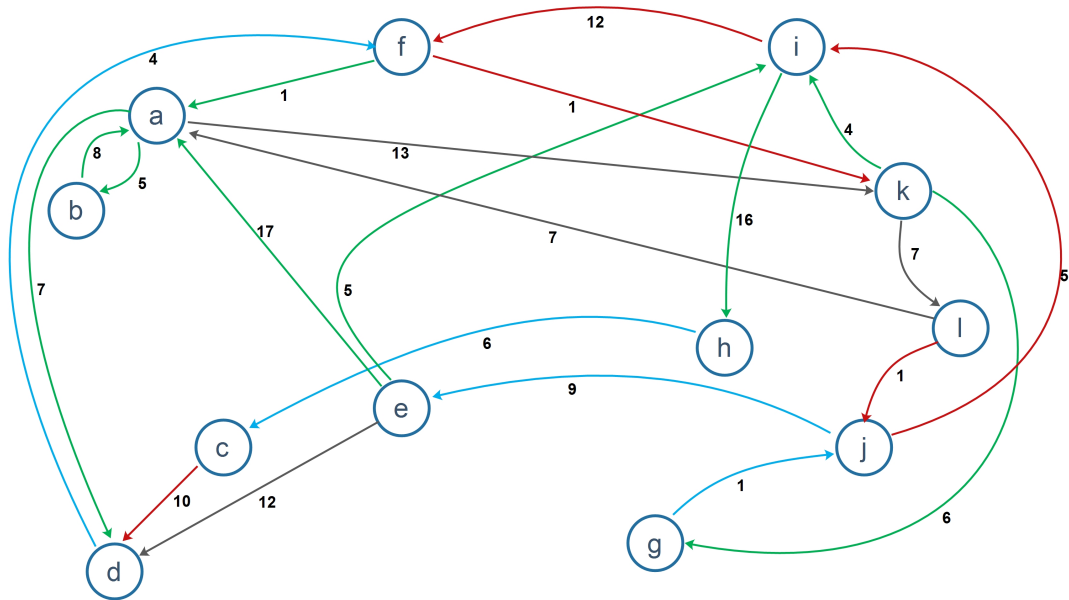


Figure 30: The computed FVSN based on Table 15(b)

5.4.5. The Routing Protocol and System Query

Any query to the FVSN is computed on the FVSN database in which the routing protocol is computed dynamically per query, so there is no pre-computed routing table. The routing protocol is computed by calculating the most popular next link in relation to the final destination. For example, in Figure 30, to travel from camera l to d (with no timeframe set), the protocol will identify and evaluate 2 paths:

1. $(l, a), (a, d)$ - total popularity of path = 14
2. $(l, j), (j, e), (e, d)$ – total popularity of path = 22

In the above example, path 2 is chosen because it is more popular, with a popularity score of 22 compared to 14 in path 1. The protocol is an adaptation of the dijkstra's algorithm in [62], in which it evaluates the next node based on the links with the highest weight rather than the least cost. Based on the database properties of the metadata presented in section 4.3.6, we can search for an object in the database. For example, to predict the current location of a person who was instantiated as 'traveller', and spotted at node 'a' in the last 10 minutes) – we run the following MySQL query (for rank 5. To search for rank 1 match, replace the OR with AND):

```
query = "select * from `fvsn` WHERE
`traveller_id` = traveller.id OR
`top_colour` = traveller.top_colour OR
`bottom_colour` = traveller.bottom_colour OR
`height` = traveller.height OR
`texture` = traveller.texture AND
`startLocation` = 'a' AND
dateadd(minute, -10, start_timestamp ())
ORDER By rank DESC"
```

This query will return all journeys in the last 10 minutes where the traveller id or any of the other property matches the values passed. The result is also ordered by ranking so we can select the 4 highest ranked, for example. Equally, this presents the level of accuracy of the match such as 100% match for the person. Running above example query on a database of a city CRD will return a match for the traveller, irrespective of the owner of the camera. To establish the owner of the camera, we add a join to the city's metadata table to retrieve the CamNet details.

5.5. Problem Formulation

The initial experiment simulates an area of 1Km^2 (that is, 1000m^2), which is represented in Matlab by a 1000×1000 grid – so that a cell represents 1-meter square on ground. Then 100 cameras, 1000 people, and 1000 journeys were randomly added to the grid. The simulation was repeated 10 times to increase the reliability and stability of the results. And to achieve a reasonable level of consistency and traceability, the size of the parameters assigned to each simulation is left the same.

In order to evaluate the reliability of the experiments under different conditions, the parameters were varied to simulate differences in persons' appearance due to variation in colour of clothing etc. In particular, this effect of the following conditions were observed - different size of crowd, different number of cameras, and the period in which a person is being tracked. The option from which the system choses parameters at random are presented in Table 16, showing 10 options each for the top colour, bottom colour, heights, texture, and personality. However the cameras, and the dimensions of the area covered by the experiment are the same through the simulations.

Table 16: CamNet's simulation parameters

| Parameters | Value |
|----------------|---|
| Top colours | Brown, Black, White, Red, Blue, Black, Grey, Tan, Pink, Red |
| Bottom colours | Yellow, Cyan, Olive, Gold, Black, Sand, Maroon, Sand, Maroon Blue |
| Heights | 2, 4, 5, 6, 3, 6, 1, 1, 1, 1 |
| Texture | Adidas, Nike, Hugo, Tommy, BP, RayBan, Pebble, Silk, Sony, Apple |
| Personality | 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 |

5.6. Predicting the Location of a Person

This research aims to use the operation of the algorithm in section 5.6.1 to predict the location of a person, who was once identified on the CamNet. To achieve this, we use the query in section 5.4.5 to search the FVSN database and then compute the most likely 4 cameras to locate the person – based on the search result ranking. The algorithm (which is presented in section 5.6.1) accepts 3 parameters – first, the system id of the person (in practice, this is one of the person’s descriptors such as their clothing colour), second is initial to final time, and third is either the period of time or the exact timestamp after which to predict the person’s location. For example, we could be interested in predicting the location of a person with id = 24, from 1pm to 2pm. Alternatively we could be interested in their location 10 minutes after 1pm.

The algorithm used for computing the location of a person evaluates each path from camera to camera to decide which path is most popular to the destinations that are within the time frame to search for a person. This algorithm is presented below in section 5.6.1. The algorithm accepts a weighted digraph (that is, the FVSN), period in consideration, and then the start node. The FVSN nodes correspond to cameras while paths are derived from summation of the successful re-id between cameras, as described in section 5.4.4 - the algorithm is an adaptation of the dijkstra’s algorithm in [62], as described in section 5.4.5.

5.6.1. *Algorithms 5B: Predicting the Location of a Person*

Action: traverse a weighted digraph (V, E) , and find the 4 most popular paths from vertex A to any vertex v where the journey from A to v is completed within time $t[v]$.

Input: the Graph (V, E) , time P .

Process: For any 2 vertices u and v , $p(u, v)$ is the popularity of the arc uv , and $PATHTO(v)$ lists the vertices in the current most popular path A to v .

Output: the 4 most likely cameras the person is located based on ranks 1 to 4.

Begin

```

1. Foreach  $v \in V$  do
2.   Begin
3.      $t[v] := p(A, v);$ 
4.      $PATHTO(v) := A;$ 
5.   End
6. Mark vertex  $A;$ 
7. for  $rank = 0; rank < 5; rank++$  do
8.   While unmarked vertices remain do
9.     Begin
10.     $u :=$  unmarked vertex whose popularity from  $A$  is max;
11.    Mark vertex  $u;$ 
12.    foreach unmarked vertex  $v$  with  $uv \in E$  do
13.      begin
14.         $t^! := t[u] + p(u, v)$ 
15.        if  $t^! > t[v]$  then
16.          begin
17.             $t[v] := t^!;$ 
18.             $PATHTO(v)rank := PATHTO(u), v;$ 
19.          end
20.        end
21.      end
22.    end

```

End

5.7. Results

Based on the data obtained from the FVSN, the accuracy of predicting the location of a person was investigated under various conditions. First, it was based on **re-identification timeframes (that is, the time since the person was spotted)**. The **short-period Re-ID** is assumed when the timeframe is within 15 minutes. Then it was based

on the **long-period Re-ID (L-Re-ID)**, which is assumed for any query to locate a person after a period over 15 minutes - the results are presented in section 5.7.1.

A second criterion employed to establish the reliability of this research's proposal is the ***proportion of people to the cameras***. We conducted 10 simulations starting with 20 cameras and increasing at the rate of 20 cameras until 200 cameras, while the number of people remained at 500 in each simulation - the results are presented in section 5.7.2.

Next the impact of the ***proportion of cameras to people*** was evaluated. In this case we kept the number of cameras at 100 while we repeat the sets of experiment with 500, 1000, 1500, 2000, and 2500 people – the result of the experiments are presented in section 5.7.3.

The fourth set of experiments (which are available in section 5.7.4) showed the impact of the ***number of descriptors*** used in the re-identification process. This experiment provided an insight into how the uniqueness of being tracked influences the reliability of tracking a person in this experiment. Initially, 2 parameters were initially introduced, and then the parameters were increased by 1 until 6.

5.7.1. The Impact of travel period on person re-id

Figures 31 (a) and (b) show the result of matches obtained when computing the location of a person after a certain period of time has lapsed. For this matches, Figure 31 (a) indicates that the match rate declines over time after achieving highest match in

the short-term. It is observed in (b) that the short period ranking (that is, the first 10 minutes) achieved a maximum of 61.8% match - for rank 1.

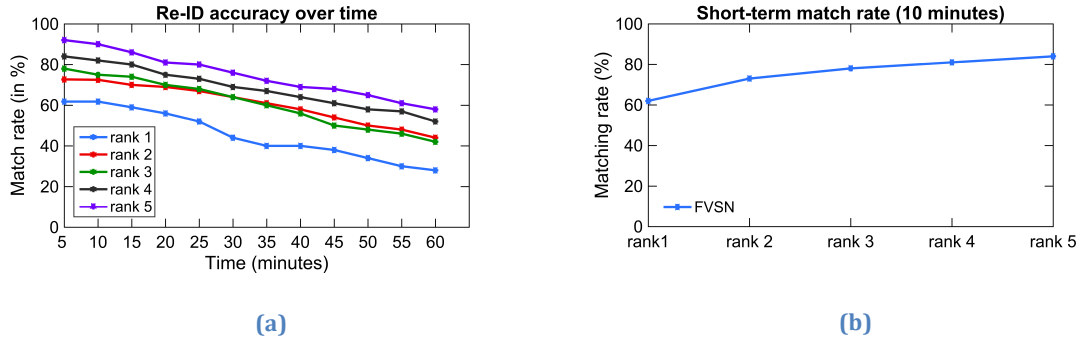


Figure 31: Period varying match rate

5.7.2. The Impact of Camera Density

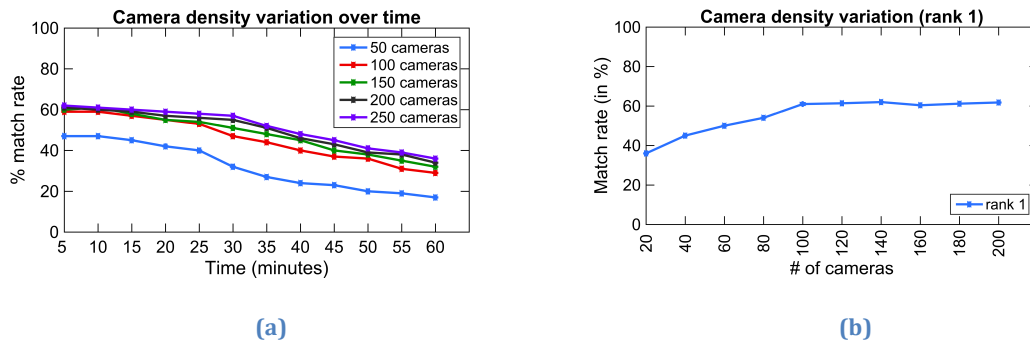


Figure 32: matching rate under varying camera density.

From Figure 32, it can be observed that the match improves with the addition of more cameras. Figure 32 (b) shows 5 rank-1 results as the number of cameras was increased from 20 to 200. A sharp improved was recorded between 60 cameras and 100 cameras. However, the improvement in match rate 'stalled' with increased cameras number above 100. In other words, increase in camera density does not significantly improve match rate beyond 100 cameras in a 1Km^2 space (or a density ratio of 1 camera to 10 metres square). Based on this observation, this author is inclined to

recommend a camera density between 80 and 100 cameras per 1000m² to deliver optimum performance.

5.7.3. *The Impact of Crowd Density*

Based on the outcome of varying the number of people travelling across the network, the result in Figure 33 (a) and (b) shows that the matching rate declines as more people are being introduced on the network. 32 (a) depict the rate of decline per number of people in the network at any time in which the rate of match is constantly higher for less number of people on the network over the whole 60 minutes of the simulation. This result confirms our expectation that the more the number of people, the more difficult is to re-id an individual. In addition, the result in 33 (b) shows that more people in the same space leads to better chance of inaccurate match. The rise in the match rate for rank 2 through rank 5 supports this behaviour.

