AUTOMATIC REAL-TIME 2D-TO-3D VIDEO CONVERSION

by

Abrar Wafa

B.A.Sc in Electrical and Computer Engineering, Effat University, 2012

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

(Electrical and Computer Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

July 2016

Abstract

The generation of three-dimensional (3D) videos from monoscopic two-dimensional (2D) videos has received a lot of attention in the last few years. Although the concept of 3D has existed for a long time, the research on converting from 2D-to-3D in real-time is still on going.

Current conversion techniques are based on generating an estimated depth map for each frame from different depth cues, and then using Depth Image Based Rendering (DIBR) to synthesize additional views. Efficient interactive techniques have been developed in which multiple depth factors (monocular depth cues) are utilized to estimate the depth map using machine-learning algorithms. The challenge with such methods is that they cannot be used for real-time conversion. We address this problem by proposing an effective scheme that generates high quality depth maps for indoor and outdoor scenes in real-time.

In our work, we classify the 2D videos into indoor or outdoor categories using machine-learning-based scene classification. Subsequently, we estimate the initial depth maps for each video frame using different depth cues based on the classification results. Then, we fuse these depth maps and the final depth map is evaluated in two steps. First, depth values are estimated at edges. Then, these depth values are propagated to the rest of the image using an edge-aware interpolation method. Performance evaluations show that our method outperforms the existing state-of-the-art 2D-to3D conversion methods.

Preface

This thesis is an original intellectual product of the author, Abrar Wafa. All of the work presented henceforth was conducted in the Digital Multimedia Laboratory at the University of British Columbia, Vancouver campus.

A version of chapter 4 has been published in the proceedings of the *IEEE Quality of Multimedia Experience (QoMEX), 2015 Seventh International Workshop* as "Automatic Real-time 2D-to-3D Conversion for scenic views" co-authored by Wafa, A., Nasiopoulos, P., Leung, V. C., and Pourazad, M. T. (2015, May). I was the lead investigator responsible for all areas of research, data collection, as well as the majority of manuscript composition. M. T. Pourazad was involved in the early stages of research concept formation and aided with manuscript edits. P. Nasiopoulos was the supervisor on this project and was involved with research concept formation, and manuscript edits.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

2D	Two Dimensional
3D	Three Dimensional
3D-TV	Three Dimensional Television
ARPS	Adaptive Rood Pattern Search
CfP	Call for Proposal
CfE	Call for Evidence
DIBR	Depth Image Based Rendering
fps	Frames per second
HEVC	High Efficiency Video Coding
HVS	Human Visual System
ITU-T	International Telegraph Union-Telecommunication Standardization Sector
JCT-VC	Joint Collaborative Team on Video Coding
MB	Macro-block
MOS	Mean Opinion Score
MPEG	Moving Pictures Experts Group
MV	Motion Vector
PSNR	Peak Signal-to-Noise Ratio
TV	Television
VSRS	View Synthesis Reference Software

Acknowledgements

First of all, all praises go to **Allah** most merciful for his guidance.

I would like to start by expressing my utmost gratitude to my supervisor, **Dr. Panos Nasiopoulos** for his support through the past two years and for being a constant source of guidance and inspiration through normal and extraordinary circumstances.

I would also like to thank **Dr. Mahsa T. Pourazad** for her time, thorough feedback, support, and help during different stages of this thesis.

My gratitude goes to **Dr. Rabab Ward** for her passionate help and support during my first year in UBC and Vancouver.

I am incredibly grateful to **my mother and father**, who always believed in me and never stopped encouraging me. I am thankful for their love and their unwavering dedication to my education throughout the years. For all the sacrifices they made for me so that I can be who I am today.

I express my sincere and wholehearted thanks to my husband, my love, **Omar** for encouraging me to go after my passion and believing in me. Without his support, I would not have been able to complete this work. I am thankful for his help in taking care of our daughter and sharing the housework during the most difficult time of working on this thesis.

I am thankful to my siblings, **Hatem**, **Ebtihal**, and **Fouad,** for their love and emotional support throughout my whole life.

To my daughter **Serene**, thank you for being my guiding light.

Finally, many thanks to the **Saudi government** and **Prince Sultan University** for their funding and financial support.

Dedication

To my beloved family

Chapter 1: Introduction

Three-dimensional (3D) video technology imitates the stereoscopic perception of the human visual system, providing the viewer with a more realistic and immersive viewing experience. 3D allows viewers to perceive depth the same way as if they are looking at a live scene. The term 3D in this context denotes *stereoscopic*, which means a two-view system that is used for a close to real-life visualization. Such stereoscopic videos that are displayed on three-dimensional televisions (3D TVs) can increase the visual impact and heighten the sense of presence for viewers [1]. The visual ability to perceive the world in 3D is identified as *depth perception*, which arises from a variety of depth cues that determine the distances between objects.

1.1 Motivation

The technologies behind 3D displays and digital video processing are reaching technical maturity, with the stereoscopy technology being widely regarded as one of the major advancements in the entertainment industry. Although the concept of 3D has existed for a long time, the research on converting 2D to 3D in real-time is still ongoing [2]. The vast amounts of existing 2D video content and their conversion to 3D is one of the highest priorities of content owners such as Hollywood studios and broadcast facilities. As 3D content is limited and capturing 3D is still a challenging and costly process, it is imperative that converting 2D content to 3D will play an important role in enabling the 3D consumer market. TV manufacturers have tried to address this issue by introducing 2D-to-3D real time conversion but with limited success, as the resulting quality leaves a lot to be desired.

Figure 1.1. Example of an original image and its corresponding depth map

Lately, TV manufacturers are placing their hopes for the future of 3D TV on the so-called “glasses-free” 3D TV technology, where “autostereoscopic” displays show multiple views of 3D (ranging from 8 to more than 100) without the need for glasses. In this attractive technology, rendering/synthesizing views from an existing view is a must, as it is not practical to transmit multiple views or to know how many views each display supports. This challenge is similar to the 2D-to-3D conversion, as views need to be synthesized from an existing real view.

In general, the main purpose of the 2D-to-3D conversion is to generate a second view video based on the content of the 2D video. This conversion mainly involves two processes: Depth Map Estimation and Depth Image Based Rendering [2].

The depth map is an image/frame with different grey values that represents depth information. Each intensity value of the depth map represents the distance from the viewer, where the farther/closer the point in the image from the viewer, the darker/lighter the intensity value. Fig. 1.1 shows an example of an image and its depth map [3].

Synthesized views can then be generated from this depth map using Depth Image Based Rendering techniques.

1.2 Contribution

There has been a lot work on the conversion of 2D-to-3D as explored in Chapter 2. Our goal in this thesis is to design an efficient real-time 2D-to-3D conversion scheme that significantly outperforms previous approaches in terms of quality while maintaining real-time performance.

More specifically, we identify our contributions as follows:

- Implementation of a classification scheme to classify indoor and outdoor videos.
- Taking a comprehensive look at existing 2D-to-3D conversion techniques and extract depth cues that have been identified as important cues for estimating depth information.
- Using multiple features for each category to prevent our proposed scheme from failing in case one of the cues is absent from the scene.
- Estimating depth values at edges to get reliable depth estimation in those areas.
- Formulating an optimization problem to propagate depth values from edges to the entire frame.
- Verifying the quality of the produced 3D videos of our approach with a subjective evaluation.

1.3 Thesis Organization

In our work, we focus on the depth map generation from a single-view 2D video in real-time using multiple depth cues. The rest of this thesis is organized as follows: Chapter 2 is an overview of 3D and existing 2D-to-3D conversion techniques. Chapter 3 presents an overview of the proposed framework, and describes the implementation details of feature extraction and depth estimation. Chapter 4 discusses the experimental results and the subjective evaluation. Finally, Chapter 5 provides the concluding remarks and describes future work.

Chapter 2: Background

In this chapter, we describe how humans perceive depth (section 2.1) and look at the different technologies for generating 3D content (section 2.2). In section 2.3, we explain the different depth cues that can be used for perceiving depth. Finally, we discuss existing 2D-to-3D conversion techniques in section 2.4.

2.1 Human Depth Perception

Human beings have two eyes spaced a short distance apart, each eye taking a view of the same scene from a slightly different angle. The two eye views have plenty in common, but each eye captures its own view and the two separate images are sent on to the brain where they are fused into one picture as shown in Figure 2.1 [4].

The brain fuses the two views by matching up the similarities and adding in the small differences along with other cues to finally generate a three-dimensional stereo picture. Stereo is the added perception of the depth dimension that makes stereovision so rich and special. With such stereovision, we see an object as solid in three spatial dimensions: width, height and depth.

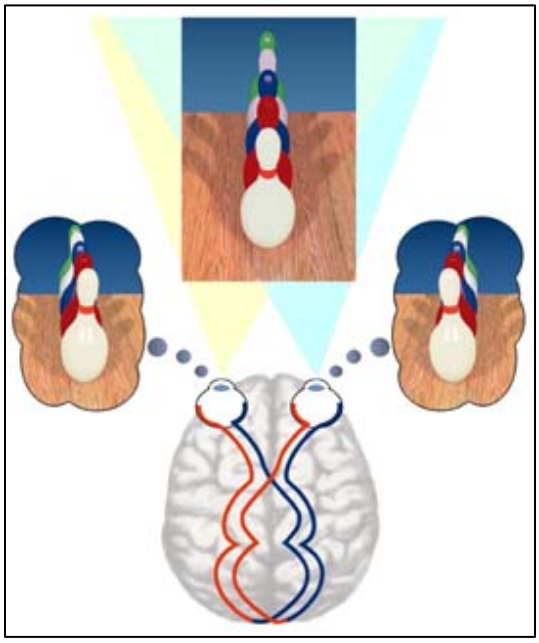


Figure 2.1 How our brain perceives depth [4]

2.2 Three-dimensional Technology

Three-dimensional videos are moving our visual entertainment towards a greater perceptual realism. They are attracting more attention in the different applications of digital multimedia ranging from medicine and education to training and entertainment.

