Using wireless multimedia sensor networks for 3D scene acquisition and reconstruction
Title: Using wireless multimedia sensor networks for 3D scene acquisition and reconstruction

Keywords: Wireless Multimedia Sensor Networks, Disparity Map, 3D Stereo Vision, 3D Scene Reconstruction.

Abstract:
Nowadays, the WMSNs are promising for different applications and fields, specially with the development of the IoT and cheap efficient camera sensors. The stereo vision is also very important for multiple purposes like Cinematography, games, Virtual Reality, Augmented Reality, etc. This thesis aim to develop a 3D scene reconstruction system that proves the concept of using multiple view stereo disparity maps in the context of WMSNs. Our work can be divided in three parts. The first one concentrates on studying all WMSNs applications, components, topologies, constraints and limitations. Adding to this stereo vision disparity map calculations methods in order to choose the best method(s) to make a 3d reconstruction on WMSNs with low cost in terms of complexity and power consumption. In the second part, we experiment and simulate different disparity map calculations on a couple of nodes by changing scenarios (indoor and outdoor), coverage distances, angles, number of nodes and algorithms. In the third part, we propose a tree-based network model to compute accurate disparity maps on multi-layer camera sensor nodes that meets the server needs to make a 3d scene reconstruction of the scene or object of interest. The results are acceptable and ensure the proof of the concept to use disparity maps in the context of WMSNs.
# Contents

Table of Contents 4

List of Figures 6

List of Tables 6

List of Algorithms 6

List of Abbreviations 7

Acknowledgments 9

## 1 Introduction

1.1 General Introduction ............................................. 11
1.2 Objectives and problem statement ............................... 13
1.3 Main contributions ............................................. 14
1.4 Thesis Outline .................................................. 15
1.5 Publications ................................................... 18

## I WMSN, Disparity Map and 3D Scene Reconstruction

## 2 Understanding WMSN

2.1 Introduction ................................................... 21
2.2 WMSNs Applications ............................................. 21
2.3 WMSN Components ............................................. 22
2.4 WMSN network architecture ..................................... 23
2.5 WMSN challenges ............................................... 25
  2.5.1 Energy consumption ....................................... 25
  2.5.2 High bandwidth demand ................................... 26
  2.5.3 Architecture scalability and protocol flexibility ............ 26
  2.5.4 Localized processing and data fusion ....................... 26
# CONTENTS

4.1 Introduction ......................................................... 51
4.2 Multiple View 3D Scene Reconstruction .......................... 51
4.3 Related Work ....................................................... 53
4.4 Conclusion ......................................................... 55

II Disparity Map Computation in WMSN ............................... 57

5 Using Disparity map techniques in WMSN .......................... 59
  5.1 Introduction ....................................................... 59
  5.2 Stereo Vision methods applicable on WMSNs ....................... 60
  5.3 Experimentations Related to Complexity .......................... 61
  5.4 Wireless Multimedia Sensor Network deployment for disparity map calcu-
      lation .......................................................... 66
  5.5 Experimentations Related to Coverage ............................ 71
  5.6 Conclusion ....................................................... 75

III 3D Reconstruction Model using Disparity Map in WMSN ............ 77

6 Disparity Based Reconstruction Model ............................... 79
  6.1 Introduction ....................................................... 79
  6.2 Problematic ...................................................... 79
  6.3 Proposed Framework .............................................. 81
     6.3.1 Disparity computation based on binary tree distribution .......... 81
     6.3.2 Discarding Similar Disparity .................................. 84
        6.3.2.1 Monitoring algorithm .................................... 84
     6.3.3 Disparity classification using Principal Component Analysis ....... 84
  6.4 Disparity based reconstruction .................................... 90
     6.4.1 Triangulation ............................................... 90
  6.5 Experimentation Results ......................................... 94
     6.5.1 Nodes Distribution Scenarios ................................ 94
     6.5.2 Tree topology .............................................. 95
     6.5.3 Another sensor deployment topology .......................... 99
     6.5.4 Indoor scene .............................................. 103
  6.6 Conclusion ...................................................... 105
IV Conclusion

7 Conclusion

7.1 Thesis Summary, Back to the Main Contributions

7.2 Future Works and Perspectives

Bibliographie
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Limitations of WMSNs</td>
<td>12</td>
</tr>
<tr>
<td>1.2</td>
<td>Phases of our system</td>
<td>13</td>
</tr>
<tr>
<td>1.3</td>
<td>Complete pipeline of the contributions with reference to the thesis chapters.</td>
<td>17</td>
</tr>
<tr>
<td>2.1</td>
<td>General WMSN layout</td>
<td>21</td>
</tr>
<tr>
<td>2.2</td>
<td>WMSN architectures</td>
<td>24</td>
</tr>
<tr>
<td>3.1</td>
<td>Disparity map computation process</td>
<td>29</td>
</tr>
<tr>
<td>4.1</td>
<td>Multiple View 3D Scene Reconstruction example</td>
<td>52</td>
</tr>
<tr>
<td>4.2</td>
<td>Triangulation</td>
<td>52</td>
</tr>
<tr>
<td>4.3</td>
<td>CopyMe3D: Scanning and Printing Persons in 3D [81]</td>
<td>54</td>
</tr>
<tr>
<td>5.1</td>
<td>Phases of our system</td>
<td>60</td>
</tr>
<tr>
<td>5.2</td>
<td>Different views of the scene</td>
<td>62</td>
</tr>
<tr>
<td>5.3</td>
<td>Disparity Map using SAD and SSD</td>
<td>63</td>
</tr>
<tr>
<td>5.4</td>
<td>Processing time per resolution, views and number of pairs</td>
<td>64</td>
</tr>
<tr>
<td>5.5</td>
<td>Proposed model</td>
<td>66</td>
</tr>
<tr>
<td>5.6</td>
<td>Occlusion and overlapping problems in video sensor network coverage</td>
<td>67</td>
</tr>
<tr>
<td>5.7</td>
<td>Stereo vision system baseline and orientation</td>
<td>69</td>
</tr>
<tr>
<td>5.8</td>
<td>Stereo Coefficient principle</td>
<td>70</td>
</tr>
<tr>
<td>5.9</td>
<td>Indoor scene: elderly man walking in his private bedroom</td>
<td>71</td>
</tr>
<tr>
<td>5.10</td>
<td>Outdoor scene: random populations walking in city streets</td>
<td>72</td>
</tr>
<tr>
<td>5.11</td>
<td>Camera positions from top</td>
<td>72</td>
</tr>
<tr>
<td>5.12</td>
<td>Applying SSIM on a computed disparity map of two different views</td>
<td>73</td>
</tr>
<tr>
<td>6.1</td>
<td>General framework for 3D reconstruction based on DM</td>
<td>86</td>
</tr>
<tr>
<td>6.2</td>
<td>Binary tree applied on the system</td>
<td>97</td>
</tr>
<tr>
<td>6.3</td>
<td>Algorithm flowchart</td>
<td>88</td>
</tr>
<tr>
<td>6.4</td>
<td>Monitoring Principle</td>
<td>39</td>
</tr>
<tr>
<td>6.5</td>
<td>Triangulation</td>
<td>91</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>6.7</td>
<td>Reconstructed scene from left and right images</td>
<td>93</td>
</tr>
<tr>
<td>6.8</td>
<td>Our tree topology</td>
<td>94</td>
</tr>
<tr>
<td>6.9</td>
<td>Tree topology for house monitoring</td>
<td>96</td>
</tr>
<tr>
<td>6.10</td>
<td>Real house scene image set from outdoor</td>
<td>97</td>
</tr>
<tr>
<td>6.11</td>
<td>House disparity maps</td>
<td>98</td>
</tr>
<tr>
<td>6.12</td>
<td>Tree scenario PCA results</td>
<td>98</td>
</tr>
<tr>
<td>6.13</td>
<td>House 3D reconstruction results</td>
<td>99</td>
</tr>
<tr>
<td>6.14</td>
<td>Real car scene</td>
<td>100</td>
</tr>
<tr>
<td>6.15</td>
<td>Car disparity maps</td>
<td>101</td>
</tr>
<tr>
<td>6.16</td>
<td>Ring PCA results</td>
<td>101</td>
</tr>
<tr>
<td>6.17</td>
<td>Indoor scene</td>
<td>103</td>
</tr>
<tr>
<td>6.18</td>
<td>Indoor scene disparity maps</td>
<td>104</td>
</tr>
<tr>
<td>6.19</td>
<td>Mesh PCA results</td>
<td>104</td>
</tr>
</tbody>
</table>
WSN . . . . Wireless Sensor Network
WMSN . . Wireless Multimedia Sensor Network
CMOS . . . . Complementary Metal Oxide Semiconductor
HVS . . . . Human Visual System
QoS . . . . Quality Of Service
SAD . . . . Sum of Absolute Differences
SSD . . . . Sum of Squared Differences
PSNR . . . Peak signal-to-noise ratio
SSIM . . . . Structural Similarity Index
MSE . . . . Mean Squared Error
PCA . . . . Principal component analysis
AD . . . . Absolute Differences
GPU . . . . Graphical Processing Unit
SD . . . . . Squared Differences
SIFT . . . Scale Invariant Feature Transform
SMW . . . . Symmetric Multi Window
MRF . . . . Markov Random Fields
NCC . . . . Normalized Cross Correlation
DSP . . . . Digital Signal Processor
RT . . . . Rank Transform
CT . . . . Census Transform
FW . . . . Fixed-size Window
MW . . . . Multiple Windows
AW . . . . Adaptive Windows
ASW . . . . Adaptive Support Weights
WTA . . . . Winner Takes All
abbreviations

BP ....... Belief Propagation
SGM ...... Semi-Global Matching
IoT ...... Internet of Things
SFM ...... Structure From Motion
SFS ...... Shape From Silhouette
TOF ...... Time of Flight
FOV ...... Field of View
SSIM ...... Structural Similarity
PSNR ..... Peak Signal-to-Noise Ratio
ACKNOWLEDGMENTS

This thesis is dedicated to my parents, specially my father Atty. Charbel TANNOURY, my mother Maguy and sisters Carmen and Josianne who they didn’t stop supporting me during all my life stages.

A special thanks to my role model, my uncle Abbot Paul TANNOURY, who helped and advised me in all my life’s phases.

I would like to thank my supervisors, Prof. GUYEUX, Dr. MAKHOUL and Dr. DARAZI for their guidance and support.

I would like also to thank Reverend Rector Germanos GERMANOS, Reverend Rector Michel JALEKH, Reverend Fathers Raymond HACHEM, Joe BOU JAOUDE, Toufik MAATOUK, and Mr. Tony Reaidy from the Antonine University for their continuous motivation to pursue my PhD.

Very special thanks to my colleagues in the world of research, who I met in different conferences and events. Without missing the authors who I cited, without them we cannot achieve the required improvements.

I want to thank my friends, my colleagues at Antonine University who I enjoy working and collaborating with in different projects and challenges.
Environmental monitoring and surveillance are gaining more attention these days. Hence, remote monitoring for quality control and event prevention is a requirement. Wireless Sensor Networks (WSN) was introduced as a solution for environmental surveillance [76], providing continuous observation of a place or an ongoing activity in order to gather information of a particular physical phenomenon. Basically, collected data associated with WSN are related to acoustics, light, humidity, temperature, imaging, and seismic, etc. A Wireless Multimedia Sensor Network (WMSN) is a subset of WSN, it is composed of interconnected nodes, mostly cheap CMOS (Complementary Metal Oxide Semiconductor) cameras, interacting with each other and capturing multimedia data (audio/video) describing their environment. During the last few years, many researchers focused on WMSNs because of their easy implementation related to their small size, their hardware low cost and their diversity of applications. Many challenges arise in WMSNs like high bandwidth demand, low energy consumption, efficient processing algorithms, and application-specific Quality of Service (QoS).

However, nowadays many activities in real time monitoring require more insight on an ongoing activity in order to gather information related to the depth of an object or if a building is subject to a progressive deformation. This particular information is related to the 3D representation of an object or the environment and cannot be delivered with traditional collected data. Most of existing 3D reconstruction techniques [69] use advanced and expensive cameras, centralized server schema, and network topology to reconstruct all scene objects in an acceptable period of time. Most of these techniques are not in real-time.

Stereo Vision is used to represent a detailed environment via multiple algorithmic methods to define the surrounding world we see, based on multiple images taken from different angles. A Stereo Vision system is composed by two stereo cameras (left camera and right camera) that capture simultaneously two images from the same scene. The pair of images is processed to calculate a disparity map that recovers the Depth information. The disparity map is a gray-scale image resulted from a Stereo Correspondence algorithm in order to represent corresponding pixels that are horizontally shifted between the left image and the right one. There are many methods and algorithms to solve this problem that differ in accuracy and time consumption.

Our contribution is to introduce the use of the disparity map that is computed via two or several images in order to monitor the depth information in an object or another phe-
nomenon under surveillance using WMSN.

Sensor nodes are driven by batteries and have very low energy resources, so the primary limiting constraint for WMSNs is the energy consumption that affects the network lifetime. The radio consumes the majority of the system energy in WMSNs (listening & transmitting). So an efficient WMSN should minimize communication between nodes. In our context, we oversee to use WMSN to make a 3D-depth representation of an object or an intrusion in a scene. Our main challenge is to gather 3D-depth representation while maintaining low power consumption, high speed, and QoS in WMSNs.

Another added values of our proposed method is the processing of data locally based on various nodes, hence calculating the disparity map between various images of the same scene and then transmitting it to the coordinator via the network. Consequently, the transmission of high resolution images is reduced between sensor nodes. For a pair of video sensors, only low resolution gray-scale disparity maps are transmitted on the network in addition to a reference image. Our aim is to have a dynamic, scalable, and efficient network for real time monitoring purposes. Because real-time 3D scene reconstruction utilizing centralized procedures is impossible for very large systems, but the use of distributed computation can scatter the computation over the different sensors in the network, and so decrease the computation cost on a single node.
Accordingly, two major constraints are addressed. First, high power consumption caused by high transmission rate. Second, sensing/processing capabilities for event detection and 3D reconstruction.

1.2 OBJECTIVES AND PROBLEM STATEMENT

WMSNs consist of devices (called nodes) for acquisition and transmission, deployed in an area of interest. These networks are applied particularly to prevent or detect natural disasters, monitor fires, retrieve vital signs from a patient, or even track military troops and make nuclear attacks assessments, etc. Subject to energetic and computational constraints, sensor networks have limited processing and communication capabilities, but different domains profit from their miniaturization of hardware. The context of application that we envisage is the 3D information(s) acquisition and the supervision by a network of multimedia sensors. We thought about the surveillance of borders, private property, the management of public events, etc. These networks must detect thefts, traffic violations, unlawful territory, car accidents, and even unexplored places. Thus produce audiovisual data relevant for real-time or retrospective use (in subsequent surveys). We will consider as known the parameters describing the sensors, carrying directional cameras and defining the captured area limited to a portion of a cone: the range of detection, field of view and orientation. On the other hand, the access to several sources of image/video data provided by cameras often allows a more accurate interpretation of events: views taken from different points can help to solve the occlusion problem. Moreover, even without occlusion, a view obtained from a single source may be insufficient to make a decision, while the combination of many sources would lead to a safer interpretation. Finally, as the camera image is a planar projection, correlating several perspectives of the element can help a better identification. Stereoscopic vision is associated with the fact that the Human Visual System (HVS) perceives a scene in 3D by simultaneously viewing a scene from slightly different positions. Therefore, providing pairs of stereo images simultaneously for each eye, makes the scene perceived in 3D by the human brain. The disparity map represents the difference between horizontal disparity term between the left image and the right one. Taking into account the nature of the multimedia content, we intend to exploit...
CHAPTER 1. INTRODUCTION

the disparity map between the different views to acquire the scenes in 3D via WMSNs. The objectives of the thesis are summarized by:

- Maximize the lifetime of the sensors, this responding to a common problem of WSNs. Hence the need to aggregate the captured data. One of the objectives would be to be limited to a reference image of the scene and to exploit the disparity map instead of sending all the images of the different sensor nodes.

- Guarantee the largest coverage (Quality of Service).

Thus we proposed to dynamically adapt the parameters of the reconstruction of the scene on sensing and aggregating level. Different aggregation solutions have been developed in the context of traditional scalar sensor networks [11, 61], but they are not adapted to WMSNs because of the nature and volume of the data exchanged.

1.3/ MAIN CONTRIBUTIONS

The primary objective of this work is to design an efficient 3D scene reconstruction monitoring system. The traditional WMSNs capture images and videos and send them via aggregator(s) to a final gateway where the server(s) interprets the multimedia data in order to take a decision or trigger an alert.

- Multimedia data are rich and have large size so they affect directly the power consumption of the overall network. On the other hand, disparity maps are low size gray-scale images, holding the depth value which is sufficient to detect a change or intrusion in the scene on the nodes level before reaching the server. So we proposed to calculate disparity maps on the nodes level and transfer them to the server. In this way, we decrease the size of the transmitted data, while holding an important parameter, sufficient for event detection (ex.: change in the depth of a water container) on the node(s) level or 3D scene reconstruction on the server level.