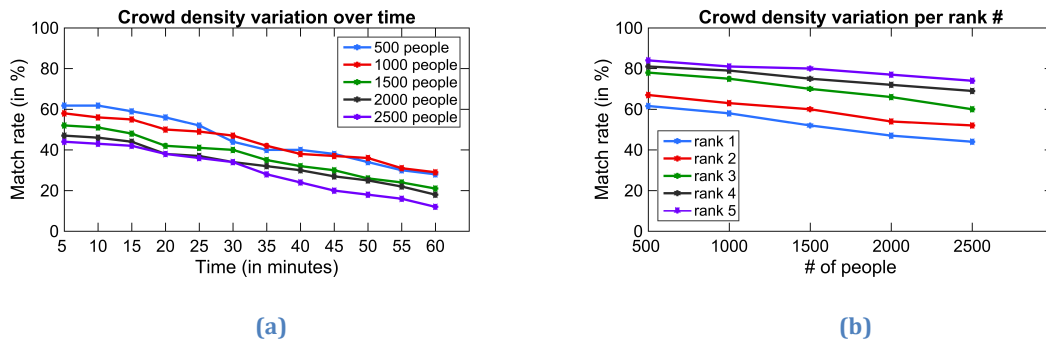


Figure 33: matching rate under varying crowd density.

5.7.4. *Uniqueness of Persons*

This research is interested in evaluating the effect of the uniqueness of the people being tracked on the reliability of the result. First we considered only 2 descriptors by

matching our ‘person of interest’ to only 2 their system ID and top colour, and then we matched to 3 descriptors, until 6. The result, which is presented in Figure 34 revealed an interesting finding in that the match rate of our result dropped significantly - from 78% to 61.8% for 2 and 6 descriptors respectively. Since the accuracy decreases with increase in parameters, it can be concluded that more descriptors may further reduce the match rate since the progression downwards is still continuing – however we did not test this further due to the limitation of computing resources.

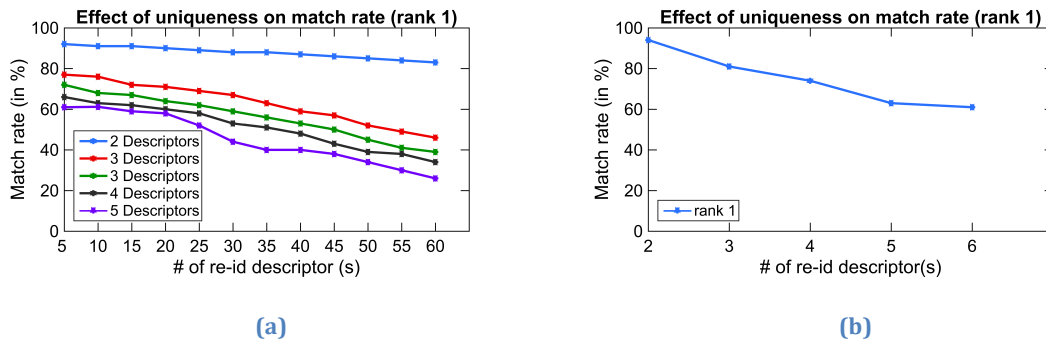


Figure 34: impact of person uniqueness on match rate

5.8. Accuracy of Predicting a Person's Location

Table 17: The degree of accuracy achievable over time

| Timeframe (minutes) | Rank-1 accuracy (%) | God's eye accuracy (%) | Prediction Accuracy (%) |
|---------------------|---------------------|------------------------|-------------------------|
| 10 | 61.8 | 84.7 | 78 |
| 20 | 56 | 85 | 69 |
| 30 | 44 | 88 | 5 |
| 40 | 40 | 87 | 59 |
| 50 | 34 | 92 | 52 |
| 60 | 28 | 94 | 30 |

The accuracy of the ‘God’s eye’ is computed by calculating the percentage of the total number of people on the network divided by the total number of people accounted for at that time. For example, we can calculate the accuracy of God’s view after 10 minutes where 500 people were originally added to traverse the network but 300 were

found by cameras across the network within the 10-minute period. This is calculated as $(300/500)\% = 60\%$.

To estimate the accuracy of a predicted location, we first calculate the percentage of the people who achieved rank 1 match within the period of interest compared with the total number of people in the crowd. For example, when 50% of people achieved rank 1 within 10 minutes where 500 people were within the network, this is calculated by $(50/500)\%$ - that is 10%. Table 17 (above) is a list of computed values of God's view and corresponding values of rank 1. From the Table, we are able to arrive at 3 finalities:

1. The highest degree of accuracy achieved is 25%.
2. The best accuracy is achieved at shorter period of time while the accuracy gets worse over time, degrading to 10% by the hour.
3. Although the God's view propels towards 100% over time, it did not reach 100%. This confirms our suspicion that someone once spotted by a camera may not be spotted again within the network since they may follow a path where no camera existed.

5.9. Validation and Testing

The work in this chapter is based on modelling and simulation. It is therefore natural to expect that system verification; validation and testing of the results are achieved through proven practices in simulation model verification and validation. For the scope of this research, we define model verification as a procedure for ensuring that processes and their implementation on our simulation model are as expected or

‘correct’ [138]. We define model validation as a concept that our model possesses a satisfactory range of accuracy consistent with its intended behaviour, when used within its domain of applicability [138] [139].

5.9.1. *Comparison of Predicted Location with God’s Eye*

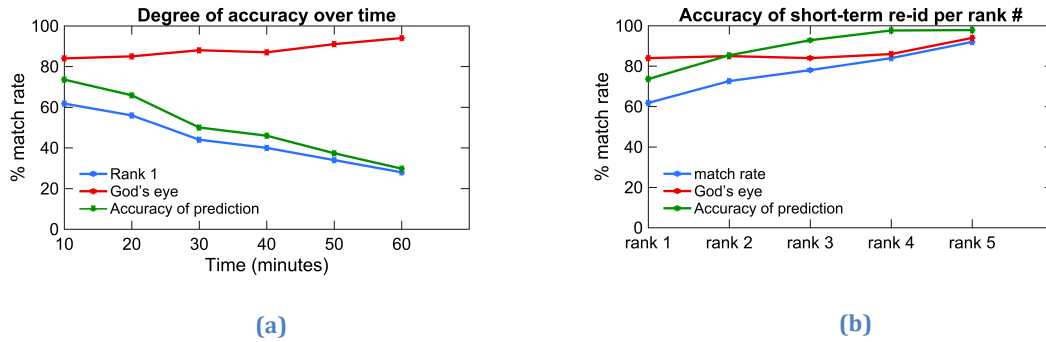


Figure 35: Comparison of predicted location of a person with their real location

For this reason, we included a means for identifying any person simulated across the network by assigning them a system ID, which can be used to track them at 100% accuracy. However, the real location information is only recorded for validation and testing purpose. The validation is achieved by comparing the real location of a person to the predicted location, computed using the algorithm in section 6.5.1

We compute the location of 10 objects within the best accuracy timeframe (that is, 10 minutes). We then compared the results of our prediction with the real location of those people after the same timeframe – the real location of people are known in this simulation, using their system ID. The result is shown in Figure 35 in which the performance of the prediction seems to decrease with time and it would appear that the longer the period requiring prediction, the lower the chance of accuracy of the

result. Based on this result, the author will consider this prediction to be useable for up to maximum of 10 minutes and probable unreliable after a period of 30 minutes.

5.9.2. *Comparison with the state of the art*

As of the time of writing this thesis, the highest rank achieved (rank 1) for person re-identification is 53.5% (based on the VIPeR Dataset) and 82.5 for rank 5 – these were achieved by the same author in [115]. Although their experiment was carried out on some openly available images sets, which have been calibrated to support research, our model is completely simulated with similar calibration but more unexpected inputs since our parameters are randomly assigned. Our result compares relatively well, showing an improvement of 8.3% in rank 1 (61.8%) and 84.7 compared to 82.6 for rank 5. Table 18 presents a comparison of our result with the 5 best results found in the literature, based on the VIPeR Dataset. It is noted however that our simulation is simulated based on randomly generated data, while those approaches were based on primed images.

Table 18: FVSA Ranking Compared with the state of the art – available at the SSIG website⁴

| Rank | Rank 1 (%) | Rank 5 (%) |
|------------------------------|------------|------------|
| SCSP [115] | 53.5 | 82.6 |
| Kernel X-CRC [140] | 51.6 | 80.8 |
| FNN [113] | 51.1 | 81 |
| Cheng et al. [116] | 47.8 | 74.7 |
| Paisitkriangkrai et al [117] | 47.8 | 77.9 |

⁴ <http://www.ssig.dcc.ufmg.br/reid-results/>

5.10. Chapter Conclusion

In this chapter, we discussed the operation the TA algorithms showing how we achieve the FVSN, which is a logical weighted digraph based on the pattern of journeys within the network. We stated the assumptions and approaches to creating the FVSN database, which accepts standard SQL queries, and presents matched results based on the features and properties of the people captured by cameras in an area. This includes a demonstration of how our protocol uses the knowledge of the physical network to compute a logical weighted digraph based on the pattern of journeys in the network.

This chapter provide answers to 2 research questions that were raised in section 1.5.

First, the primary question in **RQ1: “*Is it technically achievable to analyse and explore a video surveillance system without the full system access to the surveillance network cameras and data?*”**. Secondly this chapter provided answers to **RQ4: “*To what degree of accuracy can we systematically and analytically predict the location of an object such as a person, across a well-connected camera cluster in a smart city?*”**.

Our objective is to investigate the degree of accuracy of the approach for estimating the location of a person on the network. Based on the matches of rank 1, this thesis achieved a degree accuracy of 59% in short period re-id - see sections 5.8 and 5.9.1. However, the accuracy was as low as 28% at the 60th minute. Our results indicate that this solution is not accurate enough for predicting the precise location of someone within a network. However, it is observed that this solution could be used to eliminate

cameras that don't need to be investigated when searching for someone within a camera surveillance system, in the short-term re-id, such as within a 10-minute period.

In this chapter it was also observed that increasing the number of cameras monitoring an area reaches saturation point between 80 and 100 cameras per 1Km^2 for our model.

6. A Scalable Resource Directory for the Globally Unified Video Surveillance Network

The greatest part of a writer's time is spent in reading, in order to write: a man will turn over half a library to make one book.

- Samuel Johnson

6.1. Overview

In the last chapter, we discussed our experiment in which we designed and simulated a CamNet. Using our experimental project, we captured and generated metadata across the network by introducing objects that simulated people traveling the CamNet. Based on the pattern the objects traverse the network, we computed the CamNet’s logical network – the FVSN. The FVSN is the graph that we used to predict the location of objects, which was presented in chapter 4.

In this chapter, we build on the design effort in chapter 4, and the implementation effort in chapter 5 to simulate a network made up of multiple CRDs. That is, the surveillance system of a country (or CoSS). We now show how the experiment supports the hierarchical design that was introduced in chapter 4. In the design, the overall global surveillance system emulates the geopolitical structure of the modern world – as in buildings, street, cities, and country (see section 2.2).

We demonstrate how the independent CRDs integrate with each other to achieve the objective(s) through the published APIs. Furthermore, we present the overall system architecture, service discovery process, and the lookup service, amongst others, which the CRDs use to discover one another, during communication. Finally, we answer the research question in **RQ2: “Can we query surveillance networks that belong to multiple independent administrative owners?”**.

6.2. System Architecture

The hierarchies in the architecture diagram presented in Figure 36 depicts the research's representation of the main geopolitical entities – that is, buildings, streets, cities, and countries. The top ring (which consists of 8 GWRD: **a** - **h**) is the P2P network of the GWRD operating at the country level, serving as the country gateways – that is, the GWRDs in countries around the world. The GWRD, which is further discussed in section 6.3, represents the access point for each country and it also holds the directory of city CRDs within the country. The GWRD simultaneously serve as a city's CRD and as the gateway for the entire CRDs in the country.

6.2.1. Global System Overview

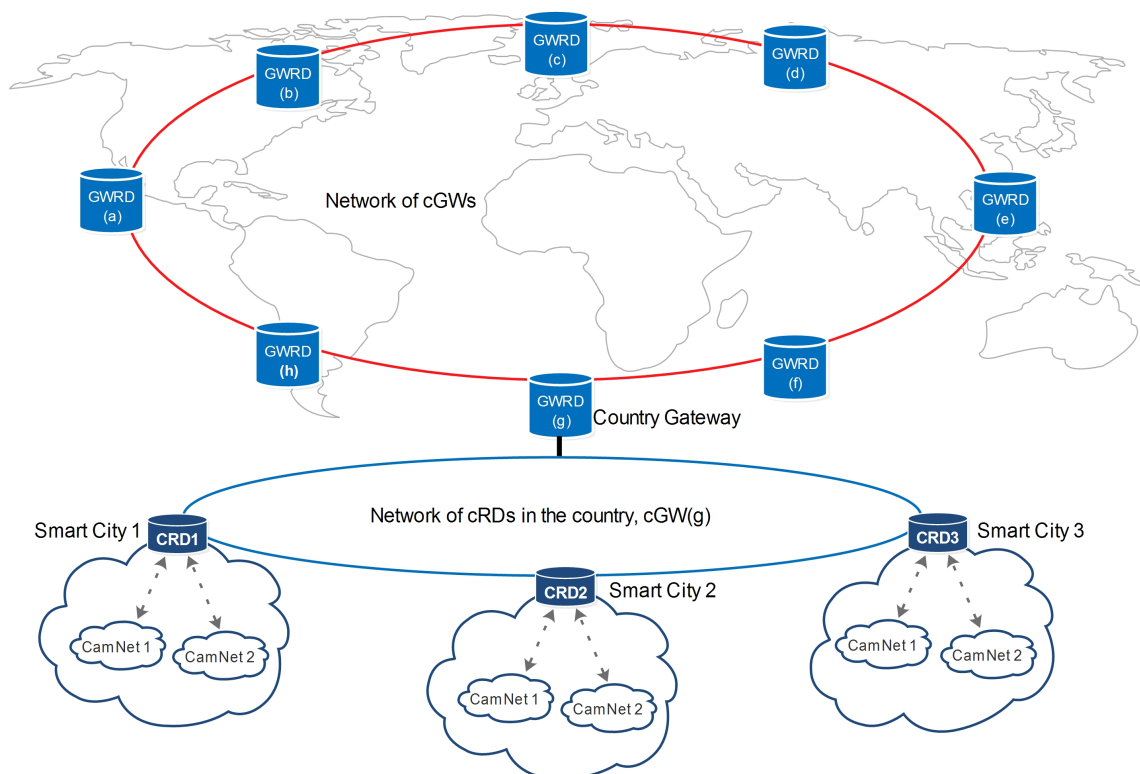


Figure 36: The high-level view of the globally connected surveillance system.