There are several different ways of generating 3D content as shown in Table 2.1. Regardless of the differences in the technologies for capturing 3D content, all of them use the same principle of producing two separate views. The stereoscopic displays then send one of these views to the viewer's left eye and the other to the right, in order to give the proper illusion of 3D.

Information can be captured with 3D depth range cameras. These cameras use a depth-sensing technology that separates objects from the background layers behind them and calculates the distance between objects, in order to produce depth and from that a second view and finally 3D video. Another way to produce 3D content is by using stereoscopic dual cameras to capture a scene from slightly two different angles, and produce two separate views.

Of course, another approach to generate 3D content is from 2D content. Legacy 2D content captured by traditional 2D cameras may be converted into 3D format if depth is somehow estimated from the existing view and then it is used to synthesize the other view.

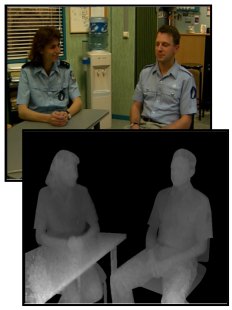
Stereoscopic Dual Camera	3D Range Camera	2D Video Camera
		
↓	↓	↓
 Left and right views	 Original image and Depth map	 Original Image Depth ↓ Estimation  Estimated depth map
↓	↓	↓
Left and right views	Left and right views	Left and right views
↘	↓	↘
View Synthesis		
↓		
3D image		

Table 2.1: Different Systems for Generation 3D Content

2.3 Depth Cues

When observing the world, the human brain usually integrates several different depth cues for the generation of depth perception. Thus, depth perception occurs as a result of observing a variety of depth cues (throughout this thesis, we sometimes refer to depth cues as image features). These cues are typically classified into binocular and monocular cues. Binocular cues are based on the reception of information in three dimensions from two eyes, while monocular cues can be represented in two dimensions and observed with just one eye.

2.3.1 Binocular

Binocular cues represent retinal disparity (also known as binocular disparity), which is caused by the fact that each of our eyes views a scene from a slightly different angle. As a result the same object appears at a slightly different position in the two views, with the distance between them known as *binocular disparity*. Our brain uses this disparity to extract depth information from the 2D retinal images in stereopsis.

Therefore, depth information can be computed using binocular disparity by calculating the difference between the positions of the same objects in the scene, which is captured from slightly different viewpoints. “First, a set of corresponding points in the image pair is found. Then, by means of the triangulation method, the depth information can be retrieved when all the parameters, i.e., camera parameters, of the stereo system are known.” [5].

According to [5], one of the most active research areas in computer vision is stereo correspondence problem (also known as stereo matching). It is defined as how one can find the matching point of the left image in the right image. Such a problem is considered to be a very time-consuming aspect of depth estimation algorithms based on binocular disparity. This is due to the inherent ambiguities of image pairs such as occlusion.

2.3.2 Monocular

Monocular cues, which is the other class of depth cues, provide depth information from a single image. These cues can be classified into two main categories. The first category is motion-based cues, which include motion parallax, velocity of motion, and occlusion from motion. The second one is pictorial cues that are depth cues in monoscopic still images. They include occlusion, linear perspective, vertical edges, texture gradient, blur, light and shading, and size. The explanation on how these cues provide depth information is detailed below.

2.3.2.1 Motion-based Cues

- *Motion parallax:* objects' patterns move within a frame to form corresponding objects on the subsequent frame. Hence, the difference in spatial domain of objects across time is what defines motion parallax [6].
- *Velocity of motion*: the nearby object corresponds to a larger displacement in a video sequence compared to the farther one, when the two objects move at a similar velocity [4].

- *Occlusion from motion*: background regions occluded by a moving foreground object will be exposed in another frame [7].

2.3.2.2 Pictorial Cues

- *Interposition/Occlusion*: an object that occludes another is closer. Interposition is considered to be a strong depth cue but it only provides information about depth order not magnitude [8].
- *Linear perspective*: Parallel lines will vanish toward one point on the horizon, so the distance between the lines decreases visually towards the horizon [9].
- *Texture gradient*: closer objects have more detailed and visible texture than farther ones. Texture in farther objects is smoother and finer [10].
- *Blur from defocus*: the object closer to the camera is usually clearer than farther objects, as blur of defocused regions increases with objects distant away from the focal plane [11].

In addition, objects characteristics in the image can be also used as indicator of how far they are located in the scene.

- *Light and Shading*: Shadows give us information about the shape and depth of objects.
- *Size*: Similar objects with different sizes are perceived at different distances. Larger objects are closer.
- *Brightness:* farther away objects are dimmer.

2.4 2D-to-3D Conversion Techniques

Current 2D-to-3D conversion techniques are based on generating an estimated depth map for each frame utilizing such different depth cues. Depending on the number of input images, we can categorize the existing conversion algorithms into two groups: algorithms based on a single image and algorithms based on two or more images.

2.4.1 Based on Two Images

In the latter case, the algorithm uses two or more input images taken by multiple fixed cameras at slightly different viewing angles. Such techniques use binocular depth cues to estimate the depth map. This is achieved by finding a set of corresponding points in the image pair.

Assume a 3D point P has two projections p_l and p_r on a left and right images when captured by stereo dual camera system that has left and right cameras, where the origin of the camera coordinate systems is O_l and O_r. As shown in Figure 2.2 [5], the depth value Z of the point P can be computed based on the relationship between similar triangles (P, p_l, p_r) and (P, O_l, O_r) as follow:

$$Z = f\frac{T}{d}$$

where $d = x_r - x_l$, which measures the difference in retinal position between corresponding image points [5]. This disparity value of a point is important as it helps in constructing the disparity map, which is essential to estimate the depth map.

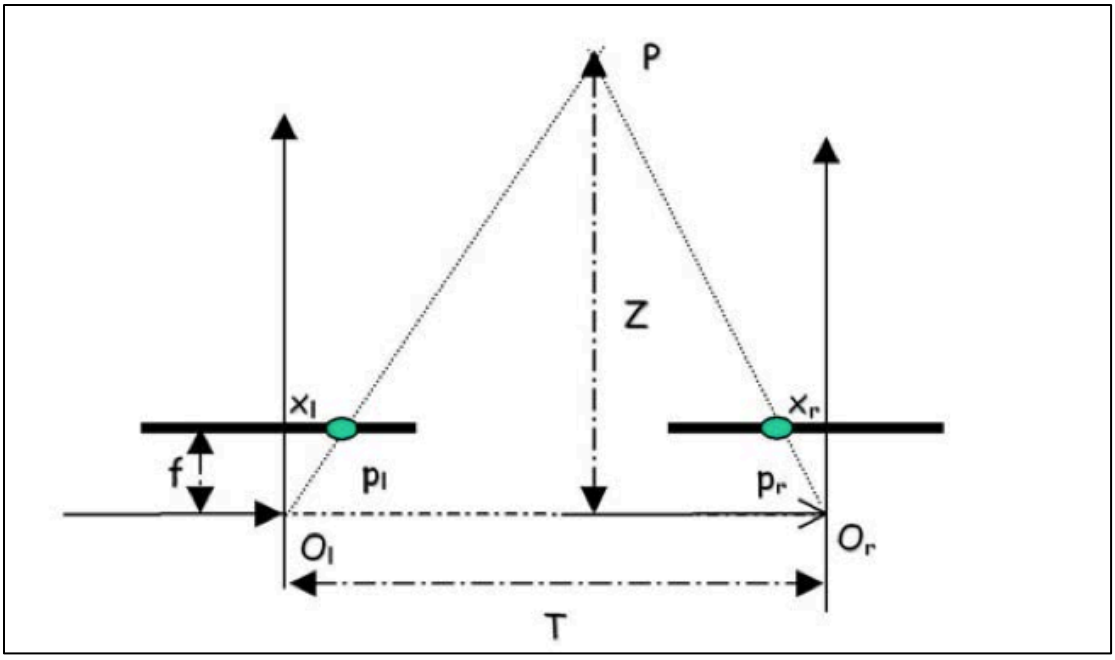


Figure 2.2 Depth Value Z of Point P (captured by stereo dual camera system) is computed based on the relationship between similar triangles (P, p_l, p_r) and (P, O_l, O_r)

As stated earlier, stereo matching problem is regarded as the most time-consuming aspect of depth estimation and it is difficult to solve. In the literature, several constraints have been introduced to make the problem solvable. The taxonomy in [12] has evaluated the performance of approximately 40 two-frame stereo correspondence algorithms, which impose various sets of constraints.