- To calculate these disparity maps, we have a big range of existing methods, divided into global and local methods. So we started by studying all WMSNs applications, constraints, and limitations. Then we surveyed disparity map calculation methods and algorithms, resulting a table of acceptable solutions to use in the context of WMSNs. We focused on the processing clock-rate since sensors have low processing power, compared to video graphic cards or computer CPUs. We payed attention for the image resolutions, because camera sensors have low resolution(s). And then, hardware requirements are also a constraint, methods that can be implemented on FPGAs are acceptable for our context.

- Then we experimented two local methods: SAD and SSD. The first experimentation was made on existing captured online dataset. We found that the processing time increases linearly with the images resolutions and the number of pairs. SAD outperformed SSD in terms of processing time and complexity.
• We have chosen SAD as convenient method to calculate disparity maps between the couples of nodes. We have created a virtual simulation of a real indoor and outdoor environment. Then we have deployed virtual nodes with different angles and distances to explore the use of disparity maps, and define the best sensors deployment and reach the required monitoring coverage.

• We have used Structural SIMilarity (SSIM) as a criterion to calculate the quality of the disparity maps. Existing researches used a ground truth disparity map as a best quality reference, and compared their calculated disparity maps to it. In our context, we don’t have a ground-truth so our approach was to calculate SSIM between a captured image and a calculated disparity map monitoring the same region. So a high SSIM reveals a high respect of the structure, so it is a good disparity map that can be transferred to the server.

• Our simulations showed that SSIM increases with the distance between the left and right camera sensor (baseline). Wide baselines are recommended for indoor scenes to ensure a small depth error. For outdoor scenes, it is better to monitor long ranged targets and increase the angle between the sensors.

• After using and exploring SAD and the deployment criteria for our sensors, we started defining the network model for our 3D scene reconstruction WMSNs based system. Mean Squared Error (MSE) calculates the similarity between the disparities and helped us to remove unnecessary maps, thus decreasing the transmission rate on the network.

• We have designed different models and scenarios and focused on tree-based network model, which is very efficient to monitor a high range of field of views and also decreases the transmission rate.

• Principal Component Analysis (PCA) is used to cluster the received disparity maps from the nodes to the server. PCA regrouped the disparity maps, in this way, the server will make a 3D scene reconstruction using a couple of disparity maps, but not all of them. This depends on the required region to monitor. For example, a scene with 3 objects, a car, tree and house. In case we need to reconstruct the car, PCA regroups the car disparity maps, then the server will reconstruct it.

• Finally, we applied a 3D scene reconstruction algorithm on the calculated disparity maps and extracted the 3D information for the monitored scene. The 3D information is clear and it is an added value compared to existing systems, that only uses flat colored images.

1.4/ THESIS OUTLINE

The overall organization of the thesis is presented in Figure 1.3

• The first part, which is divided in two chapters. Chapter 2 studies the state of the art in the domain of WMSNs. Their applications, network technologies, structure, different architectures, and challenges. WMSNs have different limitations like energy
consumption, high bandwidth demand, architectures and protocols flexibility, scalability, localized processing and data fusion, reliability, and Quality of Service (QoS).

Chapter 3 deals with understanding different stereo vision algorithms, mainly global and local methods. This helped us to regroup disparity map calculation methods applicable in the context of WMSNs. We surveyed them and chosen a couple of stereo vision algorithms, specifically the so-called Sum of Absolute Differences (SAD) and Sum of Squared Differences (SSD).

• The second part of this manuscript is devoted to our personal contributions by applying disparity map calculation methods in WMSNs for different surveillance scenarios. We experimented SAD and SSD on simulated virtual scenes in order to chose the best stereo vision algorithm for efficient WMSNs monitoring system, based on 3D scene reconstruction. Existing research in stereo vision field uses advanced and expensive technologies. We compared our computed disparity maps to existing ground-truth maps calculated using laser scanners or depth sensors. We used Structural Similarity Index (SSIM) and Peak Signal-to-Noise Ratio (PSNR) as qualitative measurements for low cost disparity map quality. So we defined the optimal sensor nodes deployment with different ranges, field of views, nodes number, and view angles. This ensure an acceptable high quality disparity map for low cost WMSNs implementation.

• The third part defines the reconstruction model of the network in the first chapter, and shows real experimentation driven by real low cost camera sensors in the last one. This will generalize a global system model, based on dynamic parameters like angle, distance, similarity, and variance, ready to use for 3D scene reconstruction using WMSNs.
1.4. THESIS OUTLINE

Chapter 2
Understanding WMSN constraints and requirements

Chapter 3
Disparity Map principles and computation techniques
- Optimal Disparity Map calculation methods for WMSNs context (Review)

Chapter 4
3D Scene Reconstruction
- Defining the main parameters for 3D scene reconstruction using disparity maps: Baseline and Focal Length

Chapter 5
Using Disparity map techniques in WMSN
- Exploring disparity map calculation on WMSNs
- Studying complexity and processing time
- Virtual Simulations, SSIM and PCA
- Defining sensors deployment based on distances and Field of Views

Chapter 6
Disparity Based 3D Reconstruction Model
- MSE and PCA for disparity map filtering and clustering
- Decreasing Transmission Rate
- 3D Scene Reconstruction

PART I
PART II
PART III
1.5/ PUBLICATIONS


UNDERSTANDING WMSN

2.1/ INTRODUCTION

Wireless Sensor Networks (WSNs), composed of scalar sensors, shifted to Wireless Multimedia Sensor Networks (WMSNs) capable to capture scalar data, images, audio, and video. The development of inexpensive CMOS cameras and distributed signal processing and coding technologies gives the WMSNs the capability to deliver multimedia content. WMSNs nodes are battery-powered with sensing, processing, and wireless communicating capabilities [3]. The network is composed of large number of sensor nodes deployed in a scene of interest with one or many base stations as shown in Figure 2.1. The base station (sink) is the main network controller or coordinator. All gathered information from different sensor nodes are collected, stored, and processed by the sink. The sink plays also the role of a gateway to other networks. Note that sinks are mostly located in a close distance to the nodes, as long range radio communications consume high energy. Let us now investigate various technical aspects of such networks, by focusing first on their applications.

![Figure 2.1: General WMSN layout](image)

2.2/ WMSNs APPLICATIONS

WMSN are used in many applications covering various fields like real-time objects tracking in industrial production, scene monitoring in agriculture, and vital signs monitoring in health-care, etc. Examples of such applications are listed hereafter.
• **WMSNs for surveillance:** Which develop old surveillance systems by using a large deployment of low-cost and low-powered audiovisual nodes. It helps to monitor, detect, and record activities like thefts, bridges structure changes, automobile accidents, traffic violations, or any cautious behavior [10]. Adding to this list, the audio data can be used to sense unusual sounds such as shooting, explosions, etc.

• **WMSNs for traffic avoidance, enforcement, and control systems:** Numerous countries suffer from traffic, specially on the highway. So using WMSNs, drivers can view all road status and decide where to go or park their vehicles. To reach this goal, GPS data can also be added to multimedia data.

• **WMSNs for advanced healthcare:** WMSNs can provide universal healthcare services by monitoring patient parameters like pulse, blood pressure, and so on [58]. Data can be recorded and transmitted to the monitors on real-time. Multimedia data is much richer than scalar one for diagnostics and physiological data.

• **WMSNs for environmental and house monitoring:** WMSNs are very convenient for habitat and house monitoring because of their full coverage possibility of implementation. For example, they can monitor animals in a zoo or island in order to study the percentage of leaving creatures [9]. At home, thefts or any intrusions can be detected easily with all details. The system can communicate with an emergency service for intervention in case of flood, fire, or chemical detection for example.

• **WMSNs for industrial or infrastructure control:** Deploying a network of video sensors in a scene can give us the opportunity to monitor visible and non-visible objects from different views, for quality control or plant monitoring [8]. Researchers visually monitored construction and operation of buildings, bridges, or any type of civil infrastructure. Note that augmented reality can cooperate with WMSNs to develop amazing efficient practical applications like visualizing 3D representations of environmental information in real time using smart phones [25].

• **WMSNs for military applications:** A bunch of military applications, like in [41], use WMSNs for monitoring friendly forces equipment and ammunition, battlefield surveillance, investigation of opposing forces and terrain, targeting, and nuclear, biological, and chemical (NBC) attack detection and reconnaissance, and finally battle damage assessment.

### 2.3/ WMSN Components

WMSNs consist of four main components: Wireless Multimedia Node (WMN), Wireless Cluster Head (WCH), Wireless Network Node (WNN), and Base Station (BS). WMN and WCH focus on data processing, and WNN on wireless network communications.

• **Wireless multimedia node** [4]. Each node is composed of a camera or audio sensor, a processing unit, a communication unit, and finally a power unit. The camera sensor has a Field Of View (FOV) of the scene. Visual processing is achieved by the processing unit. Note that WMSNs can be implemented in a way where events
are detected before transmitting images (frames) through the network. If no useful event is detected, images are discarded.

- **Wireless cluster head.** Several WMNs send data to the cluster head. Each WCH is composed by a processing unit, a communication unit, and a power unit. Sometimes WMNs FOV overlaps, so the WCH will perform an exaggeration to remove overlapping frames.

- **Wireless network node.** The WNN is similar to traditional wireless sensor network and lies as a communication unit and a power one. The communication unit sends the data between nodes as long as it arrives at the base station.

- **Base Station,** which gathers all the data and has powerful processing and energy capabilities. It is a kind of server.

---

2.4/ **WMSN NETWORK ARCHITECTURE**

Most of scalar WSNs have a flat architecture of distributed homogeneous nodes. These networks measure physical phenomenon like temperature and provide as an output a scalar value for each measurement. WMSNs have been introduced in emerging application fields that raise the need to have alternative network architectures. Scalability, efficiency, and QoS are the main requirements for new network architectures. Generally speaking, WSNs architecture can be either single-tier flat, single-tier or single-tier clustered, or multi-tier one, as shown in **Figure 2.2**
CHAPTER 2. UNDERSTANDING WMSN

(a) Single-Tier Flat Architecture
(b) Single-Tier Clustered Architecture
(c) Multi-Tier Architecture

Figure 2.2: WMSN architectures
• **Single-tier flat architecture:** The network is set up with homogeneous sensor nodes with the same capabilities and functionality. All the nodes perform functions like image capturing, multimedia processing, or data delivery to the sink. This flat architecture is easy to manage and the multimedia processing is distributed among the nodes, contributing to the extension of the network life time.

• **Single-tier clustered architecture:** The network is deployed with heterogeneous sensors like video, audio, and scalar ones. All different sensors within each cluster relay data to a Cluster Head (CH), which is connected with the sink directly or through other CHs.

• **Multi-tier architecture:** This network is deployed using heterogeneous sensors in a multi-tier architecture.
  
  – The first tier performs simple tasks via scalar sensors like motion detection.
  – The second tier performs more complex tasks like object detection or recognition.
  – The third tier may achieve an object tracking via powerful high resolution camera sensors, which is a complex task to do.

Each tier can have a central hub to enhance data processing and communication with the higher tier. This architecture meets different needs while balancing between costs, coverage functionality, and reliability requirements.

### 2.5. WMSN CHALLENGES

In this section, we discuss important limitations and challenges for WMSN applications such as energy consumption, high bandwidth demand, architectures and protocols flexibility. And their scalability to support heterogeneous applications, localized processing, data fusion, and finally reliability and QoS.

#### 2.5.1. ENERGY CONSUMPTION

The essential limiting factor in WMSN is energy consumption, because sensor nodes are driven by small batteries and have very low energy resources. So energy optimization is a dominant consideration no matter what is the application or objective. Transmission consumes the majority of the energy. So it is very important to decrease transmission distances between nodes and the amount of transmitted data. Moreover, energy consumption increases by node electronics for sensing and processing [39]. As an illustration, Zhao et al. estimated in [103] that 64kB of data consumed $377\mu J$ of power for radio (data transmission) and $0.00195\mu J$ for program execution (data processing).
CHAPTER 2. UNDERSTANDING WMSN

2.5.2/ HIGH BANDWIDTH DEMAND

WMSNs differ from the traditional WSNs in delivering multimedia content such as images, audio, and video streams. Hence, it requires a significant bandwidth. For example, the maximum transmission rate of IEEE 802.15.4 components such as Crossbow’s TelosB or MICAz motes is 250 kbps [63]. With the same power consumption, WMSNs require higher data rates when it comes to high quality images and video with high pixel resolution and high number of frames per second.

2.5.3/ ARCHITECTURE SCALABILITY AND PROTOCOL FLEXIBILITY

The WMSNs design should respect scalability and flexibility to give the opportunity to extend the network in the future. Multimedia processing algorithms and protocols, for their part, should be flexible to support a wide diversity of applications while respecting the energy, QoS, delay, and security (privacy) limitations of WMSNs. Note that mobility can avoid the effect of environmental risks like rain, snow, high humidity, etc.

2.5.4/ LOCALIZED PROCESSING AND DATA FUSION

Transmitting multimedia content requires huge bandwidth and hence transmitting unprocessed content will need a high cost. An advantage is to benefit from multimedia processing and data fusion algorithms in order to reduce the high bandwidth requirement and also decrease the transmission cost.

2.5.5/ RELIABILITY

Another great challenge in WMSNs is the reliability of the node and data transfer. The power decrease is the main cause to drop down the node. Harsh environment or physical damage can also affect the node. Sensor nodes may work in high traffic environment, indoor, outdoor, underwater, little house, or huge building. Also, big packet retransmission causes power decrease. So the reliability of the WMSNs design to resist against such undesirable accidents is a must, and can be done by creating robust physical structure for the nodes and reliable power efficient transport functions.

2.5.6/ QUALITY OF SERVICE

The requirements of WMSNs are different from traditional WSNs using scalar sensors. Multimedia data covers images, audio, and video streams. Camera nodes can capture
images in a short time but video streams take more time and require continuous capturing and delivery. This differs from an application to another, where in some surveillance applications the real time capability is mandatory. Consequently, efficient hardware and software techniques are required to maintain the recommended QoS by a specific application.

2.5.7/ WMSNs CHALLENGES FOR DISPARITY MAP COMPUTATION AND STEREO RECONSTRUCTION

The already mentioned limitations emphasize the challenge to increase the network lifetime by decreasing the power consumption. The latter is one of the main problems in designing future in-network processing techniques. To achieve a low cost transmission, we use gray scale disparity map instead of colored images. Spatial correlation between different sensor nodes also plays an essential role: for each pair of sensors, only a difference in the frames will trigger the recapturing of two images, therefore leading to recompute the disparity map.

Otherwise, only one node will be active and the second one will enter the sleep mode. In this case, no need to send, on real time, new images to the sink. Only one image with the disparity map we calculated before are sufficient to monitor the scene. High bandwidth demand is resolved by transmitting black and white disparity maps without colored images. The fact of activating one node from the pair and sleeping the other one will decrease processing and transmission on distributed pairs level. In the next chapter (resp. Chapter 3), we will survey all disparity map calculation methods and choose the group of methods that can be handled by WMSN nodes in terms of computational demands.

2.6/ PROBLEMATIC AND SOLUTION

The main problematic can be summarized as follows: how to create an efficient monitoring based on WMSNs that lasts for a long period of time, works in real-time and with a good QoS?

The transmitted data may be used later for 3D scene reconstruction or event detection. The centralized network topology is unpractical for 3D real-time applications, so we will use a distributed wireless multimedia sensor network. By this way, the calculation and processing is divided on all sensor nodes to avoid high network load and decrease energy consumption. Each couple of sensors will make a stereo matching between a pair of images in order to calculate a disparity map. Disparity maps have low size and are good enough to 3D scene reconstruction or event (difference in depth) detection, as they contain the depth values of the environment. Sending a disparity map over the network instead of sending pair of images can decrease the size of transmitted data and hence reduce the bandwidth usage.
2.7/ Conclusion

Using WMSNs for different applications and scenarios, specially for surveillance, is gainful in terms of flexibility of deployment and low cost of implementation. But we should respect the pre-mentioned limitations and constraints. The scalability and mobility of these types of networks is promising to apply stereo vision on a distributed couples of sensors. But what is the complexity of a computer vision system? How it is applicable on WMSNs? What is the best disparity map calculation method or group of algorithms that performs better, and in an efficient way on sensors? The different types of network architectures open multiple questions and problematic on how these disparity maps will travel the network from node to node, or nodes to aggregator(s). The next Chapter 3 outlines all disparity map calculation methods in details.
3.1/ INTRODUCTION

Disparity map represents corresponding pixels that are horizontally shifted between the left image and the right camera. New methods and techniques for solving this problem are developed each year and exhibit a trend toward improvement in accuracy and time consumption [68].

Our main focus is to choose the best solution to be applied on WMSN in an efficient way when considering energy consumption, storage capability, communication cost, and quality of service. Indeed, in real-time applications, a fast and accurate depth calculation method is a must. Various methods have been classified into local methods and global ones.

All disparity map calculation steps are discussed in the next section.