The blue ring depicts the network of the CRDs (that is, the country surveillance system – CoSS, which is further discussed in section 6.2.5) - it is the P2P network that connects the CRDs in the same country, resulting in a network that is only accessible to requests that originate within the country. The cloud on each CRD represents the network of CamNets in which it is the server. The CRD is the directory for the entire CamNets in a city – that is, the city surveillance system (CiSS), which is further discussed in section 6.2.2.

In theory, any CRD can connect to any other CRD anywhere, irrespective of their geographical location. However, this will result in a security vulnerability and inability to establish the locality of a CamNet (or a camera). To add a layer of security to the conceptually flat (P2P) network, we introduce a hierarchical layering where some servers are nominated as the country Gateways. So that requests to a CRD or CamNet remains within the country's GWRD. The GWRD makes the hierarchical structure possible since it serves as the access point to the entire CRDs in the country. With the hierarchy in place, the suggested global architecture is achievable.

6.2.2. *The City Resource Directory Server (CRD)*

The city resource directory server is a self-managing discoverable service, which is depicted in Figure 37. It shows 2 subsystems – the first is the subsystem that connects with the CamNets while the second is used to connect to other CRDs. The operation of each subsystem is powered by a database (directory). First is the directory of CamNets, which is used to administer the entire CamNets in the city – that is, the CiSS. It uses the database to save the configuration, contact, and ownership information about each

CamNet in the city. And depending on the level of access granted by the metadata, the database holds detailed information about the cameras in the city, once the camera's metadata is supplied.

The second database is the directory of the CRDs - it is used to administer the CRD's membership of the country's surveillance system – that is, the CoSS. As further discussed in section 6.2.6, it uses the database to obtain detailed location, and identification properties of the other CRDs on the same CoSS. Also above each database depicted in Figure 37 are some of the published web methods – that is, 'Query Metadata', 'update CamNet' etc. These web methods are those used to communicate with the CamNets or other CRD, as required.

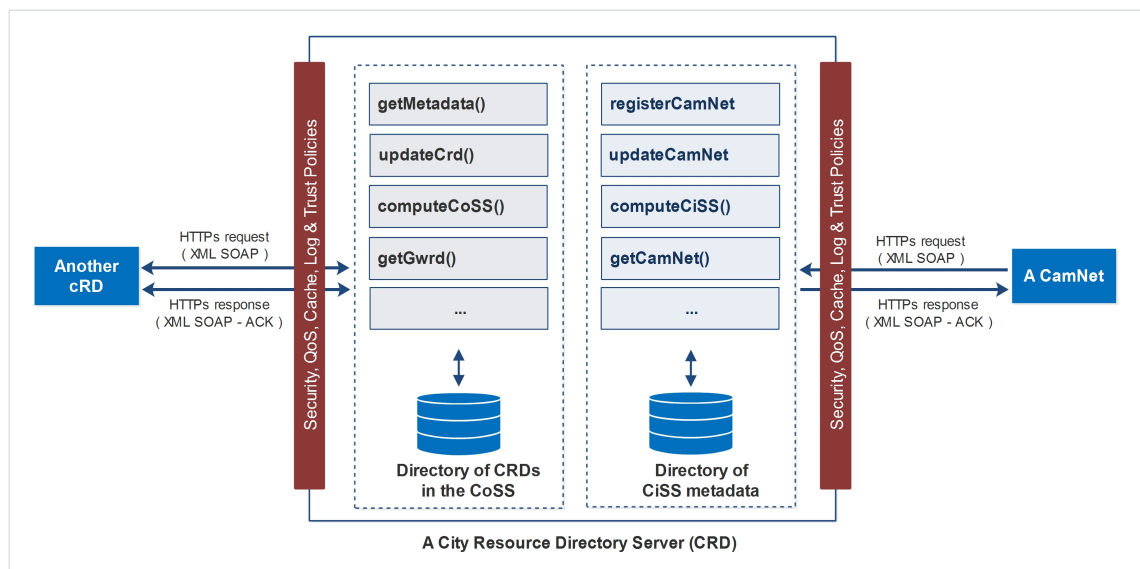


Figure 37: Architecture of the CRD

6.2.3. Implementation of the DHT

The P2P-like distributed dynamic hash table (DHT) keeps information for looking up and discovering other CRDs within a country – such that all the CRDs in a country form

a network of the country's surveillance system. As mentioned in section 6.2.2, to achieve the routing for both CamNets and the CRDs, each CRD is equipped with 2 routing tables. Table 19 is a cut-down version of the simple DHT-like implementation in this research. The Globally Unique Identity Number (GUIDN) field is unique for each entry – it is used to identify the CRD.

The GUIDN field represents the DHT's hash for each node on a P2P network. The field is used to uniquely identify each known CRD on the CoSS – it is also used to index the database table. The naming of the GUIDN allows for human readability. For example, the GUID for the Brighton city in the UK could be 'gb.east-sussex.brighton' – as seen in Table 13. The first part, 'gb' represents the name of the country's locale, the second part, and 'east-sussex' represents the county, the third part, and 'brighton1' represents the first CRD configured for the city of Brighton.

Table 19: An example DHT table on a CRD

| DHT (Resource Directory) | | |
|--------------------------|-------------------------|---|
| ID | GUIDN | URL |
| 1 | gb-east-sussex-london | https://rd.brighton.co.uk/london |
| 2 | gb-east-sussex-brighton | https://rd.london.co.uk/brighton |
| 3 | gb-east-sussex-worthing | https://rd.crawley.co.uk/worthing |
| 4 | gb-east-sussex-crawley | https://rd.washington.com/crawley |
| 5 | gb-east-sussex-hasting | https://rd.boston.com/hasting |
| 6 | gb-east-sussex-bristol | https://rd.brisbane.com.au/bristol |
| 7 | gb-east-sussex-leeds | https://rd.melbourne.com.au/leeds |
| 8 | gb-east-sussex-reading | https://rd.beijing.cn/reading |

6.2.4. The Gateway Resource Directory Server (GWRD)

The country Gateway (GWRD) is an enhanced CRD with capabilities to perform security functions, access control and lookup function for a defined geographical location such as the country. In essence, the main differentiating functionality between a GWRD and a CRD is in the scope of operation. One of the CRDs in a country is nominated and configured to serve as the GWRD. The collection of the GWRD and the entire CRDs it manages is the country's surveillance system (CoSS).

For a request sent to a device (CamNet or an CRD) outside the country of origin, the GWRD in the originating country is responsible for verifying, validating and authorizing the request and the sender. If the request is valid, the destination GWRD adds a 'token' to the message before sending it to the GWRD in the destination country. Upon receipt of a request, the destination GWRD verifies whether the message originated from an authorised sender by checking the 'token' added by the source GWRD. If valid, the destination GWRD accepts the request. Its functions include the following:

1. The GWRD represents a country in a network hierarchy that emulates the geopolitical landscape of the world.
2. It serves as an access control server for the country.
3. All requests and responses that go through the GWRD can be logged for the purposes of system audit.

6.2.5. Overview of the Country Surveillance System

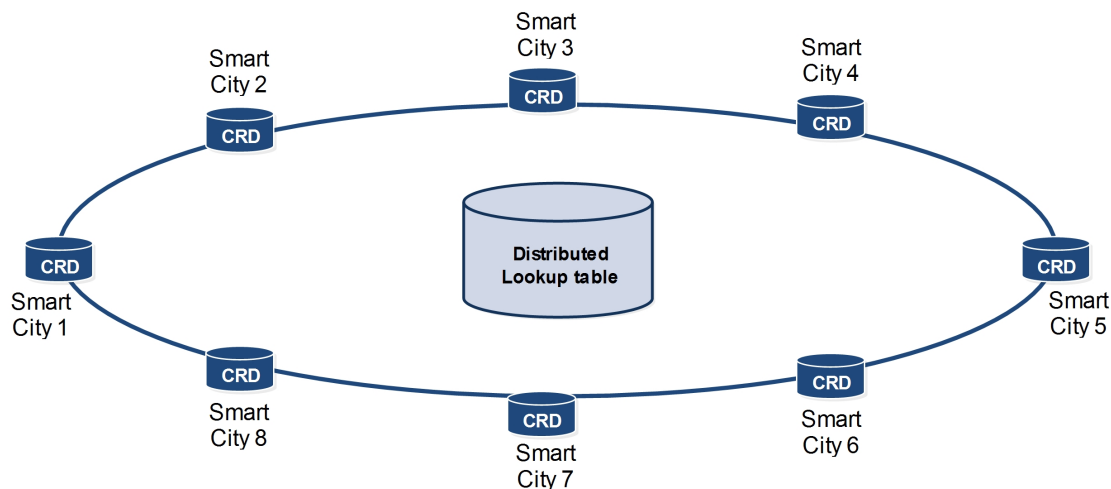


Figure 38: A system overview of the CoSS.

Figure 38 depicts the CoSS architecture, showing it comprise of the entire CRDs in a country – that is, all the CRDs that belong to the same network, which together form the country’s surveillance system, the CoSS. The resulting network is a P2P-like network in which the most up to date lookup table is maintained by the GWRD, which also acts in the capacity of the network server.

Each CRD acquires the knowledge of the GWRD during the installation or update when a human administrator adds the URI of the GWRD, which the CRD uses to notify the GWRD about itself. Upon receiving new CDR details, the GWRD saves this detail in its own lookup table. For this reason, a CRD forwards any request that was not meant for itself to the GWRD for final decision. As in Figure 39, the two CRDs operate at the same hierarchy level in which communication is established from the knowledge of each other in the lookup table – the lookup table is distributed across the CRD and the GWRD. In essence any CRD can find any other CRD on the network through its connection to the GWRD.

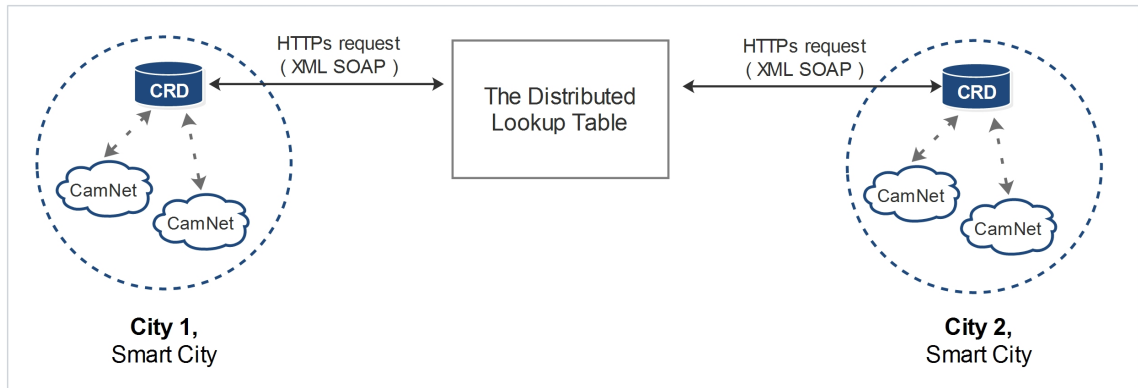


Figure 39: City connectivity through DHTs.

6.3. The System Overview of a City

The CiSS system architecture, which is depicted in Figure 40, is the client-server network of the CamNets in a city. Each CamNet joins the city's CiSS upon successful registration and sends metadata to the CRD, based on its configuration. A CamNet only has knowledge of its own existence and the CRD - and does not have the knowledge of any other CamNet or a camera outside its own network. The only connectivity a CamNet has about the CiSS is the URL for making calls to the CRD.