2.4.2 Based on a Single Image

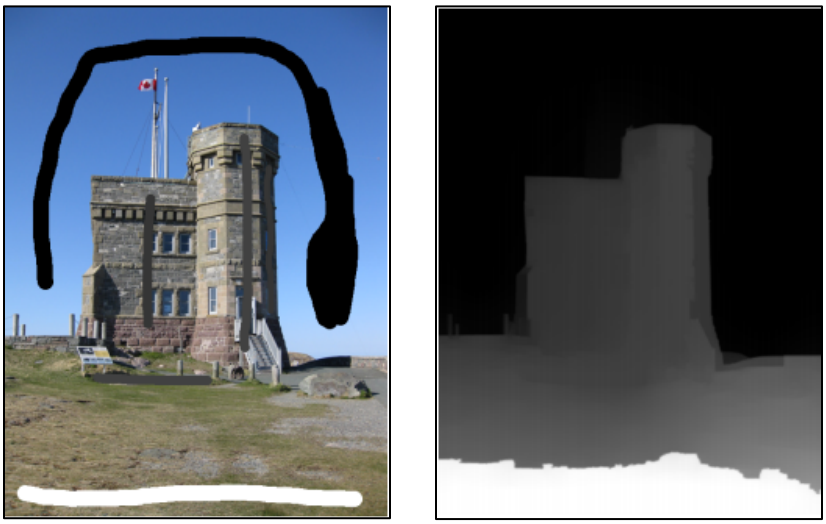
On the contrary, the former group of conversion algorithms uses monocular depth cues from a single image to produce the depth map. Then, a Depth Image Based Rendering (DIBR) technique is used to synthesize the additional views [13].

Over the past few years, many frameworks have been proposed for depth map creation based on monocular cues. These frameworks can be classified into three schemes: manual, semi-automatic, and automatic.

Manual schemes outline objects manually and associate them with an artistically chosen depth value. Some of the semi-automatic approaches outline objects with corrections made manually by an operator [14]. Other semi-automatic conversion techniques involve a skilled operator who assigns different depth values to different parts of an image as shown in Figure 2.3 (a). Then, an automatic algorithm, which employs Random Walks while incorporating information from Graph Cuts segmentation paradigm, estimates depth map over the entire image based on the user-defined manual assignments, as shown in Figure 2.3 (b) [15].

Even though, such techniques produce high quality depth maps, yet, they are time consuming and expensive. Furthermore, they are unreliable when complex outlines are encountered [16].

Lowering the conversion cost while speeding up the process, motivated the researchers to automate the conversion techniques. As a result, automated approaches, which do not require human interaction, have been proposed for synthesizing stereoscopic videos from monoscopic videos.



(a) Labeled Image (b) Depth Map

Figure 2.3 Semi-automatic Depth Estimation (a) Original 2D Labeled Image and (b) Depth Map estimated using semi-automatic approach

Some of these methods use only one monocular depth cue is used for depth map estimation. For instance, relative motion of objects is extracted from the video and used for estimating the depth map in [17]–[22]. Other studies propose estimating depth from defocus using a single image [23]–[25]. One of the major drawbacks of methods using a single depth cue is that they fail to work in the absence of that cue.

Subsequently, efficient interactive techniques have been developed in which several monocular depth cues are utilized to estimate the depth map using machine-learning algorithms [3], [26]. The challenge with these approaches is that they either cannot be used for real-time conversion as feature extraction even for testing is time consuming [3] or they produce depth maps that contain blocky artifacts or inaccurate edge estimations [26].

For real-time applications, commercialized 3D TVs are using limited a number of depth cues such as vertical edges, motion, haze, and perspective for converting 2D content to 3D automatically [27]–[29]. Some of the TVs categorize the scene to outdoors, indoor, and scenic, and depending on the category of the scene, they use different cues and methods for 2D to 3D video conversion [27]. In general, the perceptual quality of the generated 3D views using the 2D-to-3D conversion option of the existing 3D TVs is quite inconsistent and is content dependent.

In this thesis, we propose a unique real-time method for generating a high quality depth map for indoor and outdoor scenes, using multiple depth cues. We estimate our depth map in two steps. First, we estimate depth values at edges. Then, we propagate these depth values to the rest of the image using an edge-aware interpolation method. For performance evaluations, we compare our method with the defocus depth estimation method [23], since the latter also uses edge-aware interpolation method for depth map generation. In addition, we apply our method to a test dataset and compare the quality of the synthesized 3D content subjectively with the quality of 3D views automatically generated using the existing state-of-the-art 2D-to-3D conversion method and the 2D-to-3D option of a commercialized 3D TV.

Chapter 3: Our proposed 2D-to-3D conversion method

In this chapter, we present an overview of our proposed scheme in section 3.1. Further, we discuss the implementation details of the classification scheme used to categorize indoor and outdoor videos in section 3.2. We then, in section 3.3, identify and explain the depth cues extracted for each category. Finally, we describe the proposed approach for generating the full depth map in section 3.4.

3.1 Overview of Our Proposed Framework

In general, 2D-to-3D video conversion schemes try to estimate the depth map of a scene using information/cues present in the original 2D stream. In our 2D-to-3D video conversion scheme, we analyze the content of the 2D video and according to the obtained results decide on the 2D-to-3D video conversion approach for generating the three-dimensional video. More specifically as shown in Fig. 3.1, first we classify the 2D videos into indoor or outdoor categories using machine-learning-based scene classification in accordance with the low dimensional spatial information of the scene. We then estimate the depth map for each video frame using different depth cues based on the classification results. We use multiple features for each category to prevent our proposed scheme from failing in case one of the cues is absent from the scene. For indoor scenes we use four depth cues: 1) blur, 2) motion, 3) occlusion, and 4) linear perspective. On the other hand, for outdoor scenes we use three depth cues: 1) motion, 2) blur, 3) haze, and 4) vertical edges. These depth cues are chosen because in the existing literature they have been identified as important cues for estimating depth information [3], [23], [24], [26], [27], and their extraction is not computationally expensive,

which suits real time applications. We generate depth map for each cue, and then use linear depth fusion to produce the final depth map.

We apply canny edge detector to each frame of the original video. Using the aforementioned depth cues, our scheme first estimates a sparse depth map at the edges of each frame. However, noise and weak edges may cause inaccurate depth estimates at some edge locations. To solve this problem, we apply joint bilateral filtering [30] on the sparse depth map to refine inaccurate depth estimations. Then, we propagate this sparse map to the entire frame in order to obtain the full depth map. In order to do that, we formulate an optimization problem to find the full depth map that is close to the sparse depth map by minimizing a cost function. Then, a temporal median filter [31] is applied to the estimated full depth map as a post-processing step to ensure temporal consistency and prevent flickering. The following sections provide a detailed description of the steps and process involved in our approach.