3.2/ STEREO VISION DISPARITY MAP ALGORITHM STAGES

The 4 main steps for stereo vision algorithm are represented in Figure 3.1.

Figure 3.1: Disparity map computation process
3.2.1/ Matching Cost Computation

In this stage, given the values of two pixels, we determine if they accord to the same point in the scene or not. The matching cost is computed at each pixel for all considered pixels. The disparity is the difference in pixel intensity between a pair of the matching pixels in two images and can be related with depth values via 3D projection [68]. Pixel-based matching costs can be done with absolute differences, sampling-insensitive absolute differences, squared differences, or truncated versions. These algorithms can be used for gray-scale or colored images [33]. H. Ibrahim et al. introduced the fact that area based (window based) techniques can offer better data than individual pixels based ones [68]. These techniques are more accurate, because in the matching process, the entire set of pixels associated with image regions is used. The matching cost is calculated over an aggregating window that can have the form of a square or rectangle and may have a fixed or adaptive size.

Window-based technique has the following limitation: it considers that all the pixels in a support window have the same disparity values. This is not just the case near depth discontinuities or edges. Therefore, any inappropriate selection of the size or shape of the matching window can cause a low depth estimation.

3.2.1.1/ Absolute Differences (AD)

The AD algorithm aggregates the differences in intensity (luminance) between the pixel in the left image and the corresponding pixels in the right one, as following:

\[
AD(x, y, d) = |I_l(x, y) - I_r(x - d, y)|
\]  

(3.1)

- (x,y,d) represents the disparity map coordinates;
- (x,y) are the coordinates of the pixel of interest;
- d is the disparity (depth) value;
- \(I_l\) used as left reference image;
- \(I_r\) used as right target candidate image.

AD algorithm is the simplest matching cost algorithm and can be used, as in [92], for real-time stereo matching using a Graphical Processing Unit (GPU). It is functional for little textured regions but it may produce a non-smooth disparity map for a highly textured image. An updated version of AD, called Truncated Absolute Difference (TAD), was developed and implemented by Min et al. [56] and Pham et al. [65]. It uses colors and gradients and improved the results on variations in illumination.
3.2. STEREO VISION DISPARITY MAP ALGORITHM STAGES

3.2.1.2/ SQUARED DIFFERENCES (SD)

The SD algorithm aggregates the squared differences between the pixel in the left image and the candidate in the right image as following:

\[ SD(x, y, d) = |I_l(x, y) - I_r(x - d, y)|^2. \]  \((3.2)\)

Before improving the flattening of edges near depth discontinuities with a bilateral filter by Yang et al. in [96], the initial generated disparity maps contained noise. Miron et al. [57] compared SD algorithm with many other matching cost functions for intelligent vehicle applications and produced largest error because SD are reactive to brightness and noise, especially for real-time environment.

3.2.1.3/ FEATURE BASED TECHNIQUES

Huang and Wang divided the feature based techniques into [37]:

1. **Visual features of images**: Clearly detected characteristics, such as contour, shape, brightness, contrast, color, and texture.

2. **Statistical characteristics of images**: The statistical distribution of image pixels, such as amplitude mean, maximum, minimum, median, histogram, variance, and entropy.

3. **Transformation features of images**: Images using a variety of different mathematical transformations to the target image are mapped to transform domain, and then features are extracted. Commonly used methods include: Fourier transform, Hough transform, KL transform, Wavelet transform, and Gabor one. In the process of image recognition, statistical features and transform features are generally used to describe and identify an object.

Scale Invariant Feature Transform (SIFT) extracts typical invariable features from images, used for stereo matching (stereo vision). These features do not change with image scale or rotation, and provide strong matching across a considerable range of affine distortion, modification in 3D viewpoint, noise, and change in illumination [46]. The following defines the crucial steps for computation used for image features generation.

1. **Scale-space Extrema Detection.** It uses a Difference-of-Gaussian function to look for minimum and maximum interest regions that are invariant to scale and orientation.
2. **Keypoint Localization.** It estimates the location and scale of each keypoint, and then eliminates the unstable keypoints.

3. **Orientation Assignment.** It assigns one or more orientation(s) to each keypoint location built on local image gradient directions.

4. **Keypoint Descriptor.** It measures the local image gradients at the selected scale in the region around each keypoint. They are transformed into a representation that allows for significant levels of local shape distortion and change in illumination.

Sharma et al. [77] developed an updated disparity map algorithm using features from the SIFT for autonomous vehicle navigation by modifying the SIFT algorithm. They reduced computation time by improving the feature matching process using a Self-Organizing Map (SOM). Compared with the original SIFT algorithm, the computational time and cost are significantly reduced but the complete disparity map cannot be acquired.

Liang Wang concluded in his thesis [48] that feature-based approaches try to calculate correspondences for distinct feature points that can be matched unambiguously. Matching works fine on noticeable features. But it produces sparse disparities, because it uses only the matching points derived from the targeted object features. Feature-based techniques present low accuracy and are insensitive to occlusion and texture less areas. Area-based approaches take into consideration larger image regions containing richer information than individual pixels, to yield more stable matches.

Liu et al. [44] combined image segmentation and edge detection for the matching cost computation in order to reduce time and cost. Because traditional global stereo matching methods used an optimized energy function to obtain accurate disparity, but it needed high computation time and more memory, which confront real time processing. They tested it on indoor and outdoor scene (such as “Avatar”) and concluded that it does not preserve boundaries (discontinuities), but it is ideal for most of the scene cases with fast time.

Note that feature based techniques are not preferred, and not very used by researchers in the domain of disparity map calculation.

### 3.2.1.4/ **Sum of Absolute Differences (SAD)**

The SAD algorithm calculates the absolute difference between the intensity of each pixel in the reference block and that of the corresponding pixel in the target block:

\[
SAD(x, y, d) = \sum_{(x,y) \in W} |I_l(x, y) - I_r(x - d, y)| \tag{3.3}
\]

It makes the sum of differences over \(W\), where \(W\) is the aggregated support window.
3.2. STEREO VISION DISPARITY MAP ALGORITHM STAGES

The SAD algorithm is a popular algorithm for matching cost computation and it can work in real time with low computational complexity. For example, Tippetts et al. \[88\] evaluated the SAD performances on real time human poses and a resource limited system. Gupta and Cho \[26\] used SAD with two sizes of correlation windows by firstly determining the aggregation cost, then improving object boundaries (discontinuities) using smaller window size. The proposed algorithm computes more than 10 frames/s on real time, a sharp and dense disparity map for 3D scene reconstruction. Open source evaluations and comparisons have been experimented on benchmark Middlebury stereo datasets website to demonstrate qualitative and quantitative performance of the proposed algorithm and many others.

The SAD algorithm is fast but it produces low quality initial disparity maps caused by noise at object boundaries and in texture less areas.

\subsection{3.2.1.5/ \textsc{Sum of Squared Differences} (SSD)}

Equation 3.4 summarizes the SSD algorithm:

\[
SSD(x, y, d) = \sum_{(x,y)\in w} |I_l(x, y) - I_r(x - d, y)|^2. \tag{3.4}
\]

Fusiello et al. \[23\] tested SSD algorithm on multiple fixed window blocks searching for the smallest error, to select an appropriate pixel of interest in the disparity map. They introduced a novel probabilistic stereo method based on Symmetric Multi Window (SMW) algorithm. In this paper, they performed matching by correlation between different types of windows in the two images, and enforcing the left-right consistency constraint. This requires the uniqueness constraint where each point in image 1 can match at maximum one point in image 2. This is improved by using a probabilistic SMW using Markov Random Fields (MRFs). MRFs are largely used in image restoration, segmentation, and reconstruction. In \[23\], authors experimented the algorithm on indoor (“Head” stereo pairs from the Multiview Image Database, University of Tsukuba) and outdoor scenes (castle, trees and parking meters).

Many authors achieved better results in terms of speed than Fusiello’s work, but by implementing it on a platform with a GPU. That makes it impossible for WMSN usage. There is little research papers on the use of SSD algorithm for stereo vision disparity map calculation.

\subsection{3.2.1.6/ \textsc{Normalized Cross Correlation} (NCC)}

Another method to determine the correspondence between two windows around a pixel of interest is NCC. The following Equation 3.5 is the formula for NCC:
$$NCC(x, y, d) = \frac{\sum_{(x,y) \in w} I(x, y).I(x - d, y)}{\sqrt{\sum_{(x,y) \in w} I^2(x, y). \sum_{(x,y) \in w} I^2(x - d, y)}.} \tag{3.5}$$

NCC tends to blur depth discontinuities more than many other matching costs, and can only be used with local methods due to its window-based design [32]. Satoh [72] proposed a simple low-dimensional features approximating NCC-based image matching method, because NCC is robust to intensity offsets and changes in contrast, but it is computationally expensive. SAD or SSD are less robust against geometrical distortion (occlusion, clipping, perspective projection, illumination distortion), but simpler and suitable for hardware implementation such as digital signal processor (DSP).

### 3.2.1.7 Rank Transform (RT)

The RT matching cost is calculated by the following formula using the absolute difference between the rank from the reference image and the one from the target image:

$$RT(x, y, d) = \sum_{(x,y) \in w} |\text{Rank}_{\text{ref}}(x, y) - \text{Rank}_{\text{tar}}(x - d, y)| \tag{3.6}$$

where $\text{Rank}_{\text{ref}}$ and $\text{Rank}_{\text{tar}}$ are calculated as follow:

$$\text{Rank}(x, y) = \sum_{(i,j)(x,y)} L(i, j) \tag{3.7}$$

where

$$L(i, j) = \begin{cases} 0 : l(i,j) < l(x,y) \\ 1 : \text{otherwise} \end{cases} \tag{3.8}$$

in which:

- $(i,j)$ represents the coordinates of a neighboring pixel;
- $(x,y)$ represents the coordinates of the pixel of interest.

Equation 3.8 computes the number of neighboring pixels $l(i,j)$ having larger values than that of the central pixel $l(x,y)$.

Gac et al. [24] used RT for a Pipelined, Pre-fetch and Parallelized Architecture for Positron Emission Tomography (3PA – PET) and compared time performances with a PC, workstation and GPU. They obtained a reliable initial disparity map while carefully selecting the window sizes. RT deals effectively with brightness differences and image distortions but sometimes it leads to matching ambiguity when a matching pixel may look extremely similar to a neighboring one.
Zhao et al. [82] decreased this matching ambiguity using a Bayesian stereo matching model. So it considers not only the similarity between the couple of images pixels but also the ambiguity level of these pixels in their own image independently. They tested their algorithm on both intensity and color images with brightness differences and resulted effective corresponding 2D disparity maps and 3D scene reconstruction.

3.2.1.8/ Census Transform (CT)

In CT algorithm, the results of comparisons between center pixel and its neighbor pixels within a window are translated into a bit string as follow:

\[
Census(x, y) = Bitstring_{(i,j) ∈ w}(I(i, j) ≥ I(x, y))
\] (3.9)

Hamming distances between the census bit strings on the corresponding match candidates is used to calculate the CT algorithm, as follow:

\[
CT(x, y, d) = \sum_{(x,y) ∈ w} Hamming(Census_{ref}(x, y) - Census_{tar}(x - d, y))
\] (3.10)

- \(Census_{ref}\) is the census bit string from the reference image;
- \(Census_{tar}\) is the census bit string from the target one.

Humenberger et al. [38] tackled the challenge of fast stereo matching for embedded systems, because these systems have the similar limitations of WMSN as memory, processing power, and real-time capability for robotics applications. Such limitations do not allow the use of complicated stereo matching algorithms. They handled hard areas having low texture and achieved high performance results on several, including resource-limited, systems while preserving a good quality stereo matching. They gave a detailed performance analysis of the algorithm via PC, DSP and a GPU reaching a frame rate of up to 75 FPS for 640 x 480 images and 50 disparities. They compared the algorithm with existing standards (SAD algorithm) on the Middlebury Stereo Evaluation Website and achieved a 50 % quality and top performance ranking of indoor and outdoor scenes.

CT displayed higher matching quality at object boundaries (discontinuities) than those produced by SAD algorithm. One of the disadvantages is producing incorrect matching in areas with repetitive structures. Ma et al. [49] decreased the effect of repetitive structures and proposed an updated algorithm that demonstrated greater robustness when applied to a noisy image compared with conventional CT algorithm. Errors can be reduced by combining the two algorithms SAD and CT. Because CT is sensitive to areas with repetitive local structures and AD algorithm performs bad on large texture-less regions. So a
combination of SAD and CT algorithms will lead to better performance but causes an increase in computational complexity. Many researchers combined two methods like SAD and Arm Length Differences (ALD) or CT and gradient difference approaches to reach a better matching cost quality, but however matching ambiguities still can occur in some areas because of similar or repetitive texture patterns.

3.2.2/ COST AGGREGATION

Cost aggregation is the most crucial stage for the performance of a stereo vision disparity map algorithm, in a special way for local methods. In this stage, matching uncertainties are determined. Data obtained for one pixel upon calculating the matching cost is not sufficient for an exact matching, so cost aggregation is essentially needed. In local methods, matching cost is aggregated by summing them over a support region defined by a square window centered on the current pixel of interest. Applying a simple low-pass filter in the square support window is the most genuine aggregation method. Global algorithms commonly skip the cost aggregation phase and define a global energy function including a data term and a smoothness one.

3.2.2.1/ COST AGGREGATION WITH FIXED-SIZE WINDOW (FW)

The FW method experiences an increased error rate when support window size is increased over a threshold. It requires the parameters to be set to values convenient for the particular input dataset. A rectangular support window is unsuitable for pixels near object boundaries with arbitrary shapes, and a simple discontinuity interpretation method is not strict enough to conserve edges [97]. Yan et al. [97] implemented their method on both CPU and GPU. The CPU method uses C++ and OpenCV on a Core Duo 3.16GHz CPU and 2GB 800 MHz RAM with no parallelism technique. To avoid augmenting artifacts near discontinuities, multiple methods have been developed using Shifting Window or Multiple Windows (MW), Adaptive Windows (AW), windows with Adaptive Sizes, or Adaptive Support Weights (ASW).

3.2.2.2/ COST AGGREGATION WITH MULTIPLE WINDOWS (MW)

In this method, the algorithm selects multiple windows from a number of competitors in a way where support windows produce smaller matching costs. Hirschmüller et al. [34] implemented this technique for processing the local working environment of a tele-operated robot and revealed a main weakness for the multiple windows method: the computational cost. But it can be optimized using 5 windows and it is very effective. Veksler [90] developed a fast and accurate variable window approach and mentioned that the two essentials elements to achieve accuracy are the following:
3.2. STEREO VISION DISPARITY MAP ALGORITHM STAGES

• Choosing a useful range of window sizes/shapes for evaluation;

• Developing a new window cost that is particularly suitable for comparing windows of different sizes.

Their method can compute window cost over any rectangular window in stable time, regardless of window size. It is ranked in the top 4 on the Middlebury Stereo Dataset website with ground truth with comparable efficiency (tested on Tsukuba, Sawtooth, Venus and Map scenes). Hirschmüller et al. and Veksler revealed difficulties to preserve dedicated pixel arrangements in disparity maps, specifically at object boundaries. This is caused by the shape of the support windows. This method is inexact for a small number of candidates. The Adaptive Windows (AW) technique was developed to resolve this problem.

3.2.2.3/ COST AGGREGATION WITH ADAPTIVE WINDOWS (AW)

Lu et al. [47] proposed a novel stereo correspondence algorithm producing high quality disparity estimation results even with depth discontinuities or homogeneous areas using anisotropic Local Polynomial Approximation (LPA) – Intersection of Confidence Intervals (ICI) technique. They tested and evaluated results via Middlebury Stereo Data Sets on different scenes (Tsukuba, Sawtooth, Venus, and Map) implemented on a GPU for real-time applications. Chen and Su [16] expected a shape adaptive low complexity technique to eliminate computational redundancy between stereo image pairs for pixels matching. Pixels having the same depth value were grouped to reduce computations number. As the most time consuming step of stereo matching algorithm is the cost aggregation over local image regions, Fang et al. [22] published a comparative study on FW, AW, and ASW and emphasized that the best technique for cost aggregation is ASW. All the experiments were done on a NVIDIA Quadro5000 Fermi GPU.

3.2.2.4/ COST AGGREGATION WITH ADAPTIVE SUPPORT WEIGHTS (ASW)

In this approach, if the intensity of a pixel is more similar to that of the anchor pixel and if it is located at a smaller distance from the anchor pixel, it will allocate a higher weight. Many researchers like Hosni et al. [35] and Nalpantidis and Gasteratos [59] used this method on a GPU and produced great results in terms of computational efficiency and quality of disparity maps.
3.2.3/ **Disparity Computation and Optimization**

**3.2.3.1/ Disparity Computation and Optimization in the Local Approach**

The local approach uses a local Winner Takes All (WTA) strategy while selecting the disparity for each pixel in order to compute final disparities. It is defined as follow:

\[ d_p = \arg \min_{d \in D} C'(p, d) \]  

(3.11)

- At each pixel, it chooses the disparity associated with the minimum aggregated cost \(d_p\).
- \(C'(p, d)\) is the aggregated cost retrieved after the matching cost calculation.
- \(D\) shows the set of all allowed distinct disparities.