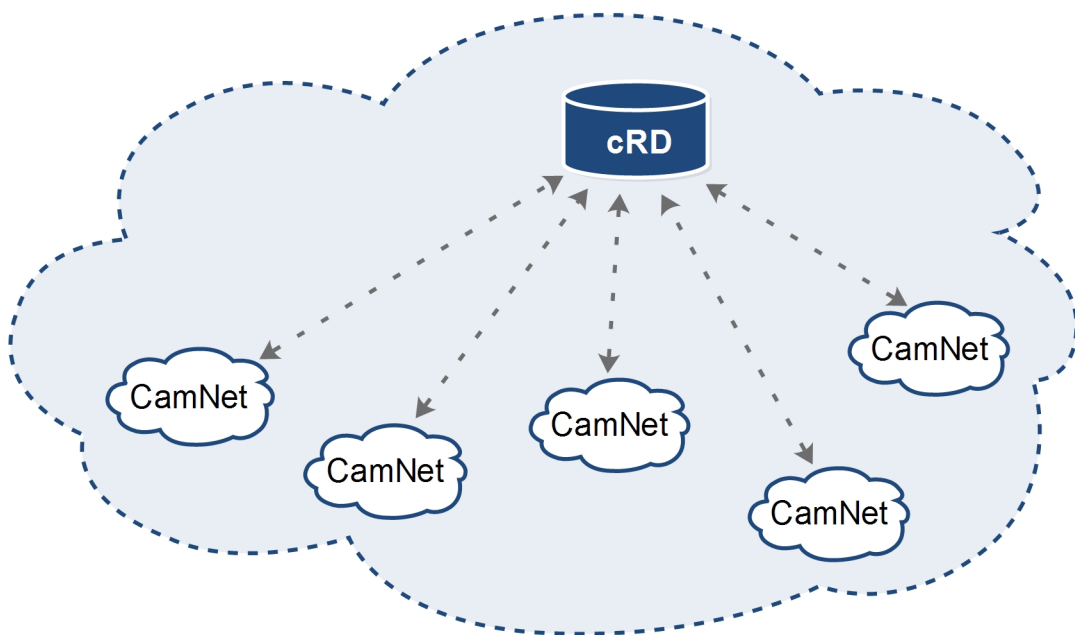


Figure 40: The system overview of the CiSS

6.4. CamNet Registration

With regards to the algorithms that operate these experiments, we identify a use case to demonstrate and validate the approach employed in this research:

1. **New registration process:** The addition of a new CamNet, which aims to join the relevant CRD in the city of its location. The experiment aims to ascertain how availability of the CRDs affects the registration algorithm, which is presented in section 6.4.2.

6.4.1. *Registration Process*

The submission of the registration form (as discussed in section 4.3.9) activates the registration algorithm presented in **section 6.4.2** below. The MDS only needs to have the registration URL for any CRD - the registration process will ensure that the appropriate CRD is located and accepts the CamNet, if it meets the registration requirement. The success of the algorithm to locate the appropriate CRD is aided by the GUIDN naming convention discussed in section 6.2. Based on the operation of the algorithm, if a CRD receives a request destined for a different CRD, it simply forwards the request to the appropriate CRD, using the CRD tables mentioned in section 6.2.5.

6.4.2. *Algorithms 6A: CamNet to CRD Registration*

Input: A request sent by CamNet *d*, CamNet's identity *e*.

Process: Lookup suitable CRD for request *f*, add CamNet to own directory *g*, send response to CamNet *h*,

Condition: CamNet is located in own city *i*, CamNet in the same country as self *j*, CRD is found for address *k*.

Response: SUCCESS, FAIL, NO_RESPONSE.

```

1. Begin
2. init count_HOP = 0;
3. if get d == TRUE AND get e == TRUE then
4.   do f
5.   if i == TRUE AND j == TRUE
6.     do g
7.     do h = SUCCESS
8.   else
9.     do count_HOP++
10.  if j == FALSE then
11.    do tag d = INTERNATIONAL
12.    forward d to GWRD
13.  else if i == FALSE AND j == TRUE then
14.    do forward d to k
15.  else if i == TRUE AND k == FALSE then
16.    do h == FAIL
17.  endif
18. endif
19. endif
20. END

```

6.4.3. Algorithms 6B: CamNet to GWRD Registration

Input: A request sent by a CRD *l*, a request tagged INTERNATIONAL *m*, GWRD is allowed to communicate with destination country *n*.

Entities: Sending CRD *o*, recipient country *p*.

```

21. Begin
22. if get l == TRUE AND m == TRUE AND n == TRUE then
23.   forward l to p
24. else do
25.   if get l == TRUE then
26.     if ping k == TRUE
27.       forward response to k; endif
28.   else
29.     broadcast message = 'k unavailable'
30.   endif
31. END

```

6.5. Experimentation and Evaluation

6.5.1. *Experiment Setup*

To demonstrate the registration process, we develop an experimental project to evaluate and assess the efficiency of the approach in relation to the success rate of CamNets registering with a city CRD. We further investigate how the end of life of a CRD (within a CoSS) could affect the reliability of the system with a view to providing answers to the following research question: ***“To what degree of accuracy is the registration process for a new CamNet in respect to locating the politically correct city CRD for the CamNet achieved?”***. In particular, we investigate these parameters under 2 conditions:

2. When all the known CRDs are operational and available.
3. When some of the known CRDs are not operational – in real life, this could be due to a fault or other network issues.

Note that our evaluation did not account for the underlying performance issues based on the end-to-end attributes of our setup. This is because our interest in the research is in the outcome at the application level and our setup does not include the network load variables in typical network-oriented experiments.

Using the Rapid Application Software development methodology, we completed a prototype, based on the CoSS architecture presented in section 6.2.5. The programming aspects of the experiment was achieved using the *‘Hypertext Pre-*

processor, PHP'⁵ (version 5.6.10), supported by the 'MySQL database community server'⁶ (version 5.5.47) to set up the cameras, MDS, CRDs and CamNets as described in chapter 4. The project was built using the community version of the *Zend framework version 2*⁷. Although the project could be implemented with other programming environments such as Java/Oracle, or .NET/MSSQL, the PHP/MySQL combination was chosen for the experiment for the following reasons:

- Ability to build a quick web-based user interface for setting up and configuring the components in this research including the cameras, MDS, CRDS, and the accompanying data, based on the design approach in chapter 4.
- They are open-source and community driven software, ease of modification and adaptability to the research implementation.
- Ability to develop, implement and deploy the SOAP-based web services with limited system infrastructure and network resources.
- Ease of set up since we needed to create multiple VMs and configure each VM as desired.

The architecture of the experimental set up is presented in Figure 41 – note the servers are not physically connected neither did we run networking software to form a network because our current experiment did not account for the underlying networking. The links between the CamNet and the CRDs are an indication that the CamNet has equal access to any of the CRDs. Each CRD hosts a database with 2

⁵ <http://www.php.net/>

⁶ <http://www.mysql.com/>

⁷ <http://framework.zend.com/manual/current/en/user-guide/overview.html>

databases as in the CRD architecture. While the CamNet setup is based on the configuration of the MDS.

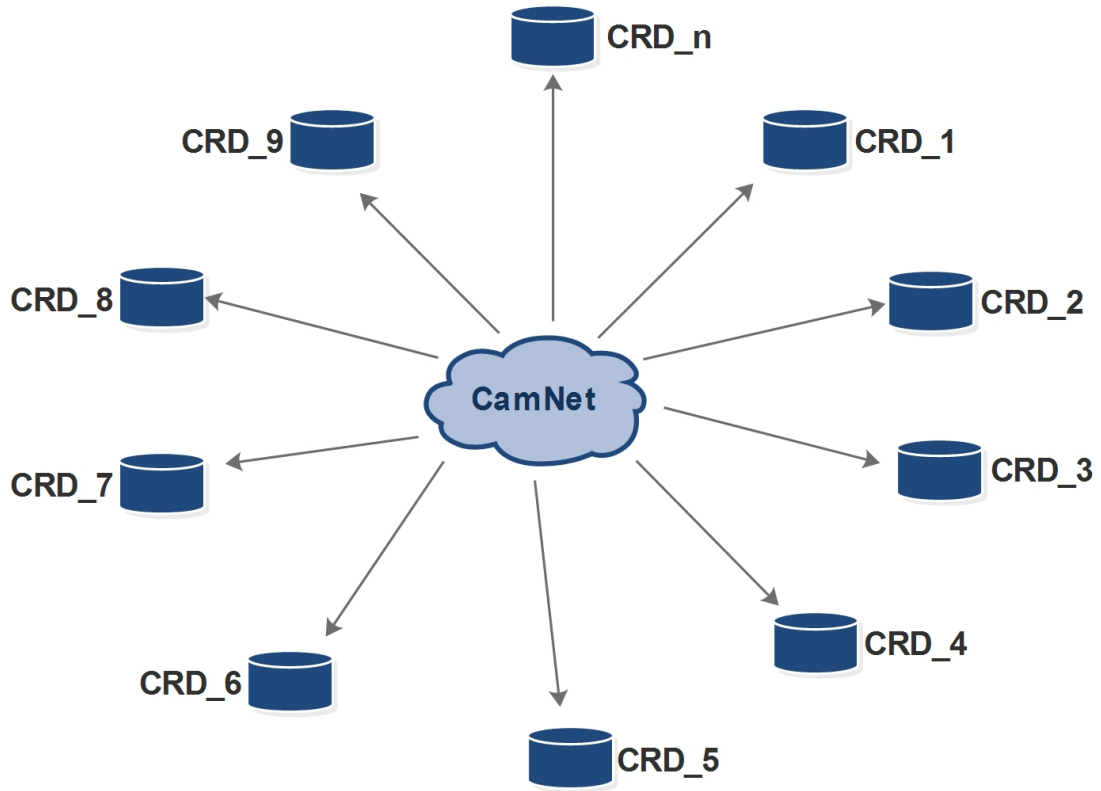


Figure 41: The Setup showing the research's arrangement of the CRDs in a country.

6.5.2. *Simulation*

The CamNet choses a CRD at random from the total n CRDs (where n can be any integer – we experiment with $n = 1000, 5000$, and 10000), and submits each registration request to the randomly chosen CRD. The project implemented the 'mt_rand()' random number generator, which is shipped with PHP library. The generator is an adaptation of the 'Mersenne Twister' random number generator [141]. To ensure that the CamNet registration is valid for one of the n CRDs, we provide a list of n IP addresses where each IP address is valid for 1 CRD. When submitting the

registration, the CamNet assigns itself one of the n IP addresses, which the receiving CRD uses to check whether the request is valid for itself.

The simulation experiments are employed to investigate the achievable reliability (by the CoSS) in detecting and registering a CamNet in the appropriate city's CRD, when the url of the city CRD is not known beforehand.

6.6. Results

The experiment was simulated first with all the deployed CRDs running as a flat network – that is, any special server does not control the registration process. Once this is achieved, we carry out a simulation in which we introduce 2% failed CRDs, then we increased the size of failure by 2% until 20%. It is observed from the result that high number of failed nodes could render the network inactive. Based on this observation, we introduce the GWRD such that the network evolved into a CoSS (see section 6.2.5), where the GWRD acts as the gateway for the client/server network. The GWRD manages communication among the CRDs to ensure that a CRD that is known as failed will be removed from the list of available CRDs.

6.6.1. *Flat and Error-Free Surveillance Directory*

Figure 42 depicts the result of experimenting all CRDs as a flat architecture in which any CRD is capable of directly communicating with any other CRD through the DHT (see section 6.2.3). In this set of experiments, it is noted that the success rate of registration is 100% for any number of requests, and for any number of CRDs. This result satisfies our suspicion since 100% success rate is expected when all the CRDs are

available and active - however, the result does not reveal anything interesting worth further investigation.

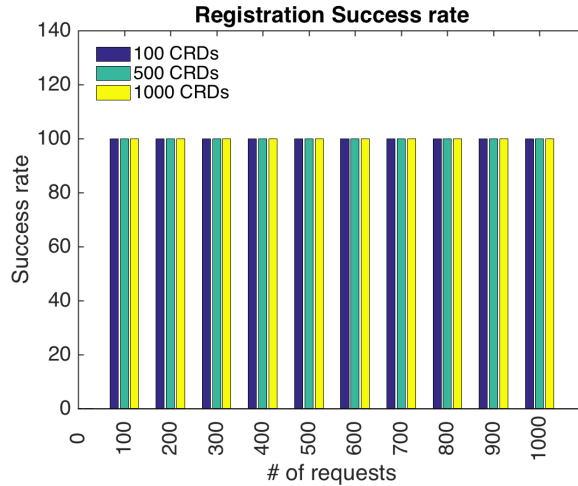


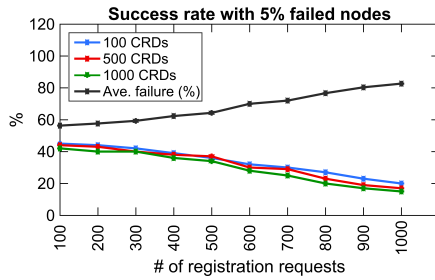
Figure 42: CamNet registration success rate remains at 100% despite variation in the number of CRDs and the number of registrations.

6.6.2. *The Effects of Failed Nodes on the Flat Architecture*

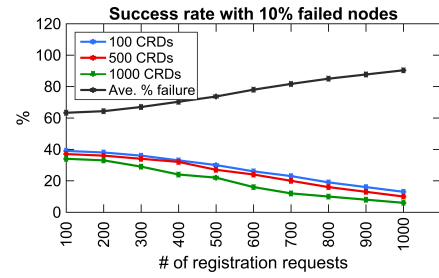
With a view to investigating the ability of our solution to function when nodes fail (that is, if a known CRD fails), we introduce a level of uncertainty to the network by making some nodes unavailable. To achieve this we disable some nodes while the details of the disabled nodes still remain in the DHT. Then we varied the number of failed node to assess the impact of network disruptions of faulty CRDs and how these would impact the availability of self-registration of the CamNets. In the first set of experiments, we introduce 2% failed nodes and then we increase this by 2% until 20 - the results are presented in Figures 43 (a) to (d).