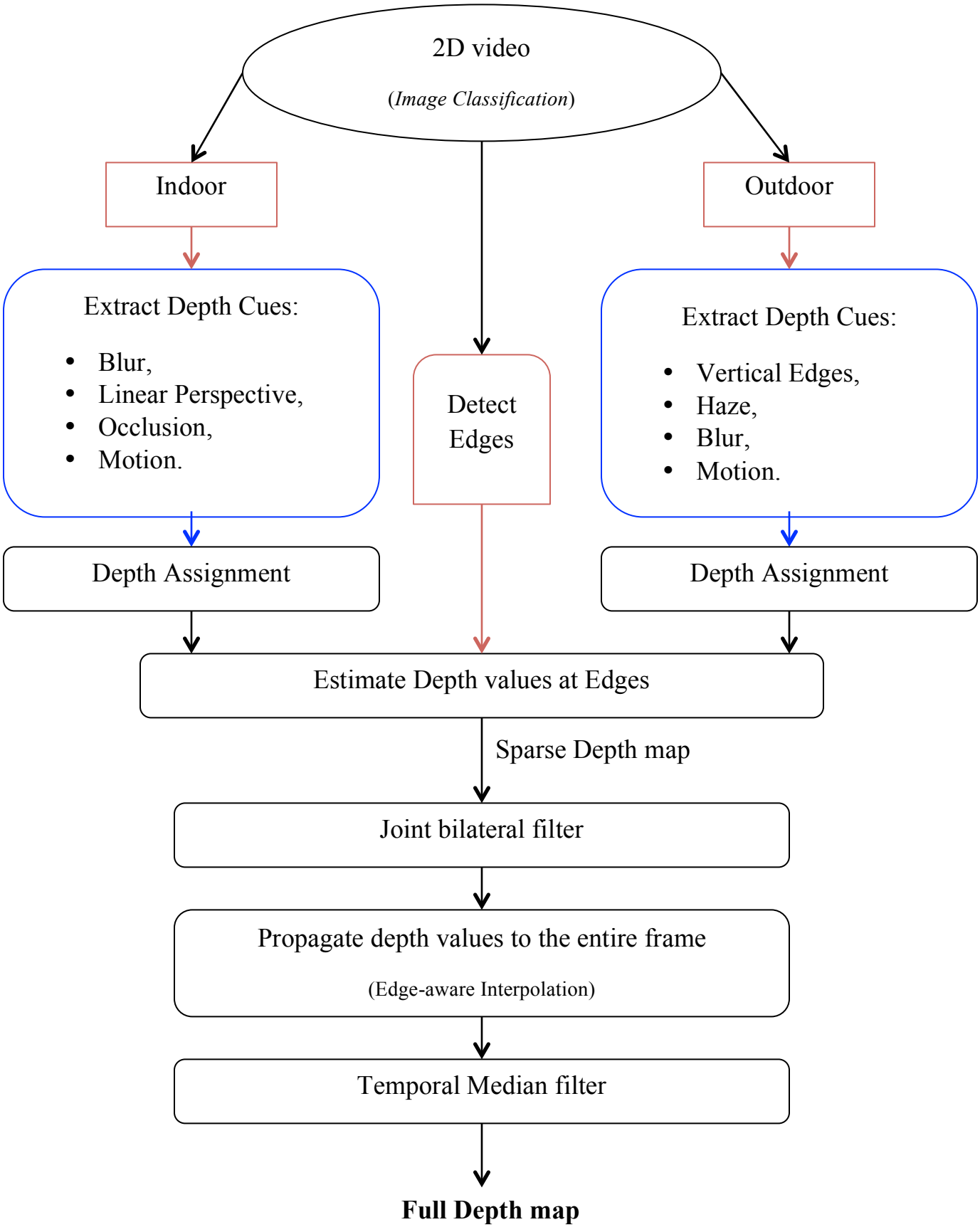


Figure 3.1 Overview flowchart of our proposed approach

3.2 Classification

Our visual system uses different depth cues for indoor and outdoor scenes, for instance while haze information is an indication of distance in outdoor scenes, for indoor scenes it is not being used. To imitate human visual system for depth estimation, in our proposed scheme the first step is to categorize the 2D video to indoor and outdoor scene. To this end, the structure or “shape of a scene” is estimated using a few perceptual dimensions specifically dedicated to describe spatial properties of the scene. As it is shown in [32] the holistic spatial scene properties, termed Spatial Envelope properties, can be reliably estimated using coarsely localized information. The scene representation characterized by the set of spatial envelope properties provides a meaningful description of the scene semantic category. These spatial envelope properties are then used as the input of a Support Vector Machine (SVM) [33] for the purpose of scene classification.

According to [32], spatial envelope properties represent the scene in a very low dimensional space in which each dimension depicts a meaningful property of the space of the scene. A set of perceptual dimensions (naturalness, openness, roughness, expansion, and ruggedness), that represents the dominant spatial structure of a scene, is estimated using coarsely localized information. As suggested in [32], we utilize the windowed Fourier transform (WFT) to estimate the localized energy spectrum (spectrogram) as follows:

$$A(x, y, f_x, f_y) = \left| \sum_{x', y'=0}^{N-1} i(x', y') H_r(x' - x, y' - y)\, e^{-j2\pi(f_x x' + f_y y')} \right| \qquad (1),$$

where $i(x',\ y')$ is the intensity distribution of the image along the spatial variables (x, y), f_x and f_y are the spatial frequency variables, and $H_r(x',\ y')$ is a hamming window with a circular support of radius *r* which provides localized structural information.

The estimation of the spatial envelope attributes from image-based features can be solved using different regression techniques. In the case of a simple linear regression, the estimation of a scene attribute s from the spectrogram features w can be written as:

$$\hat{s} = w^T d = \sum_{i=1}^{N_L} w_i d_i = \sum_x \sum_y \int \int A(x, y, f_x, f_y)^2 \times WDST(x, y, f_x, f_y) df_x df_y \quad (2),$$

with:

$$WDST(x, y, f_x, f_y) = \sum_{i=1}^{N_L} d_i \Psi_I(x, y, f_x, f_y) \quad (3),$$

where the vector d contains the WDST parameters. The WDST (Windowed Discriminant Spectral Template) describes how the spectral components at different spatial locations contribute to a spatial envelope property [34].

The template indicates that indoor scenes are characterized by low spatial frequencies at the bottom of the scene (mainly due to large surfaces), by vertical structures (wall edges, tall furniture, etc.) in the central-top part, and by horizontal structures at the top (e.g., ceiling). Outdoor scenes are mainly characterized by high spatial frequencies everywhere.

Our classification task involves separating the data into training and testing sets. The training set contains two *target values* (i.e., the class labels: indoor vs. outdoor) and

several *attributes* (i.e., the spectrogram features). The goal of the SVM classifier is to produce a model (based on the training data), which predicts the target values of the test data given only the test data attributes [35].

First, we obtain the scene attributes of 400 indoor and outdoor images from the SUN database [36], and we train the SVM classifier using these attributes with optimized parameters of the kernel function. Then, we test the model on the video sequences provided to MPEG for 3D CfP [37] and FTV CfE [38] for comparability reasons (training dataset is excluded and not used for testing). For indoor scenes, we used: Balloons (250 frames), Poznan Hall (200 frames), Kendo (250 frames), and Newspaper (250 frames) video sequences. For outdoor scenes, we used: Poznan Street (250 frames), Ghost Town Fly (250 frames), and Bee (240 frames) video sequences. We managed to achieve a classification rate of 92% when testing the model on these video sequences.

3.3 Feature Extraction

Once the 2D videos are categorized to indoor and outdoor, we need to extract depth cues to estimate the depth map for different scenes. We use multiple features for each category to prevent our proposed scheme from failing in case one of the cues is absent from the scene. We chose these depth cues because they have been identified as important cues for estimating depth information in the existing literature [3], [23], [24], [26], [27], and their extraction is not computationally expensive, which suits real time applications. We generate depth map for each cue, and then use linear depth fusion to produce the final depth map. The following subsections elaborate on the depth cues used in our scheme for indoor and outdoor scenes.

3.3.1 Indoor Scenes

For indoor scenes, in our scheme the following depth cues are used: blur, linear perspective, occlusion, and motion parallax. The process for extracting each of these features is detailed below.

1) Blur

Blur is a feature that measures depth in the sense that distant objects suffer from more blur than nearby objects. In our approach, we use the blur estimation technique presented in [39] that uses the main square deviation of the point spread function (PSF) to measure the amount of blur, which depicts depth. A blurred image $I(x, y)$ can be regarded as a convolution between the focused image $f(x, y)$ and the PSF $h(x, y)$:

$$I(x,y) = f(x,y) \otimes h(x,y) \quad (4)$$

The PSF is a function that can be approximated by a Gaussian function. Therefore, we applying a Gaussian filter with small kernel to the original image to remove high-frequency noise and suppress the blur caused by blur texture:

$$h(x,y) = \frac{1}{2\pi\sigma_s^2} \exp\left(-\frac{x^2+y^2}{2\sigma_s^2}\right) \quad (5)$$

The standard deviation parameter σ of the Gaussian function represents the amount of blur. Thus, it is proportional to the blur diameter c, $\sigma = k.c$, where *k* is a scale parameter and *c* is the amount of blur. Therefore, the larger the variance, the more blur phenomenon and the farther the object.

The amount of blur can be estimated in the areas of the image that has significant frequency content (i.e. edge locations). Therefore, we compute the blur amount at edge locations using spectrum contrast *C*, which can be represented as the absolute value of spectrum amplitude difference between one pixel and its adjacent pixels as follows:

$$C(i) = \left\| \log\big(A(i)\big) - \frac{1}{N}\sum_{j \in B} \log\left(A(j)\right) \right\| \quad (6)$$

where *A(i)* represents the spectrum amplitude of edge pixel *I*, and B is the neighbourhood of pixel *i* given.

Finally, we estimate the relative depth using linear mapping [40] as follows:

$$D_{blur}(x,y) = 255.\frac{h(x,y)-\min\{h(x,y)\}}{\max\{h(x,y)\}-\min\{h(x,y)\}} \quad (7)$$

2) Linear Perspective

Indoor scenes usually contain linear perspective information, which can be used as a cue to estimate depth. For example, corridor walls can be characterized as parallel lines that appear to converge with distance, eventually reaching a vanishing point at the horizon. The more the lines converge, the farther away they appear to be. Therefore, depending on the vanishing point, different depth gradient planes can be generated with the vanishing point being at the farthest distance.

In this thesis, we extract the vanishing lines using edge detection and Hough transform [41]. For each frame of the 2D video, we detect edges using the Sobel operator to obtain an edge gradient map, which denotes the gradient vector of edge points that point in the direction normal to the line. This allows us to use the Hough transform to compute the ρ and θ parameters for edge points. The variable ρ is the distance from the origin to the line along a vector perpendicular to the line, and θ is the angle between the x-axis and this vector. We then generate a parameter space matrix whose rows and columns correspond to these ρ and θ values.

The Hough transform is designed to detect lines using these parameters as follows [42]:

$$\rho = x * \cos(\theta) + y * \sin(\theta) \quad (8)$$

After finding the vanishing lines by computing the Hough transform, we find the peak values in the parameter space to help us detect the vanishing point. These peak values, which represent potential lines in the input image, return a matrix that contains the row and column coordinates of the Hough transform bins. Subsequently, we find the line segments corresponding to these peaks and determine the intersection points of these lines. The vanishing point is then chosen as the intersection point with the greatest number of intersections in its neighbourhood.

The linear perspective feature reflects the general tendency of the depth transition. This transition varies gradually in a scene; hence, we form a local hypothesis of depth variation. However, in order to distinguish discontinuity between regions, we segment each frame into multiple regions using graph-based image segmentation [43].