Referencing implementations made by Cigla and Alatan [17], Zhang *et al.* [101], and Lee *et al.* [42], the disparity maps obtained at this stage contained errors like unmatched pixels or occluded regions. Because in local methods the aggregation is done through summation or averaging over support regions. So their accuracy is sensitive to noise and unclear regions. Utilizing local information from a small number of pixels surrounding the pixel of interest to make the every decision will cause this problem. This can be reduced using multiple filtering techniques in the disparity map refinement step discussed later in this article.

**3.2.3.2/ Disparity Computation and Optimization in the Global Approach**

The global approach makes certain assumptions about the depth of field of the scene, expressed in an energy minimization framework. In global method, the big effort is done during disparity computation step. The well-known assumption is that the scene is smooth except object boundaries and thus neighboring pixels should have very similar disparities, this is what we call the smoothness constraint in the stereo vision literature. The main target is to find an optimal energy disparity assignment function \(d = d(x, y)\) that minimizes:

\[ E(d) = E_{\text{data}}(d) + \beta E_{\text{smooth}}(d), \]  

(3.12)

where:

- \(E_{\text{data}}(d)\) expresses the matching costs at coordinates \((x, y)\);
- \(E_{\text{smooth}}(d)\) (smoothness energy) reassures neighboring pixels to have similar disparities based on previous stated assumptions;
3.2. STEREO VISION DISPARITY MAP ALGORITHM STAGES

- $\beta$ is a weighting factor.

Belief Propagation (BP) is a well-known global method that requires big computational resources and memory for storing image data and executing the algorithm. Another global approach is Graph Cut (GC) implemented by Wang et al. [67] to optimize the energy function. Dynamic Programming (DP) assumes an ordering constraint between neighboring pixels of the same row. Arranz et al. [7] preserved high resolution disparity maps while obtaining real-time performance and keeping accurate results in the Middlebury test data set, without the requirement of parallel computing. But it fails to perform at a real time frame rate of 30 FPS for images with high resolution and different levels of disparity.

3.2.4 DISPARITY MAP REFINEMENT

At this stage, noise is reduced and disparity maps are improved. It consists of regularization and occlusion filling (interpolation). During regularization, the overall noise is reduced by filtering inconsistent pixels and little variations among pixels on disparity map. During occlusion filling, the disparity values in areas with unclear disparities are approximated. In general, it fills occluded regions with disparities similar to those of the background or texture-less areas. Left-right consistency check is used to detect occlusions.

Yang et al. [95] proposed a good algorithm that outperformed other ones on the Middlebury data sets. Their algorithm, experimented on indoor and outdoor scenes, performs well when applied on planar surfaces, because the depth information of unstable areas is increased from the stable pixels around the neighborhood by fitting a 3D plane. The performance of Yang et al. algorithm may drop if the scene is primarily composed of smooth curved surfaces like quadratic. Heo et al. [31] proposed another algorithm for depth map and consistency check. Stereo matching and performing color consistency between stereo images are a chicken-and-egg problem, since it is not a trivial task to simultaneously achieve both goals [31]. Experimental results, on indoor and outdoor scenes, show that their method achieves both accurate depth maps and color-consistent stereo images even for stereo images with severe radiometric differences.

The main local disparity refinement techniques are Gaussian convolution, median filter, and anisotropic diffusion.

3.2.4.1 DISPARITY MAP REFINEMENT USING GAUSSIAN CONVOLUTION

A Gaussian distribution defines weights for neighboring pixels in order to estimate disparities. It reduces noise but also decreases the amount of fine details in the final disparity map. Vijayanagar et al. [91] developed a real time refinement algorithm for depth maps generated from low-cost commercial depth sensors like the Microsoft Kinect, by using the weights defined by a Gaussian Filter.
3.2.4.2/ DISPARITY MAP REFINEMENT USING MEDIAN FILTER

Media filter can remove small isolated mismatches in disparity by its edge preserving property. And it is convenient for real time implementation, because it has a low computational complexity. Michael et al. [53] used Semi-Global Matching (SGM) and median filtering refinement approach on GPU for a real-time stereo vision algorithm.

3.2.4.3/ DISPARITY MAP REFINEMENT USING ANISOTROPIC DIFFUSION

Anisotropic diffusion, as implemented by Banno and Ikeuchi [12], can achieve smoothing without crossing any edge, unlike Gaussian convolution that destroys edges and fine details. Vijayanagar et al. [91] improved this approach by providing a method called “multiresolution anisotropic diffusion”. In this method, the disparity map is down sampled using 3 different resolution factors. 35 iterations of the anisotropic diffusion process are executed at each resolution. This achieved good results with no occlusion errors and refined edges in the disparity map.

3.3/ DISPARITY MAP CALCULATION METHODS REVIEW

Table 3.1 shows various disparity map methods, discussed in different articles in the field, and respecting the scientific division of these solutions like global/local or window/feature based. In addition, the hardware platforms that have been used to implement these algorithms (PC, GPU, or micro-controllers) are specified. Table 3.2 describes the technical requirements and properties that achieved the best results for each method. This is essential for choosing the best couple of methods that meets the WMSNs constraints, limitations and challenges. Tables 3.3 and 3.4 divided them in a qualitative way resolving the practical application constraints and problems like occlusion, depth discontinuities, and uniform textures areas. The computation cost, power consumption, disparity map quality, real time processing and indoor/outdoor scenario helped us to define the best method, knowing that WMSNs have limited resources working in a hard environment and different scenarios that require a high quality of service.
### Table 3.1: Disparity map method description

<table>
<thead>
<tr>
<th>Ref</th>
<th>Year</th>
<th>DM Method</th>
<th>Local/Global</th>
<th>Hardware implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>[89]</td>
<td>2013</td>
<td>Correlation Window Method (Block Matching SAD) + Features (edges) extracted by an edge detector</td>
<td>LOCAL</td>
<td>Edge Detection Unit (EDU) and Disparity Computation Unit (DCU). Xilinx ML505 FPGA</td>
</tr>
<tr>
<td>[71]</td>
<td>2013</td>
<td>Block Matching SAD and Census Transform (CT)</td>
<td>LOCAL</td>
<td>Xilinx Virtex II-Pro FPGA</td>
</tr>
<tr>
<td>[57]</td>
<td>2013</td>
<td>CT and Cross-Comparison Census (CCC) and (DIFF-Census).</td>
<td>LOCAL or GLOBAL</td>
<td>PC Intel Core 2 Duo</td>
</tr>
<tr>
<td>[37]</td>
<td>2010</td>
<td>Region Based Matching and Feature Based Matching</td>
<td>Talks about Feature Based Techniques. Doesn’t mention Disparity Map</td>
<td>PC</td>
</tr>
<tr>
<td>[46]</td>
<td>2004</td>
<td>Distinctive Invariant Features (SIFT)</td>
<td>Talks about Feature Based Techniques. Doesn’t mention Disparity Map</td>
<td>PC</td>
</tr>
<tr>
<td>[77]</td>
<td>2011</td>
<td>Scale Invariant Feature Transform (SIFT)</td>
<td>Talks about Feature Based Techniques. Doesn’t mention Disparity Map</td>
<td>-</td>
</tr>
<tr>
<td>[48]</td>
<td>2012</td>
<td>Bilateral Filtering Technique, Vector Processing, Parallelism and Dynamic Programming</td>
<td>GLOBAL</td>
<td>CPU or GPU</td>
</tr>
<tr>
<td>[44]</td>
<td>2014</td>
<td>Image Segmentation, Edge Detection and separable successive weight summation (SWS)</td>
<td>LOCAL</td>
<td>-</td>
</tr>
</tbody>
</table>

Continued on next page
Table 3.1 – continued from previous page

<table>
<thead>
<tr>
<th>Ref</th>
<th>Year</th>
<th>DM Method</th>
<th>Local/Global</th>
<th>Hardware implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>[88]</td>
<td>2011</td>
<td>Sum-of-Absolute-Differences (SAD) Block-Matching algorithm</td>
<td>LOCAL</td>
<td>CPU</td>
</tr>
<tr>
<td>[26]</td>
<td>2013</td>
<td>Area-Based Stereo Matching Algorithm (SAD)</td>
<td>LOCAL</td>
<td>CPU</td>
</tr>
<tr>
<td>[23]</td>
<td>2001</td>
<td>Area-Based Stereo Matching Algorithm</td>
<td>LOCAL</td>
<td>SUN Sparc-Station 4</td>
</tr>
<tr>
<td>[33]</td>
<td>2009</td>
<td>NCC</td>
<td>LOCAL</td>
<td>CPU</td>
</tr>
<tr>
<td>[72]</td>
<td>AND 2011</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[92]</td>
<td>2011</td>
<td>RT</td>
<td>LOCAL</td>
<td>FPGA</td>
</tr>
<tr>
<td>[38]</td>
<td>2010</td>
<td>CT</td>
<td>LOCAL or GLOBAL</td>
<td>Embedded system or PC</td>
</tr>
<tr>
<td>[97]</td>
<td>2014</td>
<td>Truncated version of Birchfield and Tomasi’s sampling-insensitive measure (BT) and Truncated Absolute Difference. Winner-Take-All (WTA).</td>
<td>LOCAL</td>
<td>CPU or GPU</td>
</tr>
<tr>
<td>[34]</td>
<td>2002</td>
<td>1. A multiple window approach that decreases errors at object borders. 2. A correlation function error filter that invalidates uncertain matches and reduces general errors. 3. A border correction method that improves object borders in a post-processing step further.</td>
<td>LOCAL</td>
<td>Pentium II</td>
</tr>
<tr>
<td>[90]</td>
<td>2003</td>
<td>Variable Window Approach</td>
<td>LOCAL</td>
<td>Pentium III</td>
</tr>
</tbody>
</table>

Continued on next page
Table 3.1 – continued from previous page

<table>
<thead>
<tr>
<th>Ref</th>
<th>Year</th>
<th>DM Method</th>
<th>Local/Global</th>
<th>Hardware implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>2010</td>
<td>Mini-Census Adaptive Support Weight (MCADSW)</td>
<td>LOCAL</td>
<td>United Microelectronics Corporation (UMC) 90nm complementary metal-oxide-semiconductor technology.</td>
</tr>
</tbody>
</table>
Table 3.2: Images sources, Images Quality, Disparity Range, Clock Frequency and Frame Rate (FPS) for each method.

<table>
<thead>
<tr>
<th>Ref</th>
<th>Images sources</th>
<th>Image Quality</th>
<th>Disparity Range</th>
<th>Clock Frequency</th>
<th>FPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>[89]</td>
<td>Calibrated stereo cameras</td>
<td>1280x1024 pixels</td>
<td>120</td>
<td>100Mhz</td>
<td>50</td>
</tr>
<tr>
<td>[71]</td>
<td>2 CMOS cameras side-by-side 7.5 cm apart</td>
<td>640 x 480 pixels</td>
<td>135</td>
<td>100Mhz</td>
<td>25-40</td>
</tr>
<tr>
<td>[57]</td>
<td>real road images from the KITTI dataset</td>
<td>1241 x 376 pixels</td>
<td>-</td>
<td>2.4 GHz</td>
<td>-</td>
</tr>
<tr>
<td>[37]</td>
<td>Video Games</td>
<td>-</td>
<td>-</td>
<td>1.86Ghz CPU 1 G 400 Mhz memory</td>
<td>60</td>
</tr>
<tr>
<td>[46]</td>
<td>Video Camera</td>
<td>233 x 189 pixels and 640 x 315 pixels</td>
<td>-</td>
<td>2 GHz</td>
<td>-</td>
</tr>
<tr>
<td>[77]</td>
<td>Stereo Vision pointgrey Bumblebee camera</td>
<td>640 x 480 pixels, 1024 x 768 pixels</td>
<td>-</td>
<td>-</td>
<td>20</td>
</tr>
<tr>
<td>[48]</td>
<td>-</td>
<td>320 x 240 pixels, 640 x 480 pixels</td>
<td>16-32</td>
<td>2.66 GHz Video Frame Rate</td>
<td>-</td>
</tr>
<tr>
<td>[44]</td>
<td>Middlebury Data Set and Video Game</td>
<td>400 x 300 pixels</td>
<td>64</td>
<td>-</td>
<td>Video Frame Rate</td>
</tr>
</tbody>
</table>
3.3. DISPARITY MAP CALCULATION METHODS REVIEW

Table 3.2 – continued from previous page

<table>
<thead>
<tr>
<th>Ref</th>
<th>Images sources</th>
<th>Image Quality</th>
<th>Disparity Range</th>
<th>Clock Frequency</th>
<th>FPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>[88]</td>
<td>Middlebury Data Set and real human poses</td>
<td>-</td>
<td>-</td>
<td>2.8 GHZ</td>
<td>-</td>
</tr>
<tr>
<td>[26]</td>
<td>Middlebury Data Set</td>
<td>320 x 240 pixels</td>
<td>16</td>
<td>3 GHZ</td>
<td>10+</td>
</tr>
<tr>
<td>[23]</td>
<td>standard image pairs from the JISCT stereo test set</td>
<td>128 x 128 pixels</td>
<td>-</td>
<td>110 Mhz</td>
<td>-</td>
</tr>
<tr>
<td>[33] and [72]</td>
<td>- x 375 pixels</td>
<td>450</td>
<td>-</td>
<td>2.8 GHZ</td>
<td>-</td>
</tr>
<tr>
<td>[38]</td>
<td>Middlebury Data Set</td>
<td>640 x 480 pixels</td>
<td>50</td>
<td>1GHZ+</td>
<td>8-75</td>
</tr>
<tr>
<td>[97]</td>
<td>Benchmark images Synthesized and real-world Stereo Pairs (Rectified)</td>
<td>Multiple x 480 pixels</td>
<td>15-59</td>
<td>3.16 GHZ</td>
<td>-</td>
</tr>
<tr>
<td>[34]</td>
<td>Stereo image pair from the University of Tsukuba and an image of a slanted object from Szeliski and Zabih</td>
<td>320 x 240 pixels</td>
<td>32</td>
<td>450 MHZ</td>
<td>1.6</td>
</tr>
<tr>
<td>[90]</td>
<td>Middlebury Data Set</td>
<td>-</td>
<td>-</td>
<td>600 Mhz</td>
<td>-</td>
</tr>
<tr>
<td>[47]</td>
<td>Middlebury Data Set</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Continued on next page
As mentioned in Section 3.1, disparity map computation methods are classified into local and global ones. Many researchers claimed in their studies and experimentation, like in [68] and [42], that global methods bring off good results but are computationally expensive. Accordingly, they are unusable for real-time applications. Furthermore, local methods are faster and computationally inexpensive, consequently suitable for the majority of real-time applications in resource-constrained WMSNs. At the same time, local methods provide acceptable results for disparity map calculation. For the above-mentioned reasons, the stereo matching algorithms based on global approach are dropped in our context. GPU implemented solutions require excessive energy consumption [89], making them unpractical for WMSNs usage.

The choice of the simplest matching cost computation that does not depend on any prerequisite complex pre-processing is fundamental for optimizing this step for WMSNs. In general, each camera has its own parameters that describe its configuration. Unique properties of each camera like focal length, central coordinate, valid pixels, and distortion coefficients are called intrinsic camera parameters. The camera parameters that show the relations between the camera and real world like rotation matrix and translation vectors are called extrinsic camera parameters. Most of stereo vision algorithms requires an essential pre-processing stage called rectification, also known as image registration. As mentioned in [29], [65], and [23], stereo pairs should be rectified in order to get horizontal epipolar lines and corresponding pixels lying on the same line in both images. Cost aggregation is the core decisive calculation step, expressly for local methods, to produce more detailed disparity maps [65] [44].
### Table 3.3: Qualitative power of each Disparity Map method on Occlusion Handling, Depth Discontinuities and Uniform textures or Texture-less areas.