Figures 43 (a) shows the result of 5% failed nodes – it can be observed that the failure rate rises with increase in the number of requests, for any number of CRDs on the network. It is also observed that the success rate decreases as the request grows

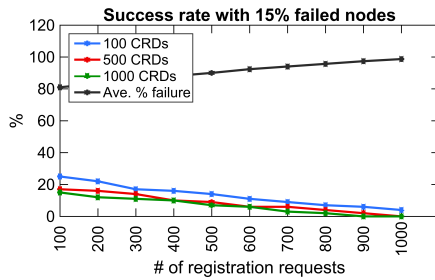
tending towards 0 reliability but it did not reach 0 with 1000 requests. The results for 10%, 15%, and 20% failed nodes are similar to those observed with the 5% except that the success rate actually was at zero with both 15% and 20% failed nodes.



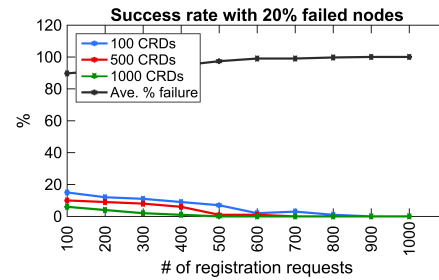
(a)



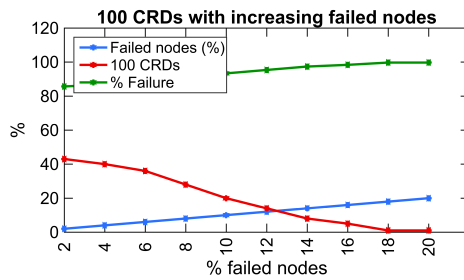
(b)



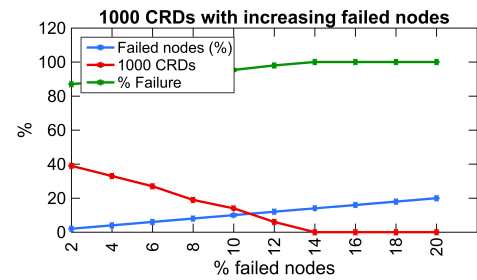
(c)



(d)



(e)



(f)

Figure 43: System failures due to failed nodes

With a view to understanding the reason for zero success rate and the exact point at which the 100% failure occurred, we repeated the simulations while we change the failed nodes by an increment of 2% and the number of requests was kept constant at 500. Figure 43 (e) depicts the effect of the simulations on a 100 CRD network, while

Figure 43 (f) shows the same experiment for 1000 CRDs. It can be observed that the failure rate reached 100% between 17% and 19% with the 100 CRDs. While it reached 100% between 13% failed nodes and 15% failed nodes for the 1000 CRDs network. This confirms our suspicion that the network can be grounded to a total crash at some point and that the crash will happen earlier with higher number of failed nodes in the network.

6.6.3. *The Effect of GWRD on System Failure*

To mitigate the negative effect of the failed nodes on the registration process that was identified in section 6.6.2 above, we introduced the GWRD to achieve client-server architecture as discussed in section 6.2.5. The GWRD serves as the server for the other CRDs by acting as the gateway for all the CRDs and by acting as the server, which manages the distributed DHT. However, it is observed that the improvement achieved by the introduction of the GWRD is negligible since the highest success rate was just above the 60% mark – this is observed in Figures 44 (a) and (b) with 100 CRDs and 1000 CRDs respectively.

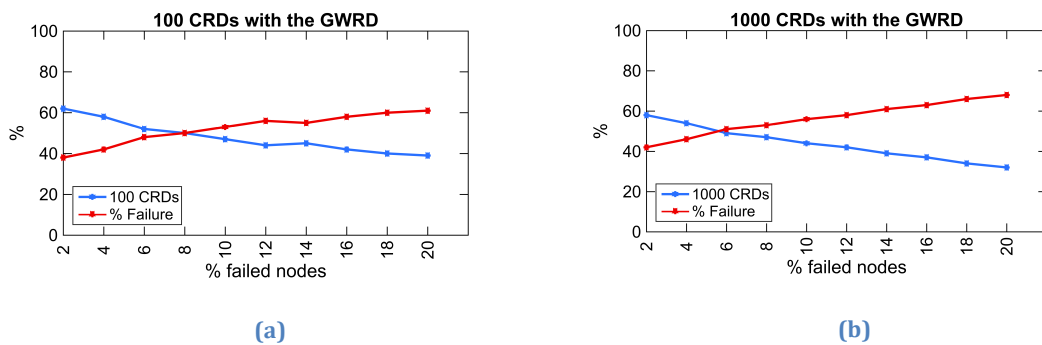


Figure 44: Success and failure with the client-server CoSS

6.6.4. The CoSS Topology with Self Management

To resolve the problem of network crash due to the failed nodes, as discussed above, we apply a level of self-management to the CoSS registration algorithm, such that the requesting CRD first checks the reachability of the destination CRD before sending a request. If a CRD is not reachable, then CRD sends the message to the GWRD instead. The GWRD then computes the best CRD for the CamNet (based on the CamNet's current IP address).

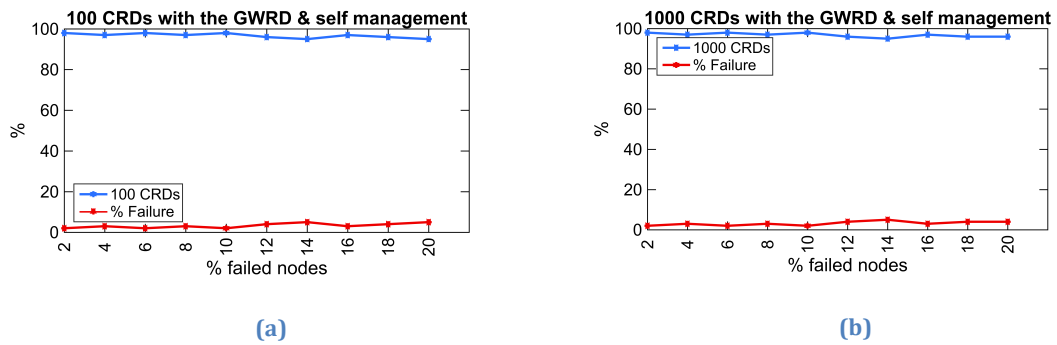


Figure 45: Post optimisation result showing the percentage success rate of the CamNet registration attempts with a 10% unreachable CRDs

If a CRD is identified for the request, the GWRD forwards the registration request accordingly. Otherwise, the GWRD sends a 'failed' response to the CamNet and the GWRD sends a message to all CRDs on the network to disable the failed CRD in the CoSS DHT. With this solution in place, we are able to prevent a network crash where the worst-case reliability achieved was at 96% with 20% failed nodes. Figures 45 (a) and (b) both showed similar level of success varying 96% to 99% for any number of failed nodes.

It is observed that achieved reliability is higher than the probability score of the system. For example, in the 20% failed node, the maximum success rate will be 80%,

based on probability score. However, the achieved success rate is an average of 97%. This is because a failed node is marked disabled and the CamNet does not further attempt registration with the disabled CRD. In a real life situation, once a failed node comes back online, it registers itself again in the DHT and becomes available to receive registration requests.

A real life scenario of the registration process can result when a CamNet is relocated to a different city. The CamNet will need to register with the CRD in the new city but the CamNet only have the detail of the old CRD. Once the CRD in the old city is contacted from the new location, the CRD assesses the CamNet's new location information and forwards the request to the appropriate CRD for the CamNet's new city.

6.7. Chapter Conclusion

A global network of CRDs (as depicted in Figure 36) is capable of providing information about any camera in the world, through the local membership of a CamNet. In practice this solution could support the public safety departments in identifying and locating surveillance networks across the city without the need for physical street inspection – this could potentially improve the efficiency and success of police investigations. In practice, we envisage that the size and number of cameras in the city will change erratically over time as cameras/CamNets are added, transferred and migrated from place to place.

In section 6.6.2, where we varied the availability of the nodes to achieve between 5% and 20% failed nodes, the accuracy of process was significantly low (46% in the best case scenario). This demonstrated that an unreachable CRD could disrupt the

functioning of the system. However, based on the introduction of the self-managing procedure where the GWRD first checks the availability of a node before communicating, the reliability of the registration process achieved at least 34% improvement to the reliability of the registration process (that is, 96% in Figure 45(b), and 62% in Figure 44 (a)). Therefore this research recommends self-update and self-management as a requirement for the CoSS.

The experiments in this chapter ultimately answered our research questions, **RQ2 – Can we query surveillance networks that belong to multiple independent administrative owners?**. The results from this research indicate that this proposal has great potential in solving security and surveillance problems. However, there is still a great deal of work needed to implement and test the different components.

7. Conclusion and Future Directions

A story has no beginning or end: arbitrarily one chooses that moment of experience from which to look back or from which to look ahead.

- Graham Greene

7.1. Overview

This thesis has researched the unification of independent video metadata grounded on the concepts of graph theory, person re-identification, and self-aware algorithms to propose a system architecture for a globally unified video surveillance system, which is capable of representing the world's geopolitical structures such as a city or country. The thesis demonstrated an approach to unify the video metadata that belong to independent owners, which makes it possible for a system user to query the unified metadata, with a view to analysing the activities of an identifiable object, such as a person across the unified surveillance system. Our findings follow a literature review of the current approaches and the future direction in technologies - this identified several issues requiring solutions for future sustainable video surveillance systems.

7.2. Contributions of the Thesis

In Chapter 3, this thesis presented the FVSA, a system architecture, which aims to provide support for unifying independent surveillance systems. We showed that each implementation of the FVSA is an independent surveillance network. However, a collection of individual FVSA could hierarchically integrate as a unified system - such as the city surveillance system, the **CiSS**. And a collection of CiSS across a country can integrate to form the country's surveillance system, the **CoSS**.

Chapter 4 was a detailed account of the technical considerations of the design of the CamNet, where this thesis demonstrates the detailed design and implementation of a simulation experiments in the research. The design is the basis for providing answers to 2 research questions that were raised in section 1.5. First the developmental

question in **RQ1**: *“Can we design surveillance systems with a view to exchanging information across independent networks?”*. The second question answered is part b of the developmental question in **RQ2**: *“Can the architecture represent the geo/political structure of the world?”*. This chapter provides an approach to answering these questions, showing the design strategies, configuration of the components, and representation of the data.

Chapter 5 involved the development and implementation of an experiment, which simulates the CamNets. The simulated CamNet was based on the design in chapter 4, where the CamNet is presented as a component within a CiSS and we showed how their output (metadata) contributes to the public level investigation and decision, using an experimental project that was developed as a citywide resource directory. The results revealed that the number of cameras monitoring an area reaches saturation point between 80 and 100 cameras per 1Km².

In chapter 6, we present the conceptual view of the global video surveillance devices and services, making it possible to approach video networks in layers such as internal system (local) or external system (global). The framework of our solution is compatible with the hierarchical structure of computer networks and emerging technological platforms such as the IoT. We discussed and implemented experiments based on the CoSS architecture. The experiment involved the automated registration of a CamNet where the CamNet did not have fore knowledge of the CRD that is appropriate for it. The results achieved a success rate of an average 97%, which demonstrated that the proposed solution is feasible.

7.3. Limitations and Challenges

This research suffered in the ability to accurately emulate existing solutions in video surveillance since the research's theme is futuristic and predictive. The size of data required to test our hypothesis and theories in the research could not be achieved in real life video surveillance systems. In addition the theme is reasonable since it is a common phenomenon in this area of research and it opens opportunities for improving the accuracy and applicability of video solutions. For example the concept re-id is still seriously evolving with several research efforts contributing to the improvement of re-identifying of people across cameras (see section 2.8). However, some aspects and components are already implemented and deployed in real life situations. For example the SSA architecture discussed in section 2.3 is being implemented.

This simulation experiment employed the open set re-id approach, where the size of gallery and probe increases as new metadata is introduced. The continuous growth of the metadata resulted in the exhaustion of computing resources (memory and CPU capacity) during the generation and processing of metadata. In other words, the use of an elastic infrastructure such as a cloud based computing facility will allow for the generation of more data, which is likely to improve the performance of the system and potentially influence the research results, as noted in section 5.7.4.

In a real life scenario, the demonstration will involve the location of physical servers in multiple cities. However, these facilities were not obtainable during the demonstration of the research, instead encapsulated database objects and virtual machines were

used to simulate components in which the virtual machines were deployed on the same network.

7.4. Ethics and Privacy

The perception of the public to the advanced exploration of video surveillance data includes privacy concerns. This research considers the wider benefits of surveillance data including business usage, service availability and improvements such as the use case for transportation planning that was discussed in chapter 3. The scope of this research is within the use of technology in improving the benefits of surveillance system, which are already noticeable everywhere, which cameras are used to achieve the notion of safety.

7.5. Future Work

This thesis has proposed solutions to some problems and challenges identified in the research gap section - it provided answers to all the research questions raised in section 1.5. However, there are still open questions and challenges that can be answered by building on the achievements of this research in the future.