Based on the detected vanishing point, we form a local depth hypothesis to estimate the general depth transition for each segmented region [44]. This depth hypothesis is determined by the Euclidean distance from the vanishing pint [45] and relative height [46]. The Euclidean distance between a point and the vanishing point is an indicator of how far this point is from the viewer. In other words, the closer a point is to the vanishing point, the farther the point locates from the viewer. Thus, we assign a

lower depth value to that point. The Euclidean distance depth hypothesis $D_E(x, y)$ depends on the vanishing point and is generalized as:

$$D_E(x, y) = \sqrt{(x - x_{VP})^2 + (y - y_{VP})^2} / \max(D_E) \quad (9),$$

where (x_{VP}, y_{VP}) represents the position of the vanishing point.

In addition, the relative height represents the gradual variation in depth because the lower part of an image is usually closer to the viewer. This relative height depth hypothesis $D_R(x, y)$ is estimated based on the y coordinate alone since it does not depend on the vanishing point:

$$D_R(x, y) = \frac{H - y}{H} \quad (10),$$

where H is the height of the input image. This hypothesis reflects the gradual variation in depth with the y coordinate value.

Finally, we estimate the depth as a weighted combination of the abovementioned depth hypotheses for each segmented region *k*:

$$D^k_{Linear\_perspective} = w^K_E D^k_E(x, y) + w^K_R D^k_R(x, y) \quad (11),$$

where $w_E + w_R = 1$, w_E and w_R denote weight coefficients for the two hypotheses: Euclidean distance and relative height, respectively. The higher the y-coordinate of the vanishing point, the greater the value of weight w_R assigned to the relative height depth

hypothesis. This is due to the fact that when the vanishing point is located at the upper part of the image, the lower part would be closer to the viewer.

The lower part of the image would be farther away from the viewer, if the detected vanishing point is located lower along the negative y coordinate. Therefore, the Euclidean distance cue would have higher contribution. In case, the vanishing point y coordinate is detected within the range of the image's height, then the depth hypothesis is determined from both Euclidean distance and relative height. On the other hand, if the detected vanishing point is farther along the positive *y* coordinate and is located far outside the image, this means the upper part of the image is farther away from the viewer. Thus, the contribution of the relative height depth cue is higher. Consequently, the weight for the relative height hypothesis is formed depending on where the vanishing point is as follows:

$$w_R = \begin{cases} 0, & if\ y_{VP} \leq 0 \\ \frac{y_{VP}}{2H}, & if\ 0 < y_{VP} < H \\ \min\left\{\frac{y_{VP}}{2H}, 1\right\}, & otherwise \end{cases} \quad (12),$$

and

$$w_E = 1 - w_R \quad (13)$$

If the segmented region does not have a detected vanishing point, the depth is determined by the relative depth hypothesis only:

$$D^k = D_R^k(x, y) \quad (14)$$

3) Occlusion

Occlusion detection plays an important role in depth estimation. Although it does not provide sense of depth magnitude, it gives important information about depth order: an object that occludes another is closer.

Occlusion is detected when one part of the scene is visible in one frame, and not in preceding/succeeding frames. The portions of the scene that are not visible temporally, can be mapped onto one another by a domain deformation, called optical flow. Optical flow refers to the deformation of the domain of an image that results from camera or scene motion. It is, in general, the distribution of objects' apparent velocities caused by the relative motion between the camera and the scene.

In this thesis, we adopt the work presented in [47] to detect occlusion using optical flow under the assumptions of (a) Lambertian reflection ("It is the property that defines an ideal reflecting surface, where the apparent brightness of a Lambertian surface is the same regardless of the observer's angle of view" [48]) and (b) constant illumination (The scene is moving relative to a light source). Most surfaces with reflectance properties (diffuse/specular) can be approximated as Lambertian almost everywhere under sparse illuminates, thus, this assumption is embraced as customary. Similarly, constant illumination is a reasonable assumption for camera-motion and even for objects moving slowly relative to the light source. Under the assumptions (a) and (b) the occlusion detection and optical flow estimation task can be posed as a variational optimization problem, and its solution may be approximated using convex minimization.

The resulting optimization problem is solved jointly with respect to the unknown optical flow field, and the indicator function of the occluded region.

We implement the work presented in [47]. First, we do domain deformation mapping of each frame (image) $I(x,t)$ onto $I(x,t+dt)$ everywhere except at occluded regions as follow:

$$w(x,t) = x + v(x,t) \tag{15}$$

where $v(x,t)$ denotes the optical flow. Under the abovementioned assumptions, the relation between two consecutive frames in a video sequence $\{I(x,t)\}_{t=0}^{T}$, is then written as:

$$I(x,t) = \begin{cases} \mathrm{I(w(x,t),t\ +\ dt) + n(x,t),} & \mathrm{x\ \in\ D\backslash\Omega(t;\ dt)} \\ \mathrm{\rho(x,t),} & \mathrm{x\ \in\ \Omega(t;\ dt)} \end{cases} \tag{16}$$

where the occluded region Ω can change overtime on the temporal sampling interval dt. Inside the occluded region Ω, the image can take any ρ value that is in general unrelated to $\mathrm{I(w(x,t),t\ +\ dt)}$.

Then, we define the residual *e* on the entire image domain:

$$e(x,t;\ dt)\ =\ I(x,t)\ -\ I(w(x,t),t\ +\ dt) \tag{17}$$

which can be written as the sum of two terms, e_1 and e_2:

$$I(x,t)\ =\ I(w(x,t),t\ +\ dt)\ +e_1\ (x,t;\ dt)\ +e_2\ (x,t;\ dt) \tag{18}$$

Note that e_2 is undefined in Ω, and e_1 is undefined in $D\backslash\Omega$, in the sense that they can take any value there, including zero.

Next, we define the data term depending on the optical flow and the residual. This is due to the fact that we do not know anything about the residual e_1 except the fact that it is sparse and that what we are looking for is where the residual is non-zero, because these non-zero elements indicate the occluded region at each pixel in the image. Therefore, the data term can be written as:

$$\psi_{data}(v, e_1) = \|\nabla I_v + I_t - e_1\|_{\ell^2} + \lambda\|e_1\|_{\ell^0} \tag{19}$$

where λ is a tuning parameter. Further, we impose regularization, because the problem is ill posed, by requiring that the total variation *TV* to be small:

$$\psi_{req}(v) = \mu\|v_1\|_{TV} + \mu\|v_2\|_{TV} \tag{20}$$

where v_1 and v_2 are the first and second components of the optical flow v, and μ is a multiplier factor to weight the strength of the regularizer. TV is desirable in the context of occlusion detection because it does not penalize motion discontinuities significantly. The overall problem can then be written as the minimization of the cost functional

$$\psi = \psi_{data} + \psi_{req}, \tag{21}$$

which is a model that formulates occlusion detection and optical flow estimation that is represented as a joint minimization problem under sparsity prior on the occluded regions [47]:

$$\hat{v}_1, \hat{v}_2, \hat{e}_1 = \arg\min_{v_1,v_2,e_1} \frac{1}{2}\|A[v_1, v_2, e_1]^T + b\|_{\ell_2}^2 + \lambda\|We_1\|_{\ell_0} + \mu\|v_1\|_{TV} + \mu\|v_2\|_{TV} \quad (22)$$

where the vector field components $v_1(x,t)$ and $v_2(x,t)$ stacked into MN-dimensional vectors v_1, v_2. The occluded region, which is represented by e_1, is the vector obtained from stacking the values of $e_1(x,t)$ on top of one another. A is the spatial derivative matrix, the temporal derivative $I_t(x,t)$ values are stacked into b, and W is a diagonal matrix. The error term e_1, which represents the occluded region, is the spatial difference between two frames.

Since this problem is NP-hard when solved with respect to the variable e_1. A relaxation into a convex would simply replace the ℓ_0 norm with ℓ_1. However, this would imply that "bright" occluded regions are penalized more than "dim" ones, which is clearly not desirable. Therefore, they relax the ℓ_0 norm with the weighted-ℓ_1 norm.

$$\hat{v}_1, \hat{v}_2, \hat{e}_1 = \arg\min_{v_1,v_2,e_1} \frac{1}{2}\|A[v_1, v_2, e_1]^T + b\|_{\ell_2}^2 + \lambda\|We_1\|_{\ell_1} + \mu\|v_1\|_{TV} + \mu\|v_2\|_{TV} \quad (23)$$

As suggested by [47], Nesterov's first order scheme [49] is used to solve this problem. This algorithm provides $O(1/k^2)$ convergence in k iterations, which is a considerable advantage for a large-scale problem such as (14).

We start from the standard ℓ_1 relaxation of ℓ_0 norm on the error term e_1 and then reweight the ℓ_1 norm iteratively, thus improving sparsity and achieving a better approximation of ℓ_0 norm. The error term contains the difference between two frames in the spatial domain (pixels). Thus, it can be used as occlusion indicator since it identifies the pixels of the first frame that become occluded in the second one.

We then perform segmentation on each frame, and analyze the pixels in each segment to check if the occluded pixels belong to it. Subsequently, we assign depth order using occlusion information based on scene segmentation.

4) Motion Parllax

Motion parallax is another important cue to perceive depth information. This is based on the fact that objects with different motions usually have different depths, such that, close objects usually seem to move faster than farther objects.

In our implementation, we estimate motion using a fast block-matching algorithm called *Adaptive Rood Pattern Search* (ARPS) [50]. Motion estimation using block-matching algorithms demand minimal computation and are fast, which make them suitable for real-time applications. The idea behind block matching is to divide the current frame of a video sequence into a matrix of *macroblocks* (MBs) that are then compared with the corresponding block and its adjacent neighbours in the previous frame to create a motion vector (MV) that represents the movement of a MB in current frame from one location to another with respect to the previous frame [6]. This movement is calculated for all the macroblocks in a frame in order to estimate the motion in the current frame. The search area for a good MB match is constrained to a search range parameter (p). The search range is basically is the area with a distance of p pixels from all four sides of the corresponding macroblock in the previous frame.