<table>
<thead>
<tr>
<th>Reference</th>
<th>Occlusion Handling</th>
<th>Depth Discontinuities</th>
<th>Uniform textures or Texture-less areas</th>
</tr>
</thead>
<tbody>
<tr>
<td>C. Ttofis et al. [89]</td>
<td>Resolved</td>
<td>Resolved</td>
<td>Resolved</td>
</tr>
<tr>
<td>P. Santos et al. [71]</td>
<td>Resolved</td>
<td>-</td>
<td>Resolved</td>
</tr>
<tr>
<td>A. Miron et al. [57]</td>
<td>Resolved</td>
<td>-</td>
<td>Not Resolved</td>
</tr>
<tr>
<td>H. Huang et al. [37]</td>
<td>-</td>
<td>-</td>
<td>Not Resolved</td>
</tr>
<tr>
<td>D. Lowe [48]</td>
<td>Resolved</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>K. Sharma et al. [77]</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>L. Wang [48]</td>
<td>Resolved</td>
<td>Resolved</td>
<td>Resolved</td>
</tr>
<tr>
<td>J. Liu et al. [44]</td>
<td>-</td>
<td>Resolved</td>
<td>Resolved</td>
</tr>
<tr>
<td>B. Tippetts et al. [88]</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>R. Gupta [26]</td>
<td>-</td>
<td>Resolved</td>
<td>Resolved</td>
</tr>
<tr>
<td>A. Fusiello et al. [23]</td>
<td>Resolved</td>
<td>Not Resolved</td>
<td>-</td>
</tr>
<tr>
<td>H. Hirschmüller et al. [33] and S. Satoh et al. [72]</td>
<td>-</td>
<td>Not Resolved</td>
<td>-</td>
</tr>
<tr>
<td>G. Zhao et al. [82]</td>
<td>-</td>
<td>-</td>
<td>Not Resolved</td>
</tr>
<tr>
<td>M. Humenberger et al. [38]</td>
<td>Can be filtered with a Left/Right Consistency Check</td>
<td>Resolved</td>
<td>Not Resolved</td>
</tr>
<tr>
<td>Q. Yang et al. [97]</td>
<td>Resolved</td>
<td>Resolved</td>
<td>Resolved via Post-Processing</td>
</tr>
<tr>
<td>H. Hirschmüller et al. [34]</td>
<td>Resolved with Left/Right Consistency Check</td>
<td>Resolved</td>
<td>-</td>
</tr>
<tr>
<td>O. Veksler [90]</td>
<td>-</td>
<td>Resolved because Window Shape is variable</td>
<td>-</td>
</tr>
<tr>
<td>J. Lu et al. [47]</td>
<td>-</td>
<td>Resolved</td>
<td>Resolved</td>
</tr>
<tr>
<td>N. Chang et al. [15]</td>
<td>Decided to use the overall error rate instead of the non occlusion error rate</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Table 3.4: Computation Cost, Power Consumption, Disparity Map Quality, Real Time and Indoor/Outdoor application for each Disparity Map method

<table>
<thead>
<tr>
<th>Reference</th>
<th>Computation Cost</th>
<th>Power Consumption</th>
<th>Disparity Map Quality</th>
<th>Real Time</th>
<th>Indoor/Outdoor</th>
</tr>
</thead>
<tbody>
<tr>
<td>C. Ttofis et al. [89]</td>
<td>Medium</td>
<td>Medium (3W)</td>
<td>Good</td>
<td>Yes</td>
<td>Both</td>
</tr>
<tr>
<td>P. Santos et al. [71]</td>
<td>Medium</td>
<td>Medium (1.04 W)</td>
<td>Good</td>
<td>Yes</td>
<td>Indoor</td>
</tr>
<tr>
<td>A. Miron et al. [57]</td>
<td>Medium</td>
<td>Medium</td>
<td>Good</td>
<td>Yes</td>
<td>Outdoor</td>
</tr>
<tr>
<td>H. Huang et al. [37]</td>
<td>Medium</td>
<td>Medium</td>
<td>-</td>
<td>Yes</td>
<td>For video games</td>
</tr>
<tr>
<td>D. Lowe [46]</td>
<td>Medium</td>
<td>Medium</td>
<td>-</td>
<td>Yes</td>
<td>Both</td>
</tr>
<tr>
<td>K. Sharma et al. [77]</td>
<td>Low</td>
<td>Low</td>
<td>-</td>
<td>-</td>
<td>Indoor</td>
</tr>
<tr>
<td>L. Wang [48]</td>
<td>High</td>
<td>High</td>
<td>High Quality</td>
<td>Yes</td>
<td>Both</td>
</tr>
<tr>
<td>J. Liu et al. [44]</td>
<td>Medium</td>
<td>Medium</td>
<td>Good</td>
<td>Yes</td>
<td>Both</td>
</tr>
<tr>
<td>B. Tippetts et al. [88]</td>
<td>Medium</td>
<td>Medium</td>
<td>Good</td>
<td>Yes</td>
<td>Indoor</td>
</tr>
<tr>
<td>R. Gupta [26]</td>
<td>Low</td>
<td>Low</td>
<td>Good</td>
<td>Yes</td>
<td>Both</td>
</tr>
<tr>
<td>A. Fusiello et al. [23]</td>
<td>Low</td>
<td>Low</td>
<td>Good</td>
<td>-</td>
<td>Both</td>
</tr>
<tr>
<td>H. Hirschmuller et al. [33] and S. Satoh et al. [72]</td>
<td>High</td>
<td>High</td>
<td>Medium</td>
<td>No</td>
<td>Both</td>
</tr>
<tr>
<td>G. Zhao et al. [82]</td>
<td>Low</td>
<td>Low</td>
<td>Medium</td>
<td>No</td>
<td>Both</td>
</tr>
<tr>
<td>M. Humenberger et al. [38]</td>
<td>Low to Medium</td>
<td>Low to Medium</td>
<td>Good</td>
<td>Yes</td>
<td>Indoor</td>
</tr>
<tr>
<td>Q. Yang et al. [97]</td>
<td>High</td>
<td>High</td>
<td>High Quality</td>
<td>No</td>
<td>Indoor and Synthesized Stereo pairs</td>
</tr>
<tr>
<td>H. Hirschmüller et al. [34]</td>
<td>Low</td>
<td>Low</td>
<td>Good</td>
<td>Yes</td>
<td>Indoor</td>
</tr>
<tr>
<td>O. Veksler [90]</td>
<td>Low</td>
<td>Low</td>
<td>Medium</td>
<td>Yes</td>
<td>Both</td>
</tr>
<tr>
<td>J. Lu et al. [47]</td>
<td>High</td>
<td>High</td>
<td>High Quality</td>
<td>Yes</td>
<td>Indoor</td>
</tr>
<tr>
<td>N. Chang et al. [15]</td>
<td>Low</td>
<td>Low</td>
<td>Good</td>
<td>Yes</td>
<td>Indoor</td>
</tr>
</tbody>
</table>
3.4. CONCLUSION

Chapter 3 explained all disparity map computation phases and techniques. We can conclude from this overview that these methods are implemented on different hardwares (PC, GPU or Microntroller) with specific specs like image(s) resolutions(s), disparity range, clock frequency, FPS. Indoor and outdoor scenes have their own properties and constraints like occlusions, discontinued or uniform textures and light conditions. So our aim is to implement disparity maps calculation over WMSNs in order to get the depth value, which is essential for event detection or 3D scene reconstruction. Chapter 5 filters stereo vision algorithms so we can choose a group of methods and an efficient method to use disparity maps in the context of WMSNs surveillance, while respecting the network constraints and limitations that we already discussed in Chapter 2. We seek a low power consumption and computation cost for an efficient WMSNs disparity map based surveillance system.
4

3D SCENE RECONSTRUCTION

4.1/ INTRODUCTION

3D Scene Reconstruction is the modeling of a three-dimensional shape for a scene or an object of interest using a set of images. The added value for the result of this process is getting the depth because projecting from a 3D scene into a 2D plane, while capturing an image with a camera, will lose the depth. It is applied in different fields, like in [5] for robot navigation, in [62] for railways systems safety, in [100] for human body reconstruction, etc. So we can profit from this technique for multiple purposes in movies, automotive industry, surgery, augmented reality and so on. This chapter explains the concept of 3D Scene Reconstruction, while focusing on how we will contribute in the last chapter and reconstruct a scene using disparity maps calculated via WMSNs. Because most of existing applications use this technique traditionally with lasers and light scanners [99]. But this process is expensive, so our aim is to achieve a 3D Scene Reconstruction of a scene or object of interest using cheap CMOS camera based WMSNs.

4.2/ MULTIPLE VIEW 3D SCENE RECONSTRUCTION

Multi-view 3D reconstruction deduces the geometrical structure of a scene, captured by different cameras from multiple angles. We assume that we know the camera position and internal parameters. Recovering 3D information is achievable by solving the correspondence problem pre-explained in Chapter 3. Figure 4.1 shows an example for reconstructing a masterpiece from different angles.

Standard 3D scene reconstruction approach moves a camera around an object and takes multiple shots. In Chapter 3 we defined stereo vision as the way to calculate a disparity map from two camera nodes. These cameras acquire the images for the same scene from different views at the same time. If we know the distance between the cameras and the position of the same points in the couple of images, captured from left and right cameras, triangulation method can determine the location of 3D points [79]. Triangulation can reconstruct the 3D position (X,Y,Z) of a point of interest P by perspective projection
CHAPTER 4. 3D SCENE RECONSTRUCTION

Figure 4.1: Multiple View 3D Scene Reconstruction example

of P on image planes of the camera sensors. Considering that we know the two cameras position and orientation. Figure 4.2 shows the Triangulation parameters for a couple of cameras. L is the left camera and R the right one. f is the focal length. b is the baseline connecting the two lens centers.

Figure 4.2: Triangulation
4.3. RELATED WORK

So stereo triangulation respects the following equations:

\[ Z = \frac{b \ast f}{x_1 - x_2} \]  \hspace{1cm} (4.1)

\[ X = x_1 \ast \frac{Z}{f} \]  \hspace{1cm} (4.2)

\[ Y = y_1 \ast \frac{Z}{f} \]  \hspace{1cm} (4.3)

The depth \( Z \) is inversely proportional to the disparity \( d(x_1-x_2) \). So we can 3D reconstruct a scene by applying the triangulation on the disparity map calculated with one of the methods we surveyed in Chapter 3. It is good to note that the techniques related to 3D scene reconstruction are not in the scope of our thesis. Our purpose is to exploit the use of the disparity map at the different levels from the acquisition till the reconstruction in the WMSNs.

The next section shows a summary of different related work where researchers applied 3D scene reconstruction for different scenarios and hardware implementations.

4.3/ RELATED WORK

Shawn McCann [52] applied Structure From Motion (SFM) on a set of 16 images for a sculpture from a predefined dataset. He mentioned an example by applying 3D reconstruction on a set of unstructured collections of photographs collected from social medias. Which address multiple key challenges, specially the order of these unordered images captured from different users and cameras from all over the world.

OSM Bundler [1] is an open source code used for Open Street Map project. It is available online and uses Structure from Motion (SFM), Open Multiple View Geometry, Taung, PMVS and many algorithms are open source and available on Open Source Photogrammetry.

Michoud et al. [54] estimated the 3D shape of a moving man from silhouette images using Shape From Silhouette (SFS) technique. It relies on the intersection of the visual cones of silhouettes seen from multiple cameras. They provided a real-time system without any restriction(s) on capturing space and the placement of the cameras. They used 640x480 resolution cameras at 30 fps. Each camera is connected to a PC for silhouette extraction and sends the data to a server.

Sturm et al. [81] created a 3D miniature of a person via Kinect and 3D color printers. The character stands on a rotating chair and the result is evaluated with respect to the FPS and rotation speed. The objective of this work is to develop a low cost 3D print machine. Figure 4.3 illustrates a live experimentation.

Hsiao et al. [36] investigated and experimented a 3D scene reconstruction algorithm that helps a robot to move in an unknown environment. The system is based on stereo vision
we explained in Chapter 3, using Sum of Squared Differences SSD. They used 320x240 pixels images and announced that using larger images sizes affects computation time. Their algorithm over-performed kinect sensors with longer vision. The robot was able to navigate even in darker environment.

Anwer et al. [6] used Microsoft Kinect depth sensor for real-time underwater 3D environment reconstruction. They mentioned that even noisy data can be filtered using a Median Filter in a pre-processing phase, thus helping to generate smoother 3D mesh.

Milani and Memo [55] studied the impact of drone swarm formations in 3d scene reconstruction. Computer generated simulations ensured how occlusions between different drones affect negatively the field of view and so the reconstruction estimation. So recognizing the partner drones and synchronization between them plays an essential role for having accurate 3D scene reconstruction.

Yang and Fan [98] used 3d scene reconstruction for buildings via 3D LiDAR laser data. They adopted Velodyne VLP-16 3D LiDAR with 16 channels for different angles. Depth scene is created with Time of Flight (TOF).

Zarean and Kasaei [100] used 3d scene reconstruction for soccer scenes where human body reconstruction is the main challenge. One of the main issues is inaccurate camera calibrations and the limited number of camera sensors. They achieved good results in terms of reconstruction accuracy with a minimal number of 4 up to 9 views.

Shöps et al. [75] implemented a 3d scene reconstruction system on Google’s Project Tango Tablet. The user walked 373 meters in 12 minutes and the process was fine finished on real-time. For the first step, input videos are processed to compute depth maps via stereo matching. Subsequently, volumetric depth map fusion fused these depth maps globally. The mobile device contains a fish-eye camera capturing 320x240 images. The system resulted good models for outdoor as well as indoor large-scale scenes.

Finally, Xie and Cooperstock [94] experimented their system on two Baumer TZG01 ToF 176x144 pixels cameras and three 1296x966 pixels Basler Ace 1300-30gc color cameras. The cameras are deployed on a semi-circular configuration to increase the common Field of View (FOV) between them.
4.4. Conclusion

The impact of the third dimension is very important for a scene of interest in different domains and applications. The systems summarized in the related work show how the 3D Scene Reconstruction is applied on a big range of high cost devices like laser scanners and mobile devices. Our assumption is to reconstruct a scene using a WMSNs. The triangulation process relies on the depth map, which is inversely proportional to the disparity map respecting the formula \( 4.1 \). We are focusing on calculating disparity maps from different views via low cost WMSNs nodes, and then achieving a 3D Scene Reconstruction on the aggregator(s) level with disparity maps and few reference image(s). This process will decrease the transfer of multimedia content via the network, thus extending the network lifetime. In the next Chapter 5, we will exploit the disparity map computation on WMSNs respecting the constraints and limitations defined in Chapter 2 and Chapter 3. Our system will take into consideration the angles, distances, FOVs, camera and scene(s) parameters, and different network topology(s).
II

DISPARITY MAP COMPUTATION IN WMSN
Monitoring and surveillance attract many researchers, specially with the development of the Internet of Things (IoTs). Within this discipline, WSNs are able to sense scalar data like temperature, light, and so on [76], while WMSNs add the access to audio, images, or video data. They are built with low cost [58] Complementary Metal Oxide Semiconductor (CMOS) cameras and have a big range of applications in different fields like health, military, environmental, etc. The main challenges in WMSNs is to capture and process needed information with low energy consumption, fast computation time, and high Quality of Service (QoS).

The depth of an object or a monitored character gives us an essential clue for tracking or triggering an event. This depth will create a 3d representation of any object and it cannot be supplied by conventional gathered data. Most existing 3D reconstruction methods [69] use professional, complex, and expensive sensors with a specific network topology to obtain the required results. However, camera sensors in WMSNs have low energy resources because they are powered by limited batteries, so energy consumption is a primary constraint for WMSNs as we aim to prevent from decreasing uselessly the network lifetime. Our main objective is thus to compute 3D-depth while maintaining low power consumption, high speed, and QoS in the specific context of wireless multimedia sensor networks.

This is why we propose the use of low-complexity disparity maps computed by a couple of low cost sensor nodes in a WMSN. This is done to efficiently provide, to the monitoring system, an important depth information of the object under surveillance. Indeed, having multiple images captured at low cost from different views, and applying Stereo Vision on different couples after regrouping each pair of sensors as left camera and right one, our system can calculate at low cost some disparity maps to recover the depth information.

We will start by filtering disparity maps techniques.

Chapter 2 explained all WMSNs constraints, limitations, and challenges, while Chapter 3 listed all disparity map methods. We will now filter disparity map techniques, in order to choose a group of methods that can be used with WMSNs. We will investigate low energy consumption, low computation cost, and high QoS. By defining the best methods for disparity map computation, multimedia sensor nodes will be ready to compute these maps and then send them over the network. So, no more need to deliver a big bunch of images with high size, even the big computation is made in a distributed way, decreasing the traffic on the network and increasing network lifetime. Disparity maps will hold depth values of the scene, good enough for 3D reconstruction or event detection.

This system is summarized in Figure 5.1. As can be seen, data processing is made locally in each couple of sensors. This will reduce the transmission rate that has a direct and important effect on the network lifetime. Our system will thus not transmit high resolution images, but only gray-scale disparity maps, and on demand – depending on the triggered event or changes in the scene.

![Figure 5.1: Phases of our system](image)

We now intend to extend the aforementioned study by a deep experimentation and complexity calculation, to choose a good trade-off between speed, computation cost, and disparity map quality.

The experiment and complexity computations, provided in the next Section 5.3 ensure
5.3. EXPERIMENTATIONS RELATED TO COMPLEXITY

how Sum of Absolute Differences (SAD) is better than Sum of Squared Differences (SSD) with respects to computation time and complexity. SAD and SSD formulas are pre-explained in Chapter 3. The SAD algorithm is recalled and represented in the following algorithm 1:

Algorithm 1: Disparity map using SAD

1 Initialize block size;
2 Initialize search region;
3 function SAD(image1, image2, block size, search region)
4 Read image1, image2;
5 Compute (Height, Width) image1, image2;
6 Number of vertical blocks = Height/block size;
7 Number of horizontal blocks = Width/block size;
8 for Number of vertical blocks = 1:Max(Number of vertical blocks) do
9     for Number of horizontal blocks = 1:Max(Number of horizontal blocks) do
10        for Search region = 0:Max(search region) do
11           Compute the sum of absolute difference between selected blocks from image1, image2 in search region
12        end for
13        Save disparity of the corresponding block with minimum sum of absolute
14     end for
15 end for
16 Return disparity map;
17 end function

The algorithm shows how the disparity map is calculated for a set of pixels thus reducing the resolution of the disparity map compared to the images resolution, depending on the block size.