7.5.1. *Standardisation and Terminologies*

The experiments presented in this thesis involved the development of web services and other interfaces, which are in need of review to assign standard terminologies and naming conventions that enhances universal integration between various vendors, manufacturers and implementers of the system in real life. The components introduced in this thesis such as the CamNet, and the CRDs presents a set of

unexplored interfaces that needs standardising and a conformance approach to implement and integrate them with existing systems.

7.5.2. The Scope of the Solution

The unification strategy for the city metadata assumes unique camera FOVs. However, in real life situations, the FOVs of multiple cameras are likely to overlap. Future work that caters for this would improve the accuracy of the querying the FVSN topology and could also improve the re-id results. The globally scoped distributed surveillance directory, which was demonstrated in chapter 6, was demonstrated within a country. It would be interesting to develop and deploy this across various countries to investigate and explore the applicability and challenges of internationally connected surveillance network.

7.5.3. Testing

Similar to other simulation-based research, we want to apply the hypothesis and theories of this research to real life environments where this could help discover the real behaviour of people and on surveillance network and potentially a review to the proposed solution. The deployment of servers in various geographic locations may introduce new ideas and underlying network parameters that could reveal new directions, which have not been addressed in this thesis.

Future work will encompass testing and validation of the end-to-end parameters of the global surveillance system. These were set up, tested, and validated as independent components in the current research. However, future work will set up and configure

CamNets, CiSS, and the CoSS in multiple countries to assess the real life implications of the integration approach and the achievable result.

7.5.4. Security and Accessibility

An important consideration is the management and enforcement of the security boundaries that is suggested with implementation of the GWRD (as demonstrated in chapter 6). In practice, each country has independent internal security requirements, which may require a great deal of adaptation and customisation. Since each country is usually responsible for managing own internal security, a global integration strategy, which caters for each country's requirements, needs to be investigated and applied at the local level. The process of collating and setting up the requirements will open up a set of issues, challenges, and opportunities that may lead to a completely different directions and results that were already obtained in this thesis.

The concerns and legality around peoples' privacy is suspected to be different around the world. Although the requirements of the UK data protection act is achieved since the identity of people are preserved, this may not meet the requirement in some countries. A review of the data protection requirements in various countries will help identify the potential issues around this topic and assess how it affects an implementation of this solution globally. It may be the solution is not fit for those countries or maybe only aspects of it will be implemented.

7.6. Final Notes

This thesis presents a novel approach to designing and utilising video surveillance systems by systematic exploration of relevant technologies to achieve self-awareness and integration-awareness among systems that are owned and managed by independent owners. The presented approach is achieved at the software layer of the surveillance devices to unify video metadata across a city and the ability to query the data at the public level. Thereby promoting public level analytics and exploration of video surveillance systems across a city or a country.

The solution involves configuration options for achieving data protection requirements, which protects the privacy of the people captured in the surveillance data. This is aided by segmenting the resulting globally unified system into hierarchies that emulate the geo-political structure of the world while it also complies with the trend in technology - as a component in the IoT compliant smart city. The author hopes this thesis will provide a pathway for critically evaluating the results derived from surveillance systems future research that improves the relevant technologies in the field.

8. References

- [1] T. Savolainen, J. Soininen, and B. Silverajan, "IPv6 Addressing Strategies for IoT," *IEEE Sens. J.*, vol. 13, no. 10, pp. 3511–3519, Oct. 2013.
- [2] S. Cirani, L. Davoli, G. Ferrari, R. Leone, P. Medagliani, M. Picone, and L. Veltri, "A Scalable and Self-Configuring Architecture for Service Discovery in the Internet of Things," *IEEE Internet Things J.*, vol. 1, no. 5, pp. 508–521, 2014.
- [3] S. S. Mathew, Y. Atif, Q. Z. Sheng, and Z. Maamar, "Web of Things: Description, Discovery and Integration," in *2011 International Conference on Internet of Things and 4th International Conference on Cyber, Physical and Social Computing*, 2011, no. Contribution 3, pp. 9–15.
- [4] Y. Li, Z. Zhou, and W. Wu, "Iterative pedestrian segmentation and pose tracking under a probabilistic framework," in *2012 IEEE International Conference on Robotics and Automation*, 2012, no. 61170188, pp. 1206–1211.
- [5] Y. Zhang, L. Dong, J. Li, S. Li, and Z. J. Gao, "A complex network-based approach to estimating the number of people in video surveillance," in *2013 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, 2013, pp. 1–4.
- [6] A. N. Staranowicz, C. Ray, and G. Mariottini, "Easy-to-use, general, and accurate multi-Kinect calibration and its application to gait monitoring for fall prediction," in *2015 37th Annual International Conference of the IEEE*

Engineering in Medicine and Biology Society (EMBC), 2015, pp. 4994–4998.

- [7] E. E. Stone and M. Skubic, "Unobtrusive, Continuous, In-Home Gait Measurement Using the Microsoft Kinect," *IEEE Trans. Biomed. Eng.*, vol. 60, no. 10, pp. 2925–2932, Oct. 2013.
- [8] R. L. Villars, C. W. Olofson, and M. Eastwood, "Big Data: What It Is and Why You Should Care," Farmingham, USA, 2011.
- [9] J. Burn-Murdoch, "Study: less than 1% of the world's data is analysed, over 80% is unprotected," *The Guardian*, 2014. [Online]. Available: <http://www.theguardian.com/news/datablog/2012/dec/19/big-data-study-digital-universe-global-volume>.
- [10] IBM, "Bringing big data to the enterprise," 2016. [Online]. Available: <https://www-01.ibm.com/software/data/bigdata/what-is-big-data.html>. [Accessed: 01-Sep-2016].
- [11] S. Paschalakis, K. Iwamoto, P. Brasnett, N. Sprljan, R. Oami, T. Nomura, A. Yamada, and M. Bober, "The MPEG-7 Video Signature Tools for Content Identification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 7, pp. 1050–1063, Jul. 2012.
- [12] Youtube, "Youtube Statistics and Viewership," 2013. [Online]. Available: <http://www.youtube.com/yt/press/en-GB/statistics.html>.

- [13] Cisco, "Cisco Visual Networking Index: Forecast and Methodology, 2012–2017," 2013. [Online]. Available:
https://www.ciscoknowledgenetwork.com/files/385_06-25-13-Americas_VNI_June_2013_Forecast_Update.pdf.
- [14] T. J. Barnett, A. Sumits, S. Jain, and U. Andra, "Cisco Visual Networking Index (VNI) Update Global Mobile Data Traffic Forecast," *Vni*, pp. 2015–2020, 2015.
- [15] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. Hung-Byers, "Big data: The next frontier for innovation, competition, and productivity," 2011. [Online]. Available:
http://www.mckinsey.com/features/big_data.
- [16] S. Adcock and P. Norstrom, "Just 1 in 70 CCTV Cameras are State-Owned: Survey Revelation by the British Security Industry Association (BSIA)," *Press Conference, London*, 2013. [Online]. Available:
<http://www.bsia.co.uk/home/bsia-cctv-number-of-cameras-in-uk>.
- [17] H. E. Neely, R. S. Belvin, and M. J. Daily, "Modeling threat behaviors in surveillance video metadata for detection using an Analogical Reasoner," in *2010 IEEE Aerospace Conference*, 2010, pp. 1–9.
- [18] ICO, "The Guide to Data Protection," *Inf. Comm. Off.*, pp. 1–130, 2015.
- [19] A. Lytkin, *IP Video Surveillance. An Essential Guide*. Amazon Kindle Edition,

2012.

- [20] G. Borga, R. Camporese, L. Di Prinzio, N. Iandelli, S. Picchio, and A. Ragnoli, "New technologies and EO sensor data build up knowledge for a Smart City," in *Data Flow from Space to Earth - Applications and Interoperability*, 2011.
- [21] S. Russo, "Digital Video Surveillance - with analytic capabilities," 2008.
- [22] IBM, "IBM Smart Surveillance Research - Publications," 2013. [Online]. Available: http://researcher.watson.ibm.com/researcher/view_grouppubs.php?grp=1394.
- [23] S. I. Ali, "Adoption of cloud computing in manufacturing industry supply chains, a hype or a myth?," in *Second International Conference on Future Generation Communication Technologies (FGCT 2013)*, 2013, pp. 69–72.
- [24] C.-F. Lin, S.-M. Yuan, M.-C. Leu, and C.-T. Tsai, "A Framework for Scalable Cloud Video Recorder System in Surveillance Environment," in *2012 9th International Conference on Ubiquitous Intelligence and Computing and 9th International Conference on Autonomic and Trusted Computing*, 2012, pp. 655–660.
- [25] R. S. Feris, B. Siddiquie, J. Petterson, Y. Zhai, A. Datta, L. M. Brown, and S. Pankanti, "Large-Scale Vehicle Detection, Indexing, and Search in Urban Surveillance Videos," *IEEE Trans. Multimed.*, vol. 14, no. 1, pp. 28–42, Feb.

2012.

- [26] J. Connell, Q. Fan, P. Gabbur, N. Haas, S. Pankanti, and H. Trinh, "Retail video analytics: an overview and survey," *Proc. SPIE - Int. Soc. Opt. Eng.*, vol. 8663, no. March, p. 86630X, 2013.
- [27] H. Zhou, L. Jia, and Y. Qin, "Metadata Specification of Railway Video Information and Its Application in Video Monitoring System for Qinghai-Tibet Railway," in *2009 International Symposium on Computer Network and Multimedia Technology*, 2009, no. 600332020, pp. 1–4.
- [28] H. E. Neely, R. S. Belvin, and M. J. Daily, "Converting video metadata to propositional graphs for use in an analogical reasoning system," 2013.
- [29] Broad Agency Announcement (BAA 06-01-MT), "Proposers Information Pamphlet (PIP) for the Video Analysis and Content Extraction (VACE) Program – Phase III," 2006.
- [30] R. Nevatia, J. Hobbs, B. Bolles, and M. Rey, "An Ontology for Video Event Representation," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW'04)*, 2004, pp. 119–119.
- [31] A. R. J. Francois, R. Nevatia, J. Hobbs, and R. C. Bolles, "VERL: An Ontology Framework for Representing and Annotating Video Events," *IEEE Multimed.*, vol. 12, no. 4, pp. 76–86, Oct. 2005.

- [32] D. Chu, C. Jiang, Z. Hao, and W. Jiang, "The Design and Implementation of Video Surveillance System Based on H.264, SIP, RTP/RTCP and RTSP," in *2013 Sixth International Symposium on Computational Intelligence and Design*, 2013, vol. 2, pp. 39–43.
- [33] C. Verma and S. Dey, "Methods to Obtain Training Videos for Fully Automated Application-Specific Classification," *IEEE Access*, vol. 3, pp. 1188–1205, 2015.
- [34] H. Zhou and G. K. H. Pang, "Metadata extraction and organization for intelligent video surveillance system," in *2010 IEEE International Conference on Mechatronics and Automation*, 2010, pp. 489–494.
- [35] R. Rawassizadeh, J. Heurix, S. Khosravipour, and a. M. Tjoa, "LiDSec- A Lightweight Pseudonymization Approach for Privacy-Preserving Publishing of Textual Personal Information," in *2011 Sixth International Conference on Availability, Reliability and Security*, 2011, pp. 603–608.
- [36] M. Bober and P. Brasnett, "MPEG-7 visual signature tools," in *2009 IEEE International Conference on Multimedia and Expo*, 2009, pp. 1540–1543.
- [37] J. O. Kephart and D. M. Chess, "The vision of autonomic computing," *Computer (Long. Beach. Calif.)*, vol. 36, no. 1, pp. 41–50, Jan. 2003.
- [38] S. Dobson, R. Sterritt, P. Nixon, and M. Hinchey, "Fulfilling the Vision of Autonomic Computing," *Computer (Long. Beach. Calif.)*, vol. 43, no. 1, pp. 35–

41, Jan. 2010.

- [39] P. R. Lewis, A. Chandra, F. Faniyi, K. Glette, T. Chen, R. Bahsoon, J. Torresen, and X. Yao, "Architectural Aspects of Self-Aware and Self-Expressive Computing Systems: From Psychology to Engineering," *Computer (Long. Beach. Calif.)*, vol. 48, no. 8, pp. 62–70, Aug. 2015.
- [40] T. Chen and R. Bahsoon, "Toward a Smarter Cloud: Self-Aware Autoscaling of Cloud Configurations and Resources," *Computer (Long. Beach. Calif.)*, vol. 48, no. 9, pp. 93–96, Sep. 2015.
- [41] S. Kounev, N. Huber, F. Brosig, and X. Zhu, "A Model-Based Approach to Designing Self-Aware IT Systems and Infrastructures," *Computer (Long. Beach. Calif.)*, vol. 49, no. 7, pp. 53–61, Jul. 2016.
- [42] B. Rinner, L. Esterle, J. Simonjan, G. Nebhay, R. Pflugfelder, G. Fernandez Dominguez, and P. R. Lewis, "Self-Aware and Self-Expressive Camera Networks," *Computer (Long. Beach. Calif.)*, vol. 48, no. 7, pp. 21–28, Jul. 2015.
- [43] N. Dutt, A. Jantsch, and S. Sarma, "Self-aware Cyber-Physical Systems-on-Chip," in *2015 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2015, pp. 46–50.
- [44] A. Elhabbash, R. Bahsoon, P. Tino, and P. R. Lewis, "Self-Adaptive Volunteered Services Composition through Stimulus- and Time-Awareness," in *2015 IEEE International Conference on Web Services*, 2015, pp. 57–64.