The ARPS block-matching algorithm is based on the assumption that the general motion in a frame is usually coherent. In other words, if the macroblocks around the

current macroblock have moved in a particular direction, then there is a high chance that the current macroblock also has a similar motion vector. This algorithm consists of two sequential search stages: 1) initial search and 2) refined local search. For each MB, the initial search is performed only once at the beginning in order to find a good starting point for the follow-up refined local search. The ARPS step size (between the rood pattern distributed points) is dynamically determined for each MB, based on the available motion vectors (MVs) of the neighbouring MBs. In the refined local search stage, a fixed-size search pattern is exploited repeatedly, and unrestrictedly, until the final MV is found.

In this thesis, we chose the macroblock's size to be 16x16, and the search parameter p to be 7 pixels. The MB that results in the least cost, is the one that matches the most to the current block. Here we use the Mean Absolute Error (MAE) cost functions to measure the matching criteria [6]:

$$MAE = \frac{1}{B^2}\sum_{i=0}^{B-1}\sum_{j=0}^{B-1}\left|C_{ij} - R_{ij}\right| \quad (24)$$

where B^2 is the size of the macroblock, C_{ij} and R_{ij} are the pixels, being compared in the current and reference macroblocks, respectively.

In our implementation, we use a scaled linear mapping to calculate depth values from motion vectors [51]:

$$D(i,j)_{motion} = c\sqrt{MV(i,j)_x{}^2 + MV(i,j)_y{}^2} \quad (25)$$

where $D(i,j)$ is the depth value for pixel (i,j), c is a pre-defined constant, and $MV(i,j)_x$, $MV(i,j)_y$ are the horizontal and vertical components of the motion vector for that pixel, respectively.

The estimated depth map form motion parallax is block-based. In order to reduce the blocking effect, we perform depth fusion of the block-based depth map and color-based image segmentation for each frame. First, we apply color-based segmentation using K-means clustering [52] to the original frames to identify objects. To separate these objects, we define a global histogram threshold using Otsu's method [53]. This method computes a global threshold *T* from histogram counts, where *T* is a normalized intensity value between 0 and 1. Then, we assign new depth values to these objects by using the average of the depth values from the motion estimation depth map in the area of the corresponding segment.

3.3.2 Outdoor Scenes

For outdoor scenes, we estimate the depth map using three depth cues: motion parallax, blur, haze, and vertical edges. The process for extracting each of these features is detailed below.

1) Motion Parallax

The motion parallax estimation technique is described hereinbefore under the indoor scenes section.

2) Blur

The blur estimation technique is described hereinbefore under the indoor scenes section.

3) Haze

Haze is an atmospheric diffusion of light that results in contrast loss effect in images. It can be usually observed in outdoors images, where objects in the far distance experience more haze than objects that are in close range.

The extent of haze is reflected in dark channel values, such that smaller dark channel values mean smaller amounts of haze and smaller depth. Therefore, for each 4x4 pixel block, we determine the minimum value for the red, green, and blue channels. This results in three values, where each value is the minimum for each channel within the block. From these three values, we select the minimum one that will represent the dark channel value of the block. This haze extraction approach is defined as follows [54]:

$$I_{dark}(x,y) = \min_{k\in(r,g,b)}\left(\min_{(x,y)\in\Omega} I_k(x,y)\right) \quad (26),$$

where Ω denotes the 4x4 pixel block and $I_k(x,y)$ represents the minimum values for each spatial location in the red, green, and blue channels within the pixel block.

The depth map is then calculated based on the dark channel prior:

$$D(x,y)_{Haze} = 1 - c(I_{dark}(x,y)) \quad (27),$$

where c is a constant parameter for adjusting the amount of haze for distant objects. In this thesis, the value of c is set to be 0.95.

4) Vertical Edges

For the same vertical coordinate, objects with larger vertical edges appear closer compared to the objects with smaller (or without) vertical edges. According to [27], the presence of vertical edges indicates the presence of vertical objects. To detect the vertical edges within each frame, we find the horizontal gradient value for each block. This is achieved by performing 2D convolution with a 4x4 mask that is defined as follows:

$$\bar{v}(x,y) = \sum_{(x,y)\in\Omega} \frac{v(x,y)}{M} \quad (28),$$

where Ω represents the 4x4 block, M is the number of pixels in that block, and $v(x,y)$ denotes the horizontal gradient value.

To identify objects with larger vertical edges, we apply color-based segmentation using k-means clustering [52] to the original 2D frame. Then, we assign low depth values to objects with larger vertical edges, and high depth values to objects with smaller vertical edges.

3.4 Depth Map Generation

We estimate our depth map in two steps. First, we estimate depth values at edges. Then, we propagate these depth values to the rest of the image. Our assumption is that depth varies smoothly over the image except in the areas with significant frequency content where change in depth reaches its maximum value. Therefore, we compute depth values at the edges to get reliable depth estimation in those areas.

In previous subsections, different features were obtained to generate four and three depth maps for indoor and outdoor scenes, respectively. In our work, we use linear depth fusion to estimate the combined depth values at the edges. For indoor scenes, the depth values are computed as follows:

$$D_{indoor} = \omega_B . D_B + \omega_L . D_L + \omega_O . D_o + \omega_M . D_M \quad (29),$$

with:

$$\omega_B + \omega_L + \omega_O + \omega_M = 1 \quad (30),$$

where ω_B, ω_L, ω_O, and ω_M are pre-determined weighting parameters of blur, linear perspective, occlusion, and motion parallax, respectively. These weighting parameters are determined empirically. Likewise, the combined depth values for outdoor scenes are calculated:

$$D_{outdoor} = \omega_B . D_B + \omega_H . D_H + \omega_V . D_V \quad (31),$$

where ω_B, ω_H, and ω_V are weighting parameters for blur, haze, and vertical edges, respectively.

In our implementation, we detect edges using the canny edge detector [55]. Next, we estimate depth values for edge pixels using the combined depth values. This process provides us with a sparse set of depth values at the edge locations of the frame. However, noise and weak edges may cause inaccurate depth estimates at some edge locations. To solve this problem, we apply joint bilateral filtering [30] on the sparse depth map to supress the influence of these outliers and correct the inaccurate depth values using their adjacent depth values along the edge .

In order to obtain the full depth map, we need to find a way to propagate the sparse depth map to the entire frame. Edge-aware interpolation method introduced in [56] has been used for similar purpose in [23], [54], [57]. Therefore, we formulate an optimization problem to find the full depth map that is close to the sparse map as follows:

$$(L + \lambda D_s)\alpha = \lambda s, \tag{32}$$

where α denotes the full depth map, s is the sparse map, λ is a scalar that balances between the sharpness of the sparse depth map and the smoothness of the full depth map, D_s represents a diagonal matrix whose diagonal element is 1 for sparse pixels and 0 otherwise, and L is a matting Laplacian matrix [56]. To solve the optimization problem in Eq. (23) and find the full depth map, we consider the depth map to be the global optimum of the following cost function:

$$J(\alpha) = \alpha^T L\alpha + \lambda(\alpha - s)(\alpha - s) \quad (33).$$

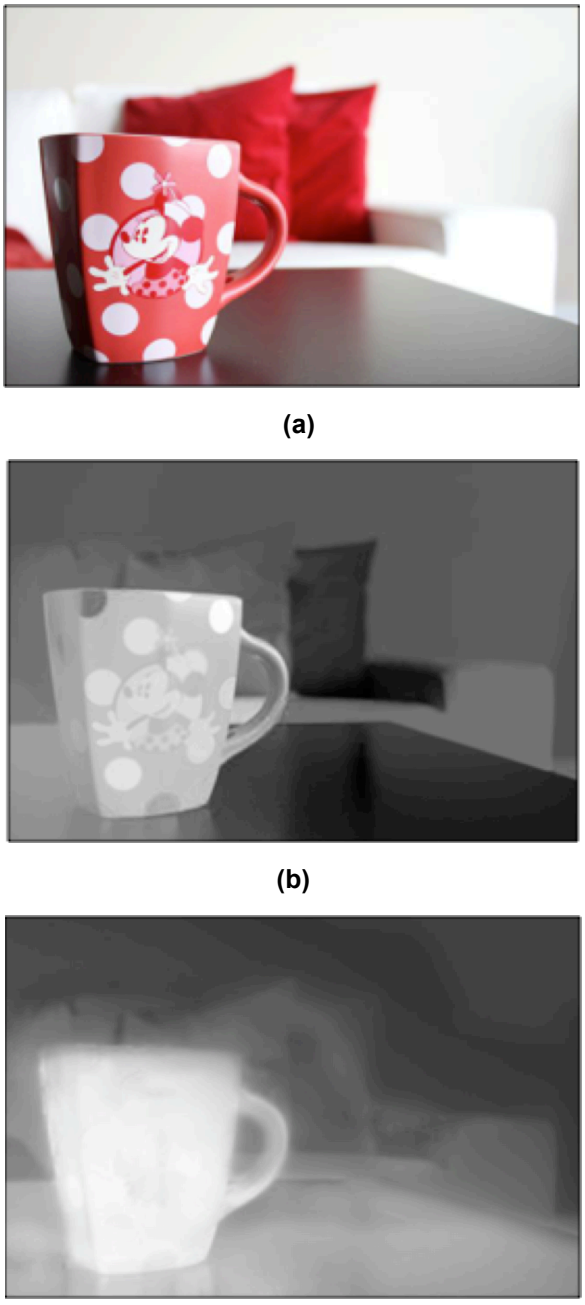
Minimizing this cost function allows us to find the globally optimal estimate for full depth map based on the sparse depth map.