5.3/ EXPERIMENTATIONS RELATED TO COMPLEXITY

The Strecha et al. [80] multi-view dataset has been used for comparison, while all simulations are computed using Matlab 2016. Fig. 5.2 shows different views of the same scene. Disparity maps using SAD and SSD are depicted in Fig. 5.3. Fig. 5.4a emphasizes that SAD is better than SSD when focusing on time performance, while the computation time decreases with the captured images resolutions. Note that the performance is independent from sensor coupling. This is obvious in Fig. 5.4b, where the processing time is approximately the same for different views. Finally, Fig. 5.4c shows how processing time increases linearly with the number of chosen couples.
Figure 5.2: Different views of the scene
5.3. EXPERIMENTATIONS RELATED TO COMPLEXITY

Figure 5.3: Disparity Map using SAD and SSD

(a) Processing Time with different image resolutions
Figure 5.4: Processing time per resolution, views and number of pairs.

Table 5.1: SSIM and PSNR

<table>
<thead>
<tr>
<th>Pair</th>
<th>SSIM</th>
<th>PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>(node 1, node 2)</td>
<td>1.0</td>
<td>$\approx$ infinity</td>
</tr>
<tr>
<td>(node 1, node 3)</td>
<td>1.0</td>
<td>$\approx$ infinity</td>
</tr>
<tr>
<td>(node 1, node 4)</td>
<td>1.0</td>
<td>$\approx$ infinity</td>
</tr>
</tbody>
</table>
Table 5.1 contains Structural SIMilarity (SSIM) and Peak Signal-to-Noise Ratio (PSNR) functions applied on different pairs. SSIM and PSNR are used to measure the similarity between the two calculated disparity maps (using SAD or SSD). Getting 1.0 and infinity for different pairs ensure that the calculated disparity maps have approximately the same quality and are too similar. This will help us later to choose SAD as the best method – because it has the same quality as SSD, but with a lower complexity leading to a more efficient processing.

Indeed, the complexity of the two methods is accessible theoretically, proving that SSD is more complex than SAD. In this first situation, WMSNs will need more time to process more complex algorithms, and then it will consume more energy [50]. Computing the complexity of the two approaches using Equations 3.3 and 3.4 we have obtained the following results. For the SAD, we have 2 loops, one for horizontal width and the second one for vertical height of the image, while the operation within the loop is a single subtraction. So the SAD complexity is \(O(n^2)\) subtractions, where \(n\) is the number of lines (or columns) in the images. The SSD, for its part, integrates a square within the same loop, increasing the cost to \(O(n^2)\) complex (sqrt) operations: a square root is far more complicated to compute than a single subtraction. Such complexities are coherent with the simulations, leading to the choice of SAD as best compromise to achieve an acceptable depth evaluation in a WMSN-based video surveillance context.

Another crucial constraint in WMSNs is the coverage problem, where random objects like trees or cars can cause occlusion with the monitored target. Also, two parallel sensors can be overlapped together. Figure 5.6 shows an example of occlusion and overlapping. Costa et al. [18] surveyed this specific issue. The next sections will take into consideration all conditions to ensure a good camera sensors placement for better disparity map calculation. This plays an important role by putting sensors in sleep mode or reactivating them based on the area of interest, thus extending the network lifetime.
5.4/ Wireless Multimedia Sensor Network deployment for disparity map calculation

The Fig. 5.5 outlines our system. As can be seen, the double-view images are processed locally via the couple of low cost sensor nodes. This will reduce transmission rate, which is the direct and important cause for altering the network lifetime. Our system will not deliver high size images but only gray-scale disparity maps, on demand or triggered by a change or intrusion in the monitored scene/object. You can see how sensors are deployed (from top) and an example of a simulated virtual scene.

Figure 5.5: Proposed model
5.4. WIRELESS MULTIMEDIA SENSOR NETWORK DEPLOYMENT FOR DISPARITY MAP CALCULATION

Figure 5.6: Occlusion and overlapping problems in video sensor network coverage

Consequently, even with different constraints, WMSNs are very advantageous for surveillance purpose including:

1. Enlarging the view: two different camera nodes can monitor the same target but from different distances, one closer from the second;

2. Large field of view: different camera with multiple Field Of Views (FOVs) can be used;

3. Different views: multiple cameras, at the same distance from the target, can point to it from different angles.

Referencing to our previous contributions [84, 83], we will explain disparity map calculation constraints respecting to WMSNs prementioned limitations. This ensure an efficient usage of WMSNs for disparity map calculation.

The main idea of using disparity maps via WMSNs came from before-mentioned network limitations and the low size of these black and white maps. Calculating the depth value(s) is the essential key to trigger any alert after scene change(s). Sensor nodes can enter sleep mode, so in case an event occurs, sensors can wake up each others to recalculate
again new disparities. So worst case, one sensor will be active and the second one sleeps thus decreasing 50% of the processing and transmission on distributed pairs level. In our last article [85], we surveyed all disparity map calculation methods and regrouped them in terms of:

- Computational demand: clock frequency (in MHz/GHz), frame rate (FPS), image quality (in pixels).
- Local or Global method: global methods bring off better results than local but are computationally expensive.
- Disparity Range: some methods reach deeper disparity ranges.
- Occlusion, discontinuities and uniform textures or texture-less areas handling.
- Hardware implementation: requires a graphical processing unit GPU, central processing unit CPU or field-programmable gate array FPGA.

Our proposed compromise operates as follows: calculate efficient and accurate disparity maps on a couple of sensors, with limited computational and energy resources. We chosen to use local methods with the minimum clock rate and image quality. Thus we compared and experimented two acceptable methods: Sum of Absolute Differences (SAD) and Sum of Squared Differences (SSD) [84] [83]. Processing time increased linearly with images resolutions and number of couples (pairs of sensors).

It makes the sum of differences over $w$, where $w$ is the aggregated support window. SAD over-performed SSD with lower computational complexity and processing time, so it is convenient for real-time applications.

Studying, experimenting, and resolving computational requirements while choosing SAD as ideal disparity map calculation method for WMSNs monitoring is principal. But, as stereo matching has its own constraints, this section outlines all stereo vision requirements for better sensor deployment. At this stage, we can wonder: what are the restrictions to compute good quality disparity maps, and monitor an outdoor or indoor scene from different views and perspectives?

For all researchers working in this field, designing and developing accurate stereo vision systems, and improving the quality of 3D projections, stereo baseline is preliminary. This is the distance between sensor 1 and sensor 2. Wutthigrai Boonsuk [13] implemented a stereo vision system for industrial robotics and concluded the following: shorter baseline distances performs better at shorter distances to the target, but at longer distances, wide baselines are better. Si et al. [78] used camera sensor networks to detect intrusions passing through a barrier area. They implemented multiple parallel cameras in order to guarantee a maximum coverage for the monitored barrier. Increasing the number of cameras lets them ensure that a minimum of two sensors will capture the intruder.

Kim et al. [40] proposed a 3d modeling method using high resolution spherical images. Their 3D reconstruction was based on stereo image pairs using vertical displacement
between camera views. The epipolar geometry depends on the cameras extrinsic and intrinsic parameters. Extrinsic parameters are the camera rotation and translation in the environment. Intrinsic parameters are the internal properties of the camera itself like focal length, pixel width, and pixel height.

In our context, the added value is computing the disparity map between two camera nodes and tracking the change(s) in depth or reconstruct the intruder in the scene.

**Figure 5.7**: Stereo vision system baseline and orientation

*Figure 5.7* illustrates the effect of increasing parallel cameras baseline or rotating them. The disparity value is between 0 and 1 and it increases with the sensing direction. So a high disparity value indicates a high difference between the couple of images: they are less correlated. Otherwise, two perpendicular cameras, weakly correlated, leads to a disparity of 1 [19].
CHAPTER 5. USING DISPARITY MAP TECHNIQUES IN WMSN

Figure 5.8: Stereo Coefficient principle

Figure 5.8 shows how depth changes with baseline $b$ and distance to target $h$. $C_1$ and $C_2$ are the centers of the sensors. $M_1$ and $N_1$ are the projections of the points $M$ and $N$ in the first image. $M_2$ and $N_2$ in the second image. $\Delta M$ and $\Delta N$ are the shifts between these positions. $\Delta M - \Delta N$ is proportional to the disparity $\Delta E$ and $\Delta E$ itself is approximately proportional to the depth difference $\Delta Z$. A larger coefficient $b/h$ leads to smaller error in depth. However, a high coefficient reflects more changes between the images, hence more difficult matching process. For example, metropolitan images where there is a big number of buildings creating occlusions and changing fast with observation angle. Thus, smaller angles between views is recommended for accurate disparity maps.

So the choice of the coefficient $b/h$ depends on the monitored scene scenario compromise. Delon et al. [20] studied deeply stereo visions with small baselines and resulted that an angle of $53\%$, typically $b/h = 1$, is ideal for indoor. Objects far from sensors lenses shift less, and objects nearer shift more. In traditional WMSNs coverage, it is advantageous to increase the angles and distances between the nodes to avoid duplication of data. Two near sensors may monitor the same object. But in our context, it is different because we tend to increase intersection between cameras, since we risk to search for a match in the second sensor that is not shown in the first reference one. The next Section 5.5 shows experimental results for two different outdoor and indoor scenes, where disparity calculation is made with different baselines, distances, and angles. This ensures and improves the sensors deployment strategy for better disparity maps quality.
5.5. EXPERIMENTATIONS RELATED TO COVERAGE

We created our own data-set using Autodesk 3ds Max 2018, where we modelled two realistic outdoor and indoor scenes. The indoor scene represents an elderly man walking in the bedroom, as shown in Figure 5.9. Similarly, Figure 5.10 displays the outdoor scene of an urban city where multiple persons are promenading in the street. This represents two different WMSNs E-Health and habitat monitoring applications.

Figure 5.9: Indoor scene: elderly man walking in his private bedroom
The simulated scenes have different textures, lighting conditions, colors, reflections, and occlusions. So it keeps natural and random environmental conditions as real as possible. Cameras are placed with predefined Field Of Views (FOV) of 45 and 60 degrees. Our experimentation takes into consideration multiple baseline distances and distance to the target (the man) in meter(s), and angles of observation in degree(s). Figure 5.11 shows an example of camera sensors placement to cover the monitored scene.
Firstly, for each camera couple, we rendered the images and exported them as Joint Photographic Experts Group (JPEG). The rendered captured images are low in resolution of 640x480 pixels (indoor) and 480x270 pixels (outdoor). This is because resolution affects directly the disparity map processing time, as experimented in our last publication [83]. Secondly, Sum of Absolute Differences (SAD) uses them as parameters to compute disparity maps via MATLAB 2017. Finally, we made a quality testing on the computed disparity maps to conclude the best sensors placements (distance and orientation). To experiment the disparity maps quality, traditional computer vision researchers use a ground truth disparity map as reference. This best quality reference is computed with a laser scanner or from middlebury data-set [73]. This is not feasible in our context since we created our own virtual scenario, so no ground truth.

The Structural Similarity (SSIM) is used to solve this issue and define a disparity map quality measurement methodology. SSIM measures the similarity between two images, thus making it convenient to compare the saved structure between the computed disparity and one of the left or right images pair. To prove the concept, we applied SSIM on different disparity maps going from bad to very high quality, and we monitored how SSIM changed linearly from 0 to 1.

![Left Image](image1.png) ![Right Image](image2.png)  
![Disparity Map](image3.png)  
![SSIM value = 0.3941](image4.png)

Figure 5.12: Applying SSIM on a computed disparity map of two different views

Table 5.2, Table 5.3, and Table 5.4 show different disparity map qualities for multiple base-
Table 5.2: SSIM values for different baselines at 1.5 m from target (indoor)

<table>
<thead>
<tr>
<th>Perfectly aligned sensors</th>
<th>Baseline b (meters)</th>
<th>Distance to target h (meters)</th>
<th>SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.1</td>
<td>0.1363</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>0.1596</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.3</td>
<td>0.1947</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.4</td>
<td>0.2350</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>0.2529</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.6</td>
<td>0.2537</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.7</td>
<td>0.2395</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>0.2364</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.9</td>
<td>0.2637</td>
<td></td>
</tr>
</tbody>
</table>

Table 5.3: SSIM values for different angles at different distances from target (indoor)

<table>
<thead>
<tr>
<th>Rotated sensors</th>
<th>Angle between sensors (degree)</th>
<th>Distance to target h (meters)</th>
<th>SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.9</td>
<td>0.9</td>
<td>0.2539</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>1.5</td>
<td>0.3377</td>
</tr>
<tr>
<td></td>
<td>3.0</td>
<td>0.9</td>
<td>0.3649</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>1.5</td>
<td>0.1754</td>
</tr>
<tr>
<td></td>
<td>3.0</td>
<td>0.9</td>
<td>0.2237</td>
</tr>
<tr>
<td></td>
<td>3.0</td>
<td>3.0</td>
<td>0.2640</td>
</tr>
</tbody>
</table>

Table 5.2 shows how SSIM increases with the baseline distance. So, for indoor scenes, it is recommended to use wide baselines thus ensuring a small FOV, small depth error but more occlusions risks. For small baselines, the stereo correspondence is simpler as images are similar with few occluded regions, but it risks a large depth uncertainty.
Table 5.4: SSIM values for different angles (outdoor)

<table>
<thead>
<tr>
<th>Angle between camera1 and camera2 (degree)</th>
<th>Distance = 4 m</th>
<th>Distance = 6 m</th>
<th>Distance = 8 m</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.1597</td>
<td>0.1602</td>
<td>0.1453</td>
</tr>
<tr>
<td>20</td>
<td>0.1318</td>
<td>0.1346</td>
<td>0.1282</td>
</tr>
<tr>
<td>30</td>
<td>0.1387</td>
<td>0.1445</td>
<td>0.1367</td>
</tr>
<tr>
<td>40</td>
<td>0.1371</td>
<td>0.1356</td>
<td>0.1344</td>
</tr>
<tr>
<td>50</td>
<td>0.1375</td>
<td>0.1294</td>
<td>0.1431</td>
</tr>
<tr>
<td>60</td>
<td>0.1395</td>
<td>0.1318</td>
<td>0.1506</td>
</tr>
<tr>
<td>70</td>
<td>0.1310</td>
<td>0.1386</td>
<td>0.1502</td>
</tr>
<tr>
<td>80</td>
<td>0.1235</td>
<td>0.1354</td>
<td>0.1533</td>
</tr>
<tr>
<td>90</td>
<td>0.1293</td>
<td>0.1354</td>
<td>0.1575</td>
</tr>
</tbody>
</table>

Based on Table 5.3 results, increasing the angle between the sensors gives us more opportunity to have good disparity maps for long distance to target. For short distances, changes in the angle results a big difference in the views. Table 5.4 insured that for outdoor scenes, monitoring a long ranged target can be done with higher angle between sensors. Yet, the ideal baseline for indoor monitoring is 0.9m. The best angle for rotated sensors is 30 degree with a baseline of 3.0m. For outdoor, putting sensors with 10 degrees rotation and 6m to target calculates good disparity maps.

5.6/ CONCLUSION

Calculating disparity maps on distributed sensors is a WMSN added value for traditional multimedia data by saving and delivering depth information. This is preliminary and essential for events/changes detection or 3D scene reconstruction. Our approach respects all WMSNs limitations, specially power consumption and quality of service. This chapter outlined an experimental contribution to ensure good sensors deployment for better disparity map quality.

In the next Chapter 6 we intend to explore and define how disparity maps will be transferred from couple to another, from nodes to aggregator(s). In this way, we will reach an efficient WMSNs based surveillance system that covers a high range of targets for long lifetime. The 3D reconstructed information(s) can help different applications in many domains like elderly people monitoring, military tracking, habitat monitoring, etc.
III

3D RECONSTRUCTION MODEL USING DISPARITY MAP IN WMSN
6

DISPARITY BASED RECONSTRUCTION MODEL

6.1/ INTRODUCTION

In this chapter, we propose a 3D scene reconstruction model based on the DM in WMSN. The DM is computed between adjacent sensor nodes, in order to reduce the number of transmitted images. The computation of the DM is distributed over the network. This allows us to have 3D information about the scene on one hand, and use the DM of the different views to make a 3D reconstruction on the other hand. The DM computation for 3D reconstruction is done based on a binary tree distribution [51]. The computation at each node of the same level can be done in parallel. Data is transmitted between consecutive levels. The disparity map is computed at each node level distributively. A threshold technique based on the Mean Square Error is used at each level to discard similar DMs. At the last level, the resulting DMs are classified and regrouped using Principal Component Analysis (PCA). The classified DM’s are finally used for 3D scene reconstruction. We evaluate our approach based on the reduced amount of data compared to the classical approach.