- [45] U. Neisser, "Five kinds of self-knowledge," *Philos. Psychol.*, vol. 1, no. 1, pp. 35–59, Jan. 1988.
- [46] U. NEISSER, "The Roots of Self-Knowledge: Perceiving Self, It, and Thou," *Ann. N. Y. Acad. Sci.*, vol. 818, no. 1 Self Across P, pp. 19–33, Jun. 1997.
- [47] J. Torresen, C. Plessl, and X. Yao, "Self-Aware and Self-Expressive Systems," *Computer (Long. Beach. Calif.)*, vol. 48, no. 7, pp. 18–20, Jul. 2015.
- [48] F. Zambonelli, N. Bicocchi, G. Cabri, L. Leonardi, and M. Puviani, "On self-adaptation, self-expression, and self-awareness in autonomic service component ensembles," *Proc. - 2011 5th IEEE Conf. Self-Adaptive Self-Organizing Syst. Work. SASOW 2011*, pp. 108–113, 2011.
- [49] M. Amoretti and S. Cagnoni, "Toward collective self-Awareness and self-Expression in distributed systems," *Computer (Long. Beach. Calif.)*, vol. 48, no. 7, pp. 29–36, 2015.
- [50] J. C. Sanmiguel, K. Shoop, C. Micheloni, and G. L. Foresti, "Self-Reconfigurable Smart Camera Networks," *Computer (Long. Beach. Calif.)*, vol. 47, no. 5, pp. 67–73, May 2014.
- [51] L. Esterle, P. R. Lewis, H. Caine, X. Yao, and B. Rinner, "CamSim: A Distributed Smart Camera Network Simulator," in *2013 IEEE 7th International Conference on Self-Adaptation and Self-Organizing Systems Workshops*, 2013, pp. 19–20.

- [52] M. Prasanth, P. Srinivasan, and M. Pendyala, "Automated waste clearance: Street-wise cleanliness," in *2013 International Conference on Human Computer Interactions (ICHCI)*, 2013, pp. 1–5.
- [53] S. Park, J. Jeong, and C. S. Hong, "DNS Configuration in IPv6: Approaches, Analysis, and Deployment Scenarios," *IEEE Internet Comput.*, vol. 17, no. 4, pp. 48–56, Jul. 2013.
- [54] J. Jeong, S. Park, and L. Beloeli, "IPv6 Router Advertisement Options for DNS Configuration," *IETF RFC 6106*, 2010. [Online]. Available: <https://tools.ietf.org/html/rfc6106>.
- [55] Y. Zhang, Y. Bao, S. Zhao, J. Chen, and J. Tang, "Identifying Node Importance by Combining Betweenness Centrality and Katz Centrality," in *2015 International Conference on Cloud Computing and Big Data (CCBD)*, 2015, pp. 354–357.
- [56] Y. Guo, H. Zhu, and L. Yang, "Service-oriented network virtualization architecture for Internet of Things," *China Commun.*, vol. 13, no. 9, pp. 163–172, Sep. 2016.
- [57] J. Lutz, C. J. Colbourn, and V. R. Syrotiuk, "ATLAS: Adaptive Topology- and Load-Aware Scheduling," *IEEE Trans. Mob. Comput.*, vol. 13, no. 10, pp. 2255–2268, Oct. 2014.
- [58] X. Wei, H. Li, K. Yang, and L. Zou, "Topology-Aware Partial Virtual Cluster

- Mapping Algorithm on Shared Distributed Infrastructures," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 10, pp. 2721–2730, Oct. 2014.
- [59] J. Chen, Y. Tang, Y. Dong, J. Xue, Z. Wang, and W. Zhou, "Reducing Static Energy in Supercomputer Interconnection Networks Using Topology-Aware Partitioning," *IEEE Trans. Comput.*, vol. 65, no. 8, pp. 2588–2602, Aug. 2016.
- [60] Q. Duan, N. Ansari, and M. Toy, "Software-defined network virtualization: an architectural framework for integrating SDN and NFV for service provisioning in future networks," *IEEE Netw.*, vol. 30, no. 5, pp. 10–16, Sep. 2016.
- [61] I. Khan, F. Belqasmi, R. Glitho, N. Crespi, M. Morrow, and P. Polakos, "Wireless sensor network virtualization: A survey," *IEEE Commun. Surv. Tutorials*, vol. 18, no. 1, pp. 553–576, 2016.
- [62] R. Haggarty, *Discrete Mathematics for Computing*, First Edit. Addison-Wesley, 2002.
- [63] P. Grossman, *Discrete mathematics for computing*, vol. 2nd. 2002.
- [64] W. Maharani, Adiwijaya, and A. A. Gozali, "Degree centrality and eigenvector centrality in twitter," in *2014 8th International Conference on Telecommunication Systems Services and Applications (TSSA)*, 2014, vol. 9, no. 2, pp. 1–5.

- [65] S. Segarra and A. Ribeiro, "Stability and Continuity of Centrality Measures in Weighted Graphs," *IEEE Trans. Signal Process.*, vol. 64, no. 3, pp. 543–555, Feb. 2016.
- [66] T. Wang, H. Krim, and Y. Viniotis, "Analysis and Control of Beliefs in Social Networks," *IEEE Trans. Signal Process.*, vol. 62, no. 21, pp. 5552–5564, Nov. 2014.
- [67] W. Wang and C. Y. Tang, "Distributed estimation of closeness centrality," in *2015 54th IEEE Conference on Decision and Control (CDC)*, 2015, no. Cdc, pp. 4860–4865.
- [68] E. Cohen, "All-Distances Sketches, Revisited: HIP Estimators for Massive Graphs Analysis," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 9, pp. 2320–2334, Sep. 2015.
- [69] J. Gao, Q. Zhao, W. Ren, A. Swami, R. Ramanathan, and A. Bar-Noy, "Dynamic Shortest Path Algorithms for Hypergraphs," *IEEE/ACM Trans. Netw.*, vol. 23, no. 6, pp. 1805–1817, Dec. 2015.
- [70] I. Claros, R. Cobos, and C. A. Collazos, "An Approach Based on Social Network Analysis Applied to a Collaborative Learning Experience," *IEEE Trans. Learn. Technol.*, vol. 9, no. 2, pp. 190–195, Apr. 2016.
- [71] E. E. Santos, J. Korah, V. Murugappan, and S. Subramanian, "Efficient Anytime Anywhere Algorithms for Closeness Centrality in Large and Dynamic

- Graphs,” in *2016 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, 2016, pp. 1821–1830.
- [72] L. C. Freeman, “Centrality in social networks conceptual clarification,” *Soc. Networks*, vol. 1, no. 3, pp. 215–239, Jan. 1978.
- [73] A. Bavelas, “Communication Patterns in Task-Oriented Groups,” *J. Acoust. Soc. Am.*, vol. 22, no. 6, pp. 725–730, Nov. 1950.
- [74] W. Wang and C. Y. Tang, “Distributed computation of classic and exponential closeness on tree graphs,” in *2014 American Control Conference*, 2014, pp. 2090–2095.
- [75] S. P. Borgatti, “Centrality and network flow,” *Soc. Networks*, vol. 27, no. 1, pp. 55–71, Jan. 2005.
- [76] A. Bertrand and M. Moonen, “Seeing the Bigger Picture: How Nodes Can Learn Their Place Within a Complex Ad Hoc Network Topology,” *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 71–82, May 2013.
- [77] E. Atsan and O. Ozkasap, “Applicability of eigenvector centrality principle to data replication in MANETs,” in *2007 22nd international symposium on computer and information sciences*, 2007, pp. 1–6.
- [78] J. P. Keener, “The Perron-Frobenius theorem and the ranking of football teams,” *SIAM review*, vol. 35, no. 1, pp. 80–93, 1993.

- [79] P. Sattari, M. Kurant, A. Anandkumar, A. Markopoulou, and M. G. Rabbat, "Active learning of multiple source multiple destination topologies," *IEEE Trans. Signal Process.*, vol. 62, no. 8, pp. 1926–1937, 2014.
- [80] M. G. Rabbat, M. J. Coates, and R. D. Nowak, "Multiple-Source Internet Tomography," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 12, pp. 2221–2234, Dec. 2006.
- [81] Shelby Z., K. Hartke, and C. Bormann, "The Constrained Application Protocol (CoAP)," *RFC 7252 (Proposed Standard)*, Internet Engineering Task Force, 2014. [Online]. Available: <http://tools.ietf.org/html/rfc7252>. [Accessed: 29-Jan-2016].
- [82] S. O. Ajiboye, P. Birch, C. Chatwin, and R. Young, "Hierarchical Video Surveillance Architecture - A Chassis for Video Big Data Analytics and Exploration," in *Proc. SPIE 9407*, 2015, vol. 9407, pp. 1–10.
- [83] A. Yachir, Y. Amirat, A. Chibani, and N. Badache, "Event-Aware Framework for Dynamic Services Discovery and Selection in the Context of Ambient Intelligence and Internet of Things," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 1, pp. 85–102, Jan. 2016.
- [84] B. Carballido Villaverde, R. D. P. Alberola, A. J. Jara, S. Fedor, S. K. Das, and D. Pesch, "Service discovery protocols for constrained machine-to-machine communications," *IEEE Commun. Surv. Tutorials*, vol. 16, no. 1, pp. 41–60, 2014.

- [85] A. Zisman, G. Spanoudakis, J. Dooley, and I. Siveroni, "Proactive and Reactive Runtime Service Discovery: A Framework and Its Evaluation," *IEEE Trans. Softw. Eng.*, vol. 39, no. 7, pp. 954–974, Jul. 2013.
- [86] S. Demers, M. A. Fecko, Y.-J. Lin, D. Shur, S. Samtani, K. Sinkar, and J. Chapin, "Scalable Registration and Discovery of Devices in Low-Bandwidth Tactical Networks," in *MILCOM 2013 - 2013 IEEE Military Communications Conference*, 2013, pp. 550–555.
- [87] M. F. Bari, M. R. Haque, R. Ahmed, R. Boutaba, and B. Mathieu, "A naming scheme for P2P web hosting," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 9, pp. 299–309, 2013.
- [88] V. Kantere, S. Skiadopoulos, and T. Sellis, "Storing and Indexing Spatial Data in P2P Systems," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 2, pp. 287–300, Feb. 2009.
- [89] Y. Tang, S. Zhou, and J. Xu, "LIGHT: A Query-Efficient Yet Low-Maintenance Indexing Scheme over DHTs," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 1, pp. 59–75, Jan. 2010.
- [90] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," in *Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications - SIGCOMM '01*, 2001, vol. 11, no. 1, pp. 149–160.

- [91] M. Picone, M. Amoretti, and F. Zanichelli, "GeoKad: A P2P distributed localization protocol," in *2010 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, 2010, pp. 800–803.
- [92] M. Picone, M. Amoretti, and F. Zanichelli, "Proactive neighbor localization based on distributed geographic table," *Int. J. Pervasive Comput. Commun.*, vol. 7, no. 3, pp. 240–263, Sep. 2011.
- [93] H. T. Nguyen and B. Bhanu, "Tracking pedestrians with bacterial foraging optimization swarms," in *2011 IEEE Congress of Evolutionary Computation (CEC)*, 2011, pp. 491–495.
- [94] C. Rother and A. Blake, "'GrabCut'-Interactive Foreground Extraction using Iterated Graph Cuts," *ACM Trans. Graph. - Proc. ACM SIGGRAPH 2004*, vol. 1, no. 212, pp. 309–314, 2004.
- [95] Y. Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images," in *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, 2001, vol. 1, pp. 105–112.
- [96] M. Björkman and D. Kragic, "Active 3D scene segmentation and detection of unknown objects," in *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 3114–3120.

- [97] L. Wang, C. Zhang, and R. Yang, "TofCut : Towards Robust Real-time Foreground Extraction Using a Time-of-Flight Camera," *15th Int. Symp. 3D Data Process. Vis. Transm.*, 2010.
- [98] Bizhong Wei, Ning Ouyang, YueLin Chen, and Xiaodong Cai, "Automatic color blob segmentation and fast arbitrary shape tracking," in *5th International Conference on Visual Information Engineering (VIE 2008)*, 2008, pp. 408–413.
- [99] H. Azhar and A. Amer, "Classification of surveillance video objects using chaotic series," *IET Image Process.*, vol. 6, no. 7, p. 919, 2012.
- [100] A. Karpenko and P. Aarabi, "Tiny Videos: A Large Data Set for Nonparametric Video Retrieval and Frame Classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 3, pp. 618–630, Mar. 2011.
- [101] D. Simonnet, E. Turkbeyler, S. A. Velastin, and J. Orwell, "Backgroundless detection of pedestrians in cluttered conditions based on monocular images: a review," *IET Comput. Vis.*, vol. 6, no. 6, pp. 540–550, 2012.
- [102] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian Detection: An Evaluation of the State of the Art," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 743–761, Apr. 2012.
- [103] M. Grundmann, V. Kwatra, M. Han, and I. Essa, "Efficient hierarchical graph-based video segmentation," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2141–2148.