Considering the dimensionality and computational cost of solving this linear system, we down-sample each frame by a factor of 4 as a pre-processing step and then up-sample it by 4 after applying the Edge-aware interpolation. Our tests have shown that this does not affect the results.

As noted above, the depth map for each depth cue is estimated by referring to only one frame or two consecutive frames. This frame-by-frame estimation has the disadvantage of unstable depths, especially when the depth cue fluctuates due to varying lighting. In addition, large depth variations may result in a discontinuity or bending of the object contours after image rendering. To address these issues, we apply a temporal median filter [31] to the full depth map before it is used for image rendering.

Chapter 4: Evaluation

4.1 Result



(a)

(b)

(c)

Figure 4.1. Experimental results of an indoor scene, (a) original image, (b) depth map generated by defocus, (c) estimated depth map by the proposed scheme.

For evaluating the performance of our method we compared it with the defocus depth estimation method [23], since the latter also uses image matting for depth map generation. Figures 4.1 (a), (b) and (c) show an original image, the depth map generated by the defocus method, and the depth map estimated by the proposed scheme, respectively. We observe that the depth order in the depth map generated by our approach is correct, as the coffee table appears to be closer to the viewer than the couch and the wall, while in the defocus results, the depth values seem to be assigned incorrectly.

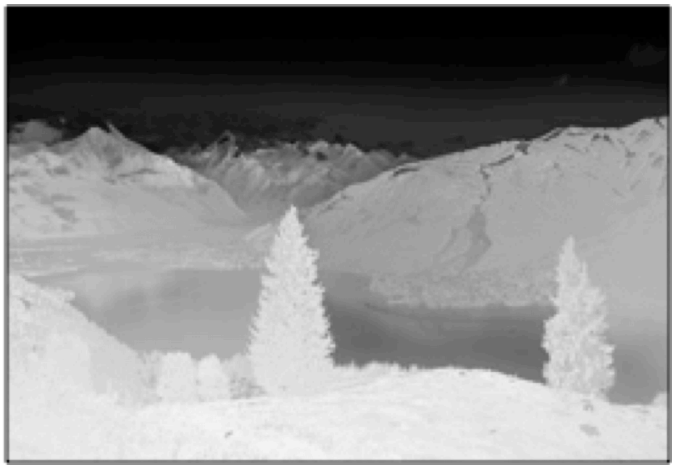
Fig. 4.2 (b) shows the depth map generated by the defocus method while Fig. 4.2 (c) shows the depth map generated by our proposed approach for the same original scene. We can observe that the defocus method wrongly assigns the region of the depth map enclosed by the red rectangle as part of the background and not the flowers. Overall, the depth map created by our approach produces more accurate results.

(a) (b) (c)

Figure 4.2 Comparison of our method with defocus estimation method, (a) original 2D scene, (b) depth map estimated by defocus method, (c) depth map estimated by proposed scheme

(a)

(b)



(c)

Figure 4.3 Experimental results of a natural image, (a) original image, (b) depth map generated by defocus, (c) estimated depth map by proposed scheme.

Furthermore, Fig. 4.3 demonstrates a natural image and the estimated depth map using our scheme and the depth map estimated based on the defocus method. We notice that the depth map estimated by our method successfully captures the continuous change of depth from the bottom to the top of the image, while the depth levels are not differentiated accurately in the one generated by defocus estimation where further objects have the same depth level as the close ones.

Fig. 4.4 and Fig. 4.5 show a snapshot of the original indoor and outdoor 2D videos, respectively. They also show the provided ground-truth depth maps, the depth maps estimated by our proposed approach, and the ones generated by [23]. As it can be observed, the depth maps generated by our method provide much more details and information compared to the ones produced by the method proposed in [23]. Table 1 and Table 2 present the weighting parameters for each depth cue to generate the depth maps by our approach.

In the balloon snapshot, Fig. 4.4 (a), the scene mainly encloses a number of balloons on the floor with different depth order. Our method captures this difference in the depth order (Fig.4.4 (d)), better than the defocus approach (Fig. 4.4 (c)). This is because in our proposed scheme, we try to adopt human visual perceptions (motion information, occlusion, blur, and linear perspective) to estimate the depth map.

Fig. 4.4 (e) shows an image of the Poznan Hall, the depth map generated by our scheme in Fig. 4.4 (h) is similar to the one generated by the defocus (Fig. 4.4 (g)) to some extent, where both schemes capture more details than the original depth map Fig. 4.4 (f). However, our depth map is slightly better than the defocus one. We can notice

that the walls behind the stairs are supposed to be farther away from the viewer, hence, they should have higher depth values and this is the case in our estimated depth map.

For the Kendo sequence shown in Fig. 4.4 (i), we can observe that our depth map (Fig. 4.4 (l)) mimics the original one shown in Fig. 4.4 (j), yet it captures more details. The depth map produced by the defocus method Fig. 4.4 (k) fails to assign the correct depth values for the smoke.

Furthermore, the depth map estimated by our proposed method for the newspaper scene and shown in Fig. 4.4 (p) surpasses the one produced by the defocus Fig. 4.4 (o).

Indoor Sequence	Weight of Blur (ω_B)	Weight of Linear Perspective (ω_L)	Weight of Occlusion (ω_O)	Weight of Motion (ω_M)
Balloons	0.15	0.10	0.45	0.30
Poznan Hall	0.10	0.20	0.35	0.35
Kendo	0.25	0.15	0.30	0.30
Newspaper	0.20	0.15	0.40	0.25

Table 4.1 Weighting parameters for depth maps of indoor videos

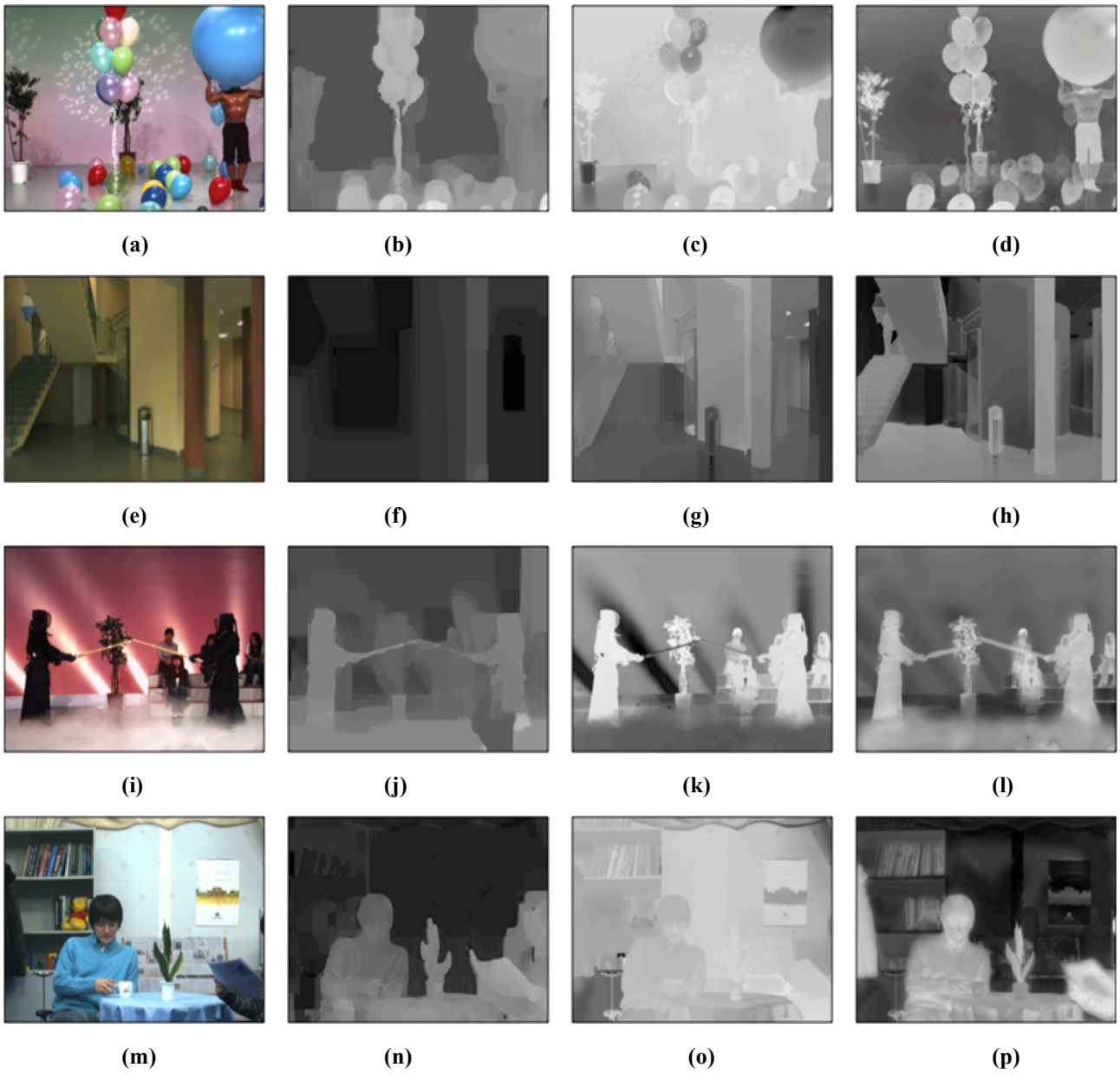


Figure 4.4. (a, e, i, m) original 2D videos, (b, f, j, n) ground-truth depth maps, (c, g, k, o) depth maps generated by defocus method, (d, h, l, p) estimated depth maps by our approach

In the ghost town fly image shown in Fig. 4.5 (a), the scene mainly contains a town in a desert where buildings are present at the top of the image. As shown in Fig. 4.5 (d), our approach is able to produce very realistic depth information compared to the

original depth map in Fig. 4.5 (b), where the buildings and the cacti details are accurately captured.