6.2/ PROBLEMATIC

Wireless Multimedia Sensor Networks (WMSNs) are widely used nowadays for several types of applications like surveillance, traffic avoidance, environmental and event monitoring. WMSNs are reasonable for living space applications as they can be heavily deployed for complete coverage. A WMSN is composed of a number of end nodes that have limited resources in terms of energy, computation and storage capacity. Monitoring applications use traditionally 2D images and videos to identify intrusions, and physical deformation of a specific subject in a scene. This multimedia data requires significant size and induces computational and transmission cost, that consumes the limited energy resources of the nodes.
Few years ago, depth information was introduced for 3D vision applications. Depth can bring additional information. It is used for monitoring, medical surgeries, environmental and military purposes. 3D depth information can be acquired using several technologies, like laser scanner, ultra sound, etc. This requires high computational resources and high cost material and cannot be done in real time. 3D stereo images have been used for 3D vision and 3D reconstruction in several applications. Disparity is the displacement between the left and the right image. The disparity can be estimated using stereo vision. A stereo vision system is composed of two stereo cameras (left and right camera) that capture simultaneously two images from the same scene. The pair of images is processed to calculate a Disparity Map (DM) that recovers the depth information. The DM is a gray-scale image resulted from a stereo correspondence algorithm in order to represent corresponding pixels that are horizontally shifted between the left image and the right. As discussed in the previous Chapter, there are many methods and algorithms to solve this problem that differ in accuracy and time consumption. DM is used to get additional knowledge about the stereo images and its representation of the scene.

Data aggregation and data reduction are presented as a solution to reduce the increased amount of data while conserving its relevance. Various work has been recently presented on how the acquired data is exploited and combined over a sensor network. Andrea Masiero et al. worked on a tree distributed reconstruction, they proposed to distribute the computational cost on a binary tree. Zhou et al. used an approach called disparity compensation to decrease the amount of data between neighboring views using differential inter-view coding. This frame concealment technique recovers the lost frames in WMSNs. Chang et al. proposed a reconstruction algorithm for processing a set of correlated images. Zhang et al. proposed a multiple camera deployment method for visual coverage of 3D surfaces. Their approach optimizes and guarantees the optimal camera pose(s). Nandhini et al. worked on Compressed Sensing (CS) by applying a background subtraction for anomaly detection and surveillance using WMSNs. They came up with a mean measurement differentiating approach with customization threshold design. Liu et al. worked on a joint reconstruction algorithm for video reconstruction from different sensed frames. They proposed a decoder that works on inter-frame, intra-frame, and inter-view.

In this chapter, we propose a 3D scene reconstruction model based on the DM in WMSN. The DM is computed between adjacent sensor nodes, in order to reduce the number of transmitted images. The computation of the DM is distributed over the network. This allows us to have 3D information about the scene on the one hand, and use the DM of the different views to make a 3D reconstruction on the other hand. The DM computation for 3D reconstruction is done based on a binary tree distribution. The computation at each node of the same level can be done in parallel. Data is transmitted between consecutive levels. A threshold technique is used at each level to reduce similarity. The disparity map is computed at each node level distributively. The set of disparity maps at the last level is used for 3D scene reconstruction. The remainder of this chapter is organized as follow: in Section, the proposed method is presented. The reconstruction process is presented in Section. Experimentation are conducted in Section while defining different nodes distributions. Finally, conclusion and future work are drawn in Section.
6.3. **PROPOSED FRAMEWORK**

The proposed framework for 3D scene reconstruction is composed of four main steps as illustrated in Figure 6.1. The disparity information is computed distributively at the node level avoiding computational load for real time 3D scene reconstruction at the base station.

- The first step is dedicated for image acquisition and sensors coupling. If the sensors are deployed based on a well known topology, where two adjacent sensors are coupled as a node, then one sensor will share the captured image with the neighboring sensor. Otherwise, if the sensors are randomly distributed, a condition is applied between nearest sensors in order to couple each pair as a single node. Disparity maps are then calculated at each node of the first level.

- In the second step, an aggregation based on Mean Squared Error (MSE) metric is applied to discard close disparity maps. This is done by a chosen master sensor between two adjacent nodes. This step will be repeated for the aggregation between all the nodes. Finally, we will have a limited number of aggregated disparity maps.

- The third step aims to classify the aggregated disparity maps into distinct clusters. This is accomplished using the Principal Component Analysis technique [21].

- The fourth step of the framework consists of 3D reconstruction of the scene using the aggregated disparities.

6.3.1/ **DISPARITY COMPUTATION BASED ON BINARY TREE DISTRIBUTION**

Although in WMSNs, a compromise is considered between the consumed energy for transmission and the one for processing, Wu et al. [93] showed that processing an amount of data requires less energy than transmitting it. According to [70][93], the energy required to transmit 1 kbit of data for 100 m is similar to the energy taken to process 3 million of instructions in a general purpose processor. For this reason, focusing on the data reduction for transmission is of interest. Our proposed framework can be modeled as follows.

The disparity between two adjacent sensor nodes is computed and denoted $\text{DM}_{(i,h)}(j)$ presented in Equation (6.1)

$$\text{DM}_{(l,h)}(j) = (\text{Image}_i, \text{Image}_{i+1})$$  (6.1)

where $l$ denotes the level of the tree distribution, $h$ represents the number of coupled
sensor nodes, $i$ denotes the index of the coupled sensor nodes, $j$ denotes the index of the computed disparity map, and $m$ the number of sensors nodes.

For the first level $l = 1$ of the distribution tree, the number of coupled nodes will be $h \in \{1, m/2\}$, and the index of the computed disparity map is $j \in \{1, m/2\}$.

For the second level $l = 2$ of the distribution tree, the number of coupled nodes will be $h \in \{1, m/2\}$, the number of sensor nodes is $i \in \{1, m/2\}$, and $j$ the index of the computed disparity map is $j \in \{1, m/4\}$.

For the remaining levels,

$h \in \{1, m/2^{(l-1)}\}$,

$i \in \{1, m/2^{(l-1)}\}$,

$j \in \{1, m/4^{(l-1)}\}$.

The main goal of our proposal is to share the computational load over the network while considering the connected sensors as a binary tree distribution as shown in Figure 6.2.

Each pair of the sensor nodes will be considered as one powerful node. Each powerful node will assume the computation of the DM. The DM computation at the same level can be done in parallel while for the remaining levels, this can be handled in a pipeline. Similar DMs from each pair are filtered using a threshold metric.

The process of the DM computation will pass through the leaves of the tree. Then it - iteratively goes to the root. At the first level, each sensor node of every pair will send an image representing $m/2$ couples.

Our proposed method will apply a threshold technique called Mean Squared Error (MSE) respecting the following formula [6.3]. The MSE represented in Formula [6.2] is a quality metric used to compare an original image to a predicted one. If it is closer to zero, this means that the images are very similar. Otherwise, they are different.

\[ MSE = \frac{1}{N} \sum_{i=1}^{N} (Y_i - \hat{Y}_i)^2, \]  

(6.2)

where $Y$ is the vector representation of the original values of the image, and $\hat{Y}_i$ is a vector representation of the predicted image.

If the score is under a predefined average:

\[ MSE[DM_{(l,h)}(j), DM_{(l,h)}(j + 1)] < \alpha, \]  

(6.3)

then the two disparity maps will be considered as redundant, therefore one of DMs is discarded one.

This means that the number of sharing(s) will decrease at each level, while allowing the system to save energy, since DMs have reduced size compared to a regular image.
Let us consider a WMSN composed of \( m \) sensor nodes. Table 6.1 shows the number of shares between the sensor nodes at different levels.

<table>
<thead>
<tr>
<th>Reconstruction</th>
<th>level 1</th>
<th>level 2</th>
<th>level 3</th>
<th>level ( n - 1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image-based</td>
<td>( m/2 )</td>
<td>( m/4 )</td>
<td>( m/8 )</td>
<td>1</td>
</tr>
<tr>
<td>Disparity-based</td>
<td>( m/2 )</td>
<td>( m/4 \leq ) nb of shares ( \leq (m/2 + 1) )</td>
<td>( m/8 \leq ) nb of shares ( \leq (m/4 + 1) )</td>
<td>1</td>
</tr>
</tbody>
</table>

The algorithm has various steps as shown in Figure 6.3. We will discuss each part to achieve our aim. The pseudo-code of the proposed algorithm is as follows in Algorithm 2.

**Algorithm 2:** The pseudo-code of the proposed model

```plaintext
1 if sensor power > 50% and sensorType = master then
2   if level == 1 then
3     Capture image(i+1);
4     Receive image(i);
5     DM(i+1) = Compute disparity map;
6   else
7     if level > 1 and level < level.length then
8       Receive DM(j);
9       if MSE(DM(j), DM(j + 1)) \( \leq \) \( \alpha \) then
10      Send(DM(j)) to level = level + 1;
11     else
12       Send(DM(j), DM(j + 1)) to level = level + 1;
13     end
14   else
15     if level == level.length then
16       DMList = set array of disparity maps;
17       Send DMList to server;
18       PCA(DMList) (server);
19       3D reconstruction (server);
20     else
21       end;
22     end
23   end
24 else
25   Change type to type = slave;
26   Notify sensor slave to change type to master;
27   Capture image;
28   Send image to master sensor;
29 end
```
6.3.2/ DISCARDING SIMILAR DISPARITY

6.3.2.1/ MONITORING ALGORITHM

Monitoring is to check the progress or quality of something over a period of time. Monitoring intervenes in many fields: biology, medicine, computing, and environmental science. Monitoring is a major problematic and it requires a lot of equipment and a lot of money to do it consistently and efficiently.

WMSNs can be implemented in a large area for monitoring purposes. Our method can help to indicate and detect an event very fast. First, left sensor will capture image and send it to the right sensor where disparity map is calculated then compared to the old one by applying MSE. If the score becomes above 0 then an event has occurred as shown in Figure 6.4.

In some cases, nodes are too close to each other. So one node can monitor and the other one enters sleep mode and save energy. The active node can calculate DM between new and previous saved frame captured by itself. For example, one node focusing on a walking man, each time the position changes, the DM changes too and reflects the walked distance. Same for a change in the structure of a bridge, the shape of a fuel or water container, etc. In this case, the system will not face failure because we always have sensors with energy power. In other cases, if the whole network was involved in monitoring, then it may shutdown after a while.

In WMSNs, as much as the capture rate of the node becomes greater, the target could be better perceptible. It is not idealistic to consider that nodes should capture at their maximum capture rate. For example, if the system is used to monitor a bridge, then nodes must capture images every month. While if the system is implemented in a dense area where people move a lot and continuously, then the system must be performing at most.

WMSNs are also used for underground purposes. Assuring the safety of workers in mine is absolutely important. In order to locate workers underground, each sensor will detect the lamp on the worker’s head and once detected the sensor will notify the system about the worker position.

6.3.3/ DISPARITY CLASSIFICATION USING PRINCIPAL COMPONENT ANALYSIS

PCA is a technique for identification of the uncorrelated variables. It is used for extraction of relevant information from a complex dataset. It is a variable lessening strategy, which is valuable to be connected on various repetitive variables. Reducing the data set into a smaller number of principal components will maintain the maximum of variance of the observed variables. Hence, relevant information could be extracted from the complex data set. This could be done using the biggest eigenvector. It is a process of eight steps.
applied on a big data set:

1. Get image data: Suppose \( x_1, x_2, x_3, \ldots, x_m \) are represented as \( N \times 1 \) vectors.

2. Calculate the average of vector \( \bar{x} = \frac{1}{m} \sum_{i=1}^{m} (x_i) \)

3. Subtract the Mean: \( \Phi_i = x - \bar{x} \)

4. Calculate the covariance matrix (\( N \times M \) matrix) \( C = \frac{1}{m} \sum_{n=1}^{m} (\Phi_n)(\Phi_n)^T \)

5. Compute eigenvalues and eigenvectors of the covariance matrix.

6. Forming a feature vector: eigenvectors are arranged by eigenvalue, most noteworthy to least. The eigenvector with the highest eigenvalue is the principle component of the data set. Feature vector is formed by choosing the highest eigenvalue.

7. Simply take the transpose of the vector and multiply it on the left of the original data set, transposed.

8. Final data = row feature vector * row data adjust.

The benefits of Principal Component Analysis (PCA) are the following:

1. PCA helps to comprehend a large number of data set, as we people can just understand three dimensional.

2. PCA is a well-settled mathematical method for reducing the dimensionality of data, while keeping as much variation as possible.

3. PCA used for dimension reduction in a complex data set. It shows the most dominant patterns.
CHAPTER 6. DISPARITY BASED RECONSTRUCTION MODEL

Figure 6.1: General framework for 3D reconstruction based on DM

Receive DMs and classification using PCA for 3D Reconstruction

$\ell=3$

$MSE \leq \sigma$

Discard DM by one of the powerful nodes

Transmit $DM_{\ell,3}(j)$

Receive $DM_{\ell,2}(j)$ and compare it with $DM_{\ell,2}(j+1)$ using MSE

$\ell=2$

Capture Image $i$ and transmit it

Capture Image $i+1$ and send it

Capture Image $i$ and send it

Capture Image $i+1$ and send it

$\ell=1$
Figure 6.2: Binary tree applied on the system

- Leaves transmit DM and compare it
- Leaves transmit images
- Leaves receive image
  - Create DM and transmit them to the next level
- Last level where DMs are regrouped for 3D reconstruction
CHAPTER 6. DISPARITY BASED RECONSTRUCTION MODEL

Figure 6.3: Algorithm flowchart
6.3. PROPOSED FRAMEWORK

Figure 6.4: Monitoring Principle

- Capture and send it to the right
- Receive image and calculate DM with local image

{ Mean Square Error calculation applied between old DM and new DM. If MSE becomes not equal to 0, A new event is detected }
CHAPTER 6. DISPARITY BASED RECONSTRUCTION MODEL

6.4 / DISPARITY BASED RECONSTRUCTION

The depth information is lost after capturing a 3D scene into a 2D image. And vice versa, recovering the 3D information from an existing 2D image, is called 3D scene reconstruction.

We figure these depth values using a disparity map. A disparity map is a one diverted picture whose pixel values indicate the level of disparity between locations in the left and right stereo images.

So we will start by explaining the triangulation which can lead to a 3D mesh using a disparity map.

6.4.1 / TRIANGULATION

The reconstruction of a target taken from an arrangement of cameras can be figured as a geometric issue of triangulation [51]. The depth information is recovered from the disparity between a pair of images after finding the corresponding points.

In computer vision, triangulation is to retrieve the 3D coordinates of an object in a couple of 2D images. This process respects the intrinsic and extrinsic parameters of the camera sensor. The optical center and focal length of the camera are the intrinsic parameters. The extrinsic parameter is the location of the camera in the 3-D scene, represented in terms of rotations and translations [43].

We recall the basic principle for 3D reconstruction developed in Section [4.2], where at a first stage the DM is combined with the intrinsic parameters (the baseline b and the focal distance f) to compute the depth information. And at a second stage, usind the depth information with the extrinsic parameters (rotation and translation matrix) in order to generate the 3D points in the cloud. Recalled and summarized in the following:
6.4. DISPARITY BASED RECONSTRUCTION

Figure 6.5: Triangulation

So stereo triangulation respects the following equations:

\[ Z = \frac{(b \ast f)}{(x_1 - x_2)} \]  \hspace{1cm} (6.4)
\[ X = \frac{x_1 \ast Z}{f} \]  \hspace{1cm} (6.5)
\[ Y = \frac{y_1 \ast Z}{f} \]  \hspace{1cm} (6.6)

The pseudo-code of the 3D scene reconstruction is presented in the algorithm 3:

**Algorithm 3**: The pseudo-code of 3D scene reconstruction

1. Initialize the stereo parameters of the camera sensors: Extrinsic and intrinsic parameters;
2. Intrinsic parameters: focal length \( f \) and baseline \( b \);
3. Extrinsic parameters: rotation matrix \( R \) and translation matrix \( T \);
4. Read reference image;
5. Read DM respecting a defined DisparityRange and BlockSize;
6. Depth computation(b,f,DM);
7. Triangulate(R,T,depth);

Our contribution consists of bypassing the DM calculation that is traditionally done at the 3D reconstruction level on the server. Since the calculation of DMs is distributed on the sensor nodes, prior to the reconstruction phase.

In order to achieve our purpose, we used Middlebury dataset to test our reconstruction algorithm [74]. We have selected two rectified images as shown in **Figure 6.6a** and **Figure 6.6b**. The first step was to create disparity map from these two images. Next, we
used the calculated disparity map and one of the images to build the point's cloud. Finally we have a reconstructed scene created from disparity map and one reference image. The result of the first reconstruction is shown in Figure 6.7.