- [104] M. Taj and A. Cavallaro, "Distributed and Decentralized Multicamera Tracking," *IEEE Signal Process. Mag.*, vol. 28, no. 3, pp. 46–58, May 2011.
- [105] S. Chen, Y. Li, and N. M. Kwok, *Active vision in robotic systems: A survey of recent developments*, vol. 30, no. 11. 2011.
- [106] Bi Song and A. K. Roy-Chowdhury, "Robust Tracking in A Camera Network: A Multi-Objective Optimization Framework," *IEEE J. Sel. Top. Signal Process.*, vol. 2, no. 4, pp. 582–596, Aug. 2008.
- [107] C. Lijun and H. Kaiqi, "Video-based crowd density estimation and prediction system for wide-area surveillance," *China Commun.*, vol. 10, no. 5, pp. 79–88, May 2013.
- [108] J. Hao, G. Wang, B. Seo, and R. Zimmermann, "Point of Interest Detection and Visual Distance Estimation for Sensor-Rich Video," *IEEE Trans. Multimed.*, vol. 16, no. 7, pp. 1929–1941, Nov. 2014.
- [109] X. Li, N. Santoro, and I. Stojmenovic, "Localized distance-sensitive service discovery in wireless sensor and actor networks," *Comput. IEEE Trans. ...*, vol. 58, no. September, pp. 1275–1288, 2009.
- [110] C. Wang, H. Liu, and L. Ma, "Depth Motion Detection—A Novel RS-Trigger Temporal Logic based Method," *IEEE Signal Process. Lett.*, vol. 21, no. 6, pp. 717–721, Jun. 2014.

- [111] Jungong Han, Ling Shao, Dong Xu, and J. Shotton, "Enhanced Computer Vision With Microsoft Kinect Sensor: A Review," *IEEE Trans. Cybern.*, vol. 43, no. 5, pp. 1318–1334, Oct. 2013.
- [112] A. Bedagkar-Gala and S. K. Shah, "A survey of approaches and trends in person re-identification," *Image Vis. Comput.*, vol. 32, no. 4, pp. 270–286, Apr. 2014.
- [113] S. Wu, Y.-C. Chen, X. Li, A.-C. Wu, J.-J. You, and W.-S. Zheng, "An enhanced deep feature representation for person re-identification," in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2016, pp. 1–8.
- [114] R. Satta, "Appearance Descriptors for Person Re-identification: a Comprehensive Review," 2013. [Online]. Available: <https://arxiv.org/pdf/1307.5748.pdf>.
- [115] D. Chen, Z. Yuan, B. Chen, and N. Zheng, "Similarity Learning with Spatial Constraints for Person Re-identification," *2016 IEEE Conf. Comput. Vis. Pattern Recognit.*, no. 3, 2016.
- [116] N. Z. De Cheng, Yihong Gong, Sanping Zhou, Jinjun Wang, "Person Re-Identification by An Multi-Channel Parts-Based CNN with Improved Triplet Loss Function," *Cvpr*, 2016.
- [117] S. Paisitkriangkrai, C. Shen, and A. Van Den Hengel, "Learning to rank in person re-identification with metric ensembles," *Proc. IEEE Comput. Soc.*

- Conf. Comput. Vis. Pattern Recognit.*, vol. 07–12–June, pp. 1846–1855, 2015.
- [118] Z. Mingyong, Z. Wu, C. Tian, Z. Lei, and H. Lei, “Efficient person re-identification by hybrid spatiogram and covariance descriptor,” *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, vol. 2015–Octob, pp. 48–56, 2015.
- [119] S. Z. Chen, C. C. Guo, and J. H. Lai, “Deep Ranking for Person Re-Identification via Joint Representation Learning,” *IEEE Trans. Image Process.*, vol. 25, no. 5, pp. 2353–2367, 2016.
- [120] X. Liu, H. Wang, Y. Wu, J. Yang, and M. H. Yang, “An ensemble color model for human re-identification,” *Proc. - 2015 IEEE Winter Conf. Appl. Comput. Vision, WACV 2015*, pp. 868–875, 2015.
- [121] R. Xue, Z.-S. Wu, and A.-N. Bai, “Application of Cloud Storage in Traffic Video Detection,” in *2011 Seventh International Conference on Computational Intelligence and Security*, 2011, pp. 1294–1297.
- [122] S. Dey, A. Chakraborty, S. Naskar, and P. Misra, “Smart city surveillance: Leveraging benefits of cloud data stores,” in *37th Annual IEEE Conference on Local Computer Networks -- Workshops*, 2012, no. 978-1-4673-2130-3, pp. 868–876.
- [123] Y. Huo, H. Wang, and L. Hu, “A Cloud Storage Architecture Model for Data-Intensive Applications,” in *2011 International Conference on Computer and*

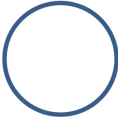






Management (CAMAN), 2011, no. 61073009, pp. 26–29.

- [124] J. Wu, L. Ping, X. Ge, Y. Wang, and J. Fu, “Cloud Storage as the Infrastructure of Cloud Computing,” *2010 Int. Conf. Intell. Comput. Cogn. Informatics*, pp. 380–383, Jun. 2010.
- [125] F. Belqasmi, R. Glitho, and C. Fu, “RESTful web services for service provisioning in next-generation networks: a survey,” *IEEE Commun. Mag.*, vol. 49, no. 12, pp. 66–73, Dec. 2011.
- [126] J. Liu, M. Tang, Z. Zheng, X. (Frank) Liu, and S. Lyu, “Location-Aware and Personalized Collaborative Filtering for Web Service Recommendation,” *IEEE Trans. Serv. Comput.*, vol. 9, no. 5, pp. 686–699, Sep. 2016.
- [127] F. Belqasmi, J. Singh, S. Y. Bani Melhem, and R. H. Glitho, “SOAP-Based vs. RESTful Web Services: A Case Study for Multimedia Conferencing,” *IEEE Internet Comput.*, vol. 16, no. 4, pp. 54–63, Jul. 2012.
- [128] J. Lee, S.-J. Lee, and P.-F. Wang, “A Framework for Composing SOAP, Non-SOAP and Non-Web Services,” *IEEE Trans. Serv. Comput.*, vol. 8, no. 2, pp. 240–250, Mar. 2015.
- [129] Z. Shelby, “Embedded web services,” *IEEE Wirel. Commun.*, vol. 17, no. 6, pp. 52–57, Dec. 2010.
- [130] C. Pautasso, O. Zimmermann, and F. Leymann, “Restful web services vs.

- 'big'web services: making the right architectural decision," in *Proceeding of the 17th international conference on World Wide Web*, 2008, pp. 805–814.
- [131] K. Lawrence, C. Kaler, A. Nadalin, R. Monzillo, and P. Hallam-baker, "Web Services Security: SOAP Message Security 1.1 (WS-Security 2004)," *Security*, vol. 2003, no. February, p. 76, 2006.
- [132] C. Tao, X. Ling, S. Guofeng, Y. Hongyong, and H. Quanyi, "Architecture for Monitoring Urban Infrastructure and Analysis Method for a Smart-Safe City," in *2014 Sixth International Conference on Measuring Technology and Mechatronics Automation*, 2014, pp. 151–154.
- [133] W. Paper, "The Cisco SONA Architectural Model in Unified Communications: A Solid Foundation for the Collaborative Innovative Enterprise," 2008.
[Online]. Available:
http://www.cisco.com/c/en/us/solutions/collateral/enterprise-networks/branch/White_paper_C11-473760.html.
- [134] R. Perlman and C. Kaufman, "Hierarchical networks with Byzantine Robustness," in *2011 Third International Conference on Communication Systems and Networks (COMSNETS 2011)*, 2011, pp. 1–11.
- [135] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Networks*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.
- [136] Shifeng Fang, Li Da Xu, Yunqiang Zhu, Jiaerheng Ahati, Huan Pei, Jianwu Yan,

- and Zhihui Liu, "An Integrated System for Regional Environmental Monitoring and Management Based on Internet of Things," *IEEE Trans. Ind. Informatics*, vol. 10, no. 2, pp. 1596–1605, May 2014.
- [137] Y. Shavitt and N. Zilberman, "A geolocation databases study," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 10, pp. 2044–2056, 2011.
- [138] R. G. Sargent, "Verification and validation of simulation models," in *Proceedings of the 2011 Winter Simulation Conference (WSC)*, 2011, pp. 183–198.
- [139] S. Schlesinger, "Terminology for model credibility," *Simulation*, vol. 32, no. 3, pp. 103–104, Mar. 1979.
- [140] R. Prates and W. R. Schwartz, "Kernel Cross-View Collaborative Representation based Classification for Person Re-Identification," *arXiv*. 2016.
- [141] M. Matsumoto and T. Nishimura, "Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator," *ACM Trans. Model. Comput. Simul.*, vol. 8, no. 1, pp. 3–30, Jan. 1998.

Appendix A: BPMN Notations Used in this thesis

| Notation symbol | Name and description |
|---|---|
|  | Start - used to signify the beginning process/sub-process. |
|  | Default flow – Used to signify the default flow for a decision. |
|  | Task – Used to describe a single action in a process. |
|  | Event Sub-Process – Used to describe events that occur in a sub-process. |
|  | Pool – represents participants - it is used to set the boundary of a business processes and it may only contain 1 process. |
|  | Exclusive Gateway – Used to assess the state of a business process and follows a path based on the conditions of the business process. |
|  | Data Object – represents a data-based object. |

Appendix B: Database Schemas



Appendix C: SOAP XML Message (Metadata Object)

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:FVSA" xmlns:ns2="http://xml.apache.org/xml-soap" xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <ns1:callResponse>
      <callReturn SOAP-ENC:arrayType="ns2:Map[32]" xsi:type="SOAP-ENC:Array">
        <item xsi:type="SOAP-ENC:Array">
          <item>
            <key xsi:type="xsd:string">id</key>
            <value xsi:type="xsd:string">1</value>
          </item>
          <item>
            <key xsi:type="xsd:string">camera_id</key>
            <value xsi:type="xsd:string">1</value>
          </item>
          <item>
            <key xsi:type="xsd:string">camnet_id</key>
            <value xsi:type="xsd:string">1</value>
          </item>
          <item>
            <key xsi:type="xsd:string">camnet_camera_id</key>
            <value xsi:type="xsd:string">1</value>
          </item>
          <item>
            <key xsi:type="xsd:string">ip_address</key>
            <value xsi:type="xsd:string">192.168.1.5</value>
          </item>
          <item>
            <key xsi:type="xsd:string">camera_name</key>
            <value xsi:type="xsd:string">Engineering, Floor 1, University of Sussex
            </value>
          </item>
          <item>
            <key xsi:type="xsd:string">camera_description</key>
            <value xsi:type="xsd:string">Engineering, Floor 1, University of Sussex
            </value>
          </item>
          <item>
            <key xsi:type="xsd:string">mds_url</key>
            <value xsi:type="xsd:string"></value>
          </item>
          <item>
            <key xsi:type="xsd:string">latitude</key>
            <value xsi:type="xsd:string"></value>
          </item>
          <item>
            <key xsi:type="xsd:string">longitude</key>
            <value xsi:type="xsd:string"></value>
          </item>
          <item>
            <key xsi:type="xsd:string">direction</key>
            <value xsi:type="xsd:string"></value>
          </item>
          <item>
            <key xsi:type="xsd:string">camera_projection</key>
            <value xsi:type="xsd:string"></value>
          </item>
          <item>
            <key xsi:type="xsd:string">owner_name</key>
            <value xsi:type="xsd:string"></value>
          </item>
        </item>
      </callReturn>
    </ns1:callResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```



```

</item>
<item>
  <key xsi:type="xsd:string">owner_email</key>
  <value xsi:type="xsd:string">test@test.com</value>
</item>
<item>
  <key xsi:type="xsd:string">owner_phone</key>
  <value xsi:type="xsd:string">12345678</value>
</item>
<item>
  <key xsi:type="xsd:string">privacy</key>
  <value xsi:type="xsd:string">full</value>
</item>
<item>
  <key xsi:type="xsd:string">address1</key>
  <value xsi:type="xsd:string">Address 1</value>
</item>
<item>
  <key xsi:type="xsd:string">address2</key>
  <value xsi:type="xsd:string">Address 2</value>
</item>
<item>
  <key xsi:type="xsd:string">city</key>
  <value xsi:type="xsd:string">Brighton</value>
</item>
<item>
  <key xsi:type="xsd:string">region</key>
  <value xsi:type="xsd:string">East Sussex</value>
</item>
<item>
  <key xsi:type="xsd:string">postcode</key>
  <value xsi:type="xsd:string">BN2</value>
</item>
<item>
  <key xsi:type="xsd:string">country</key>
  <value xsi:type="xsd:string">GB</value>
</item>
<item>
  <key xsi:type="xsd:string">metadata_frequency</key>
  <value xsi:type="xsd:string"></value>
</item>
<item>
  <key xsi:type="xsd:string">status</key>
  <value xsi:type="xsd:string">0</value>
</item>
<item>
  <key xsi:type="xsd:string">created_at</key>
  <value xsi:type="xsd:string">2016-10-17 16:09:57</value>
</item>
<item>
  <key xsi:type="xsd:string">updated_at</key>
  <value xsi:nil="true" />
</item>
</item>
</callReturn>
</ns1:callResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```