(a) Left image

(b) Right Image
Figure 6.7: Reconstructed scene from left and right images
6.5/ **Experimentation Results**

In order to test the proposed algorithm in our study, real outdoor and indoor images were taken from real camera fixed on a tripod. Specifications are the following: dimensions 216 x 216 pixels, color space RGB and Focal length 4.2 with the approximate size of (50 - 150) KB. Many scenes were located from daily life. Many topologies experimented: tree, ring, and mesh topology. We used a computer running on iOS, core i5 with 8 GB RAM. Concerning the code part, we used python 3.6 with OpenCV to calculate disparity maps.

6.5.1/ **Nodes Distribution Scenarios**

We propose a tree network topology to work with our algorithm to cover a large scene under surveillance.

![Diagram of our tree topology]

A tree topology is a mixture of a star network topology and a bus topology. The advantages of a tree topology are summarized as follows:

- Convenient for a communication between two or more big networks (large outdoor or indoor area);
6.5. EXPERIMENTATION RESULTS

- Scalable, where we can add as many nodes as the scene requires;
- A damage on a certain level will not affect the overall network performance;
- Easier maintenance and error detection.

The disadvantages of a tree topology are the following:

- A lot of maintenance is needed.
- Distance between nodes requires more energy to share information.

Here is comparative Table 6.2 based on network's needs to cover a large area with low power consumption and minimum number of nodes. So, if we want to monitor a bridge [2] it is better to use ring topology [28]. While if it is about covering a large scene, the best fit is a tree topology.

<table>
<thead>
<tr>
<th>Topologies</th>
<th>Number of nodes</th>
<th>Area coverage</th>
<th>Power consumption</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tree Topology</td>
<td>≥ Ring Topology</td>
<td>≥ Ring Topology</td>
<td>≥ Ring Topology</td>
</tr>
<tr>
<td>Ring Topology</td>
<td>≤ Tree Topology</td>
<td>≤ Tree Topology</td>
<td>≤ Tree Topology</td>
</tr>
</tbody>
</table>

6.5.2/ TREE TOPOLOGY

This scene is in outdoor and the main purpose is the surveillance of a small house with a large garden. The distance between the house and the first level is about 6 meters and the same distance is between every level. The distance between each point (the sensor) is 0.9 meters, respecting the studies and simulations achieved in Section 5.4 and published in our previous work [87]. The first level contains 8 nodes while the second level contains 4 nodes, the third level contains 2 nodes and the final level contains one unique sensor node.

All real images were taken at noon, the sun is directly above the scene. So there is no shadows or lighting problem(s) as shown in Figure 6.10.

So in worst case, we will have 7 disparity maps to achieve the reconstruction. At the first level, the algorithm has calculated 7 disparity maps as expected and shown in Figure 6.11.

In the PCA Figure 6.12, the label represents disparity maps. PC1 is the x-axis while the y-axis is always considered null because PC1 contains the most variances. At the
end, the system shows that it discarded three disparity maps and kept four as shown in the Figure 6.12. So as it seems the algorithm kept one disparity from each level. PCA graph 6.12 shows that disparity map 3 and disparity map 4 are too close while 1 and 2 are not.

Table 6.3 shows the sizes of the calculated gray-scale disparity maps compared to the sizes of the captured colored images. So we gained in terms of memory between 8/86 KB (90.7%) and 4/123 KB (96.8%).
Figure 6.10: Real house scene image set from outdoor

Table 6.3: Size of images in tree topology

<table>
<thead>
<tr>
<th></th>
<th>Im 1</th>
<th>Im 2</th>
<th>Im 3</th>
<th>Im 4</th>
<th>Im 5</th>
<th>Im 6</th>
<th>Im 7</th>
<th>Im 8</th>
<th>Im 9</th>
<th>Im 10</th>
<th>Im 11</th>
<th>Im 12</th>
<th>Im 13</th>
<th>Im 14</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image size on disk</td>
<td>86 KB</td>
<td>90 KB</td>
<td>94 KB</td>
<td>98 KB</td>
<td>94 KB</td>
<td>94 KB</td>
<td>123 KB</td>
<td>74 KB</td>
<td>78 KB</td>
<td>82 KB</td>
<td>82 KB</td>
<td>86 KB</td>
<td>82 KB</td>
<td>82 KB</td>
</tr>
<tr>
<td>DM 1</td>
<td>DM 2</td>
<td>DM 3</td>
<td>DM 4</td>
<td>DM 5</td>
<td>DM 6</td>
<td>DM 7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Disparity size</td>
<td>8 KB</td>
<td>4 KB</td>
<td>4 KB</td>
<td>4 KB</td>
<td>4 KB</td>
<td>4 KB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 6.13 shows the 3D reconstruction result applied on a pair of images based on DM 1. The results are not perfect, this is because our images pairs are not rectified,
CHAPTER 6. DISPARITY BASED RECONSTRUCTION MODEL

(a) DM 1  
(b) DM 2  
(c) DM 3

(d) DM 4  
(e) DM 5  
(f) DM 6

(g) DM 7

Figure 6.11: House disparity maps

Figure 6.12: Tree scenario PCA results
6.5. EXPERIMENTATION RESULTS

compared to the good 3D reconstruction result done using Middlebury dataset shown in the previous Figure 6.7.

(a) House Left Image
(b) House Right Image
(c) House DM
(d) House 3D reconstruction

Figure 6.13: House 3D reconstruction results

6.5.3/ ANOTHER SENSOR DEPLOYMENT TOPOLOGY

This scene is also in outdoor, for monitoring a parked car purpose. There is no need for a huge amount of sensors. The best topology in this case is ring. To achieve it, we took a fixed camera on a tripod and we measured 3 meters from the car and start taking picture every 0.9 meters around the car.

For the first experience, same amount of pictures was taken so the expected number of disparity maps at the first level is 7.
In this case the algorithm kept a higher number of disparity maps, compared to the previous scenario, as shown in Figure 6.16. The main reason is that each pair of images are pointing to a different specific part of the car.
As we can see in the Figure 6.16, the orange and the green points represent two close disparity maps. So if we look on the captured images, Image 2, Image 3, Image 4, and
Image 5 are pointing on the same surface of the car. This ensures that we have a correct classification.

Table 6.4 shows the sizes of the calculated gray-scale disparity maps compared to the sizes of the captured colored images. So we gained in terms of memory between 2/81 KB (97.6%) and 4/74 KB (94.6%).

<table>
<thead>
<tr>
<th>Image size on disk</th>
<th>Disparity size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Im 1</td>
<td>DM 2 KB</td>
</tr>
<tr>
<td>Im 2</td>
<td>DM 4 KB</td>
</tr>
<tr>
<td>Im 3</td>
<td>DM 4 KB</td>
</tr>
<tr>
<td>Im 4</td>
<td>DM 4 KB</td>
</tr>
<tr>
<td>Im 5</td>
<td>DM 4 KB</td>
</tr>
<tr>
<td>Im 6</td>
<td>DM 4 KB</td>
</tr>
<tr>
<td>Im 7</td>
<td>DM 4 KB</td>
</tr>
<tr>
<td>Im 8</td>
<td>DM 4 KB</td>
</tr>
<tr>
<td>Im 9</td>
<td>DM 4 KB</td>
</tr>
<tr>
<td>Im 10</td>
<td>DB 4 KB</td>
</tr>
<tr>
<td>Im 11</td>
<td>DB 4 KB</td>
</tr>
<tr>
<td>Im 12</td>
<td>DB 4 KB</td>
</tr>
<tr>
<td>Im 13</td>
<td>DB 4 KB</td>
</tr>
<tr>
<td>Im 14</td>
<td>DB 4 KB</td>
</tr>
</tbody>
</table>
6.5. EXPERIMENTATION RESULTS

6.5.4/ INDOOR SCENE

This scene is captured indoor inside a messy balcony filled with stuff. One angle is pointing outside the balcony while two others are pointing inside of it. This scene is captured at night around 21:00 under neon lighting as shown in Figure 6.17.

![Images of the indoor scene](image)

In this scene, a pre-calculation was made to pair sensors. So the results were: \{1, 2\} – \{3, 4\} – \{5, 6\} – \{7, 8\} as expected. Next, four disparity maps calculated as shown in Figure 6.18.

**Table 6.5** shows the sizes of the calculated gray-scale disparity maps compared to the sizes of the captured colored images. So we gained in terms of memory 8/61 KB (86.9%).

**Table 6.5: Size of images for indoor scene**

<table>
<thead>
<tr>
<th></th>
<th>lm1</th>
<th>lm2</th>
<th>lm3</th>
<th>lm4</th>
<th>lm5</th>
<th>lm6</th>
<th>lm7</th>
<th>lm8</th>
</tr>
</thead>
<tbody>
<tr>
<td>image size on disk</td>
<td>61 KB</td>
<td>57 KB</td>
<td>66 KB</td>
<td>61 KB</td>
<td>70 KB</td>
<td>70 KB</td>
<td>61 KB</td>
<td>53 KB</td>
</tr>
<tr>
<td>disparity size</td>
<td>8 KB</td>
<td>4 KB</td>
<td>4 KB</td>
<td>8 KB</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

So if we want to predict, pictures pointing to the center of the balcony must be filtered and one disparity map will be discarded. So after PCA classification, we noticed that we only
have 3 disparity maps as expected and shown in Figure 6.19.
In this chapter, we introduced a distributed 3D scene reconstruction model for WMSNs. Our model is tree based on DMs and distributes the DMs calculation between different couples of camera sensor nodes. The use of DMs is the main added value in our work, since all existing researches rely on images but not depth data. The MSE discards unnecessary similar views, so decreases the transmission cost over the network. PCA regroups and clusters the calculated disparity maps on the base station level, in order to apply the 3D scene reconstruction on these DMs. Our experimentations ensured the efficiency of our framework in terms of clustering fidelity, gain in size, power consumption and number of transmissions between nodes, fast detection of change(s) or event(s) on sensor nodes level, and 3D scene reconstruction result(s) quality.

For future work, many tools could be used to expand the WMSNs lifetime and to 3D reconstruct targets faster. This is related to the improvement(s) in the development of new updated camera sensors with high specifications and more power capacities. We need to make real experimentations on newest camera sensors available in the market, by building real WMSN. We are looking to develop the reconstruction algorithm to reach better 3D mesh. Specially when it comes to rectification phase, we will work on implementing a fast and efficient rectification pre-calculation on the sensor node.
IV

CONCLUSION
7

CONCLUSION

7.1/ Thesis Summary, Back to the Main Contributions

In this thesis, the main focus had been on efficient 3D scene reconstruction using WM-SNs.

Traditional WMSN captures images where the real 3D scene is projected on a digital 2D image, thus loosing the depth information. 3D reconstruction is the inverse operation where we build the 3D scene from a set of captured 2D images, hence recovering the depth information. In order to achieve the 3D reconstruction objective, two phases are needed: matching and triangulation.

We have chosen stereo vision technology to calculate the matching, resulting disparity maps between a couple of sensors playing the role of left and right cameras. These disparity maps are low sized gray-scale images, holding the 3D information. So transferring these disparity maps decreased the cost of transmission between the nodes in the network. And at the server level, it is the main parameter to make a 3D scene reconstruction of our target of interest by applying the triangulation.

Exploring the 3D depth information of a monitored object of interest or target was advantageous since it is adequate to detect a change or intrusion in the environment. Since most of intrusions scenarios are represented by a change in the dimensions, width, depth or position. Any change or intrusion will be detected on the sensors level before reaching the server because it is related to the change in the depth information, which is saved in the DM. And in case we need to reconstruct the scene, these disparity maps with few reference images will achieve the 3D scene reconstruction on the server level by applying a triangulation.

Such processing could be very advanced and expensive in terms of energy and com-
putation. So we started by studying all WMNs applications, constraints, limitations and
schemas in Chapter 2. The main limitation of the sensors is the power consumption,
specially that these nodes are feed by batteries. These sensors have low computation
(processing) capabilities, so this limitation should be taken into consideration, specially
when it comes to advanced and hard image or video processing. Without missing the
architecture of these WMSNs, this is complementary to the sensors deployment study
and how the DM calculation and transmission is done over the network that we have
made in Chapter 5 and Chapter 6.

In Chapter 3, we had reviewed all DM computation techniques and algorithms applied
on computers, graphic cards and micro-controllers. Our review started by explaining all
DM algorithms and phases and divided techniques into features based and pixel based
methods. We recommended pixel based techniques, and specially local methods since
global methods needs advanced hardware requirements that meets video cards but
not low cost sensors. At the end, we regrouped them in tables by studying the main
parameters and limitations that define a good DM computation technique. Starting by
technical requirements like image source quality, disparity range, clock frequency and
frame per second. Ending by qualitative study that takes into consideration occlusion
handling, depth discontinuity, uniform textures, texture-less areas, computation cost,
power consumption, possibility of real time implementation and DM quality for indoor or
outdoor scenes. This review helped us to choose a couple or family of solutions in order
to implement them on our sensors.

In Chapter 4, we explained the concept of 3D scene reconstruction by focusing on
the main parameters to achieve this task. So our DMs will serve as main parameters
for our 3D reconstruction triangulation based algorithm for the monitored scene. The
Triangulation established 3D coordinates based on the 2D images correspondences
retrieved during the matching process (calculation of DMs).

In Chapter 5, we have introduced the use of DMs in the context of WMSNs. We
experimented different scenarios and scenes by changing the models, environments,
number of nodes, order of coupling, images resolutions, distances between sensors,
orientations of the cameras, and field of views. We used SAD algorithm as it is the
lowest solution in terms of complexity and computation cost so an acceptable power
consumption.

The virtual simulations, where we modeled a 90 percent similar to real world outdoor
and indoor scenarios, ensured for us how it is recommended to deploy different sensors.
This established an efficient monitoring of a scene and resulted good DMs. This
phase is very essential because in some scenarios, it is recommended to decrease
the distance between the pair of sensors (baseline). Matching with small baseline was
easier than with wide baseline, because the depth uncertainty is directly proportional to
this parameter. Otherwise, we risked to have a far bad matching because each sensor
node is capturing a different scene. Same for the angles, in some cases rotating the
camera sensors affects in a bad way the DMs. In other cases, we have a bigger range of acceptable angles. Thus directly affecting the needed number of deployed sensors.

Finally, in Chapter 6 we have defined the 3D scene reconstruction model. This model could be used for any monitoring scenario by saving computation, energy consumption and transmission of data. This is a compromise between the required quality of reconstruction and hardware specs of the system. So no more need to transfer a big number of images, but only a black and white focused disparity maps. Even not all the DMs are transferred, because we applied MSE to threshold the similarity between the calculated DMs and discard similar ones. We experimented different network topologies applied on multiple outdoor and indoor scenes. On the base station level, we used PCA as a clustering technique in order to regroup DMs, that represent and focus on the same object or part from the scene. In this way, our 3D reconstruction triangulation based algorithm won’t use all DMs, but only the matching ones.

7.2. FUTURE WORKS AND PERSPECTIVES

Research is to see what everybody else has seen, and to think what nobody else has thought, Albert Szent-Gyorgyi. All existing research works shown how WMSNs are practical and advantageous for surveillance and monitoring, independently from the domain like E-Health, Military, Habitat, etc. Stereo Vision was used to intercept and interpret the 3D information for cinematography, 3D games, and multiple applications. What we have done in this thesis is to use the stereo images for multiple views taken from different sensors in a WMSN. We have proven the concept of calculating DMs between sensors, but not only stereo 3D cameras or advanced equipments. But this main contribution or added value was not accomplished perfectly, specially that we are facing some limitations in the resources we have. The calculated DMs and reconstructed scenes needs more improvements in terms of quality. In this thesis, our approach for qualitative experimentations used SSIM and PSNR to define the qualities of the results. We don’t have a ground truth DM or 3D model, calculated with an advanced laser scanner, as a high quality reference to compare with it. So we may find a new way to achieve this job in the future.

One of the problems is the rectification between the pair of images, this can be studied later in order to implement efficient rectification phase algorithm on the node level. Another future work is to complete the real experimentations and synchronize with the development of cheap sensors technologies in the market. We are seeking to reach a real-time efficient 3D scene reconstruction that lasts for long time and shows high quality mesh results.

We are looking to widen our experimentations by implementing all the scenarios and possible environments. Specially that each scenario or scene have its own constraints like reflection, distortion, image perspective, repetitive textures (grass for example),
lighting conditions, shadows, occlusions, etc. We will update our 3D reconstruction algorithm, implemented on the base station, by adding a module that merge different DMs and 3D reconstruct the scene with more and better details.

We intend to investigate another new reconstruction models and compare them to our distributed method. Thus opening the questions for new sensors deployments. We will work on implementing two aligned camera modules on each nodes thus transforming each node to a stereo camera.

Our approach to decrease the power consumption was based and directly dependent from the computation cost, complexity of the algorithms, transmission rate and the sizes of the data. So completing the real experimentations, that we started, will help us later on to diagnose on real time the power consumption in the WMSN.


[52] Shawn McCann. 3d reconstruction from multiple images, 2016.


[54] Brice Michoud, Erwan Guillou, Héctor M. Briceño, and Saïda Bouakaz. Removing camera placement constraints in shape from silhouette on large acquisition volumes.