In Fig. 4.5 (e), the Poznan street scene includes cars and buildings. The depth map estimated by our scheme (Fig. 4.5 (h)) captures more details than the original depth map (Fig. 4.5 (f)), which only captures the presence of the cars. We can also notice that the depth map produced by the defocus method in (Fig. 4.5 (g)) has wrong depth values. For the Bee sequence, we observe, once more, that our method is able to very accurately generate the depth map.

Outdoor Sequence	Weight of Blur (ω_B)	Weight of Haze (ω_L)	Weight of Vertical Edges (ω_O)
Ghost Town Fly	0.25	0.45	0.30
Poznan Street	0.35	0.20	0.45
Bee	0.50	0.20	0.30

Table 4.2 Weighting parameters for depth maps of outdoor videos

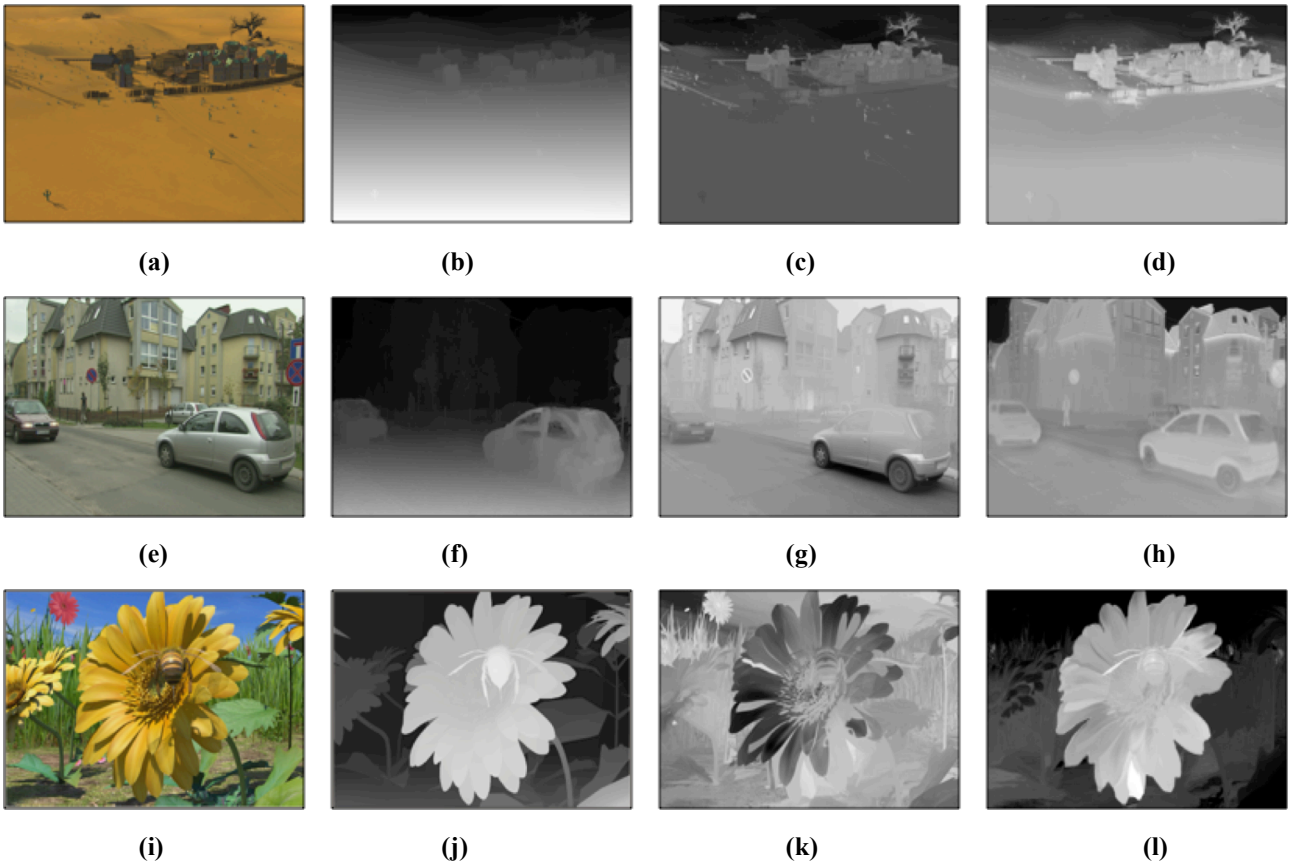


Figure 4.5 (a, e, i) original 2D videos, (b, f, i) ground-truth depth maps, (c, g, k) depth maps generated by defocus method, (d, h, l) estimated depth maps by our approach.

4.2 Quantitative Evaluation

4.2.1 Subjective Evaluation Setup

We subjectively evaluate the performance of our method using the test sequences provided to MPEG for 3D CfP [37] and FTV CfE [38]. For indoor scenes, we used the sequences presented in Table 3. For outdoor scenes, we use the sequences presented in Table 4. These sequences are provided with high quality ground-truth depth maps. Only one view (the right view) within each multi-view video was used: view 3 of the Kendo, and the Poznan Street, view 1 of the Balloons, the Newspaper, and the Ghost Town Fly, and view 5 of the Poznan Hall. The left view for each sequence was synthesized using the existing VSRS package (version 3.5) [58].

Test Sequence	Size	Frame Rate	Number of frames
Balloons	1024x768	25 fps	250
Poznan Hall	1920x1088	25 fps	200
Kendo	1024x768	25 fps	250
Newspaper	1024x768	25 fps	250

Table 4.3 Indoor scenes video sequences

Test Sequence	Size	Frame Rate	Number of frames
Poznan Street	1920x1088	25 fps	250
Ghost Town Fly	1920x1088	25 fps	250
Bee	1920x1088	25 fps	240

Table 4.4 Outdoor scenes video sequences

We subjectively evaluate the visual quality of the 3D video streams generated by our method against the defocus approach and the existing state-of-the-art automatic 2D-to-3D conversion of a commercial LG 3DTV. This evaluation is done according to the Recommendation ITU-R BT.500-13 standard using a single stimulus method for the test sessions and an “adjectival categorical judgment method” for the ratings [59]. Each test session was designed according to the single stimulus method, such that the videos of the same scene generated based on different schemes were shown to the observers in a random order, and the subjects were rating each and every video. There were five discrete quality levels (1-5: bad, poor, fair, good, excellent) for rating the videos, where score 5 indicated the highest quality and score 1 indicated the lowest quality. In particular, subjects were asked to rate a combination of “natural-ness”, “depth impression” and “comfort”.

	Number of Subjects	Male	Female	Age range	Outliers
Indoor Sequences	18	10	8	19 – 34 years old	2
Outdoor Sequences	18	15	3	24 – 36 yeas old	1

Table 4.5 Details about the participants in our subjective evaluation for the 3D video quality of the indoor and outdoor categories.

As shown in table 5, for indoor sequences, eighteen adult subjects including 10 males and 8 females participated in our experiment. The subjects' age range was from 19 to 34 years old. While for outdoor sequences, eighteen adult subjects including 15 males and 3 females participated in our experiment. The subjects' age range was from 24 to 36 years old.

Prior to the tests, all the subjects were screened for color blindness using the Ishihara chart, visual acuity using the Snellen charts, and stereo vision using Randot test (graded circle test 100 seconds of arc). Those subjects that failed the pre-screening test did not participate in the test.

The evaluation was performed using passive glasses on a 55" LG OLED smart TV with cinema 3D of 1920x1080 resolution. After collecting the subjective results, the outlier subjects were detected according to the ITU-R BT.500-13 recommendation in [59].

One outlier was detected for indoor sequences, and two outliers were detected for outdoor sequences. Those outliers were removed from the test results. The Mean Opinion Score (*MOS*) for each impaired video was calculated by averaging the scores over all the subjects with 95% confidence interval.

4.2.2 Discussion

Fig. 4.6 and Fig. 4.7 show the mean opinion scores for the test sequences. The black bar on each graph shows the 95% confidence interval viewers. As it can be observed in both figures, for all the sequences the scenes produced by our approach scored higher than those generated by defocus's and LG's approaches, confirming the superior performance of our technique.

In summary, our evaluations confirm the high performance of our proposed scheme for estimating depth map for scenic views. Based on our experiments, the proposed scheme is computationally inexpensive and has a great potential for real time applications.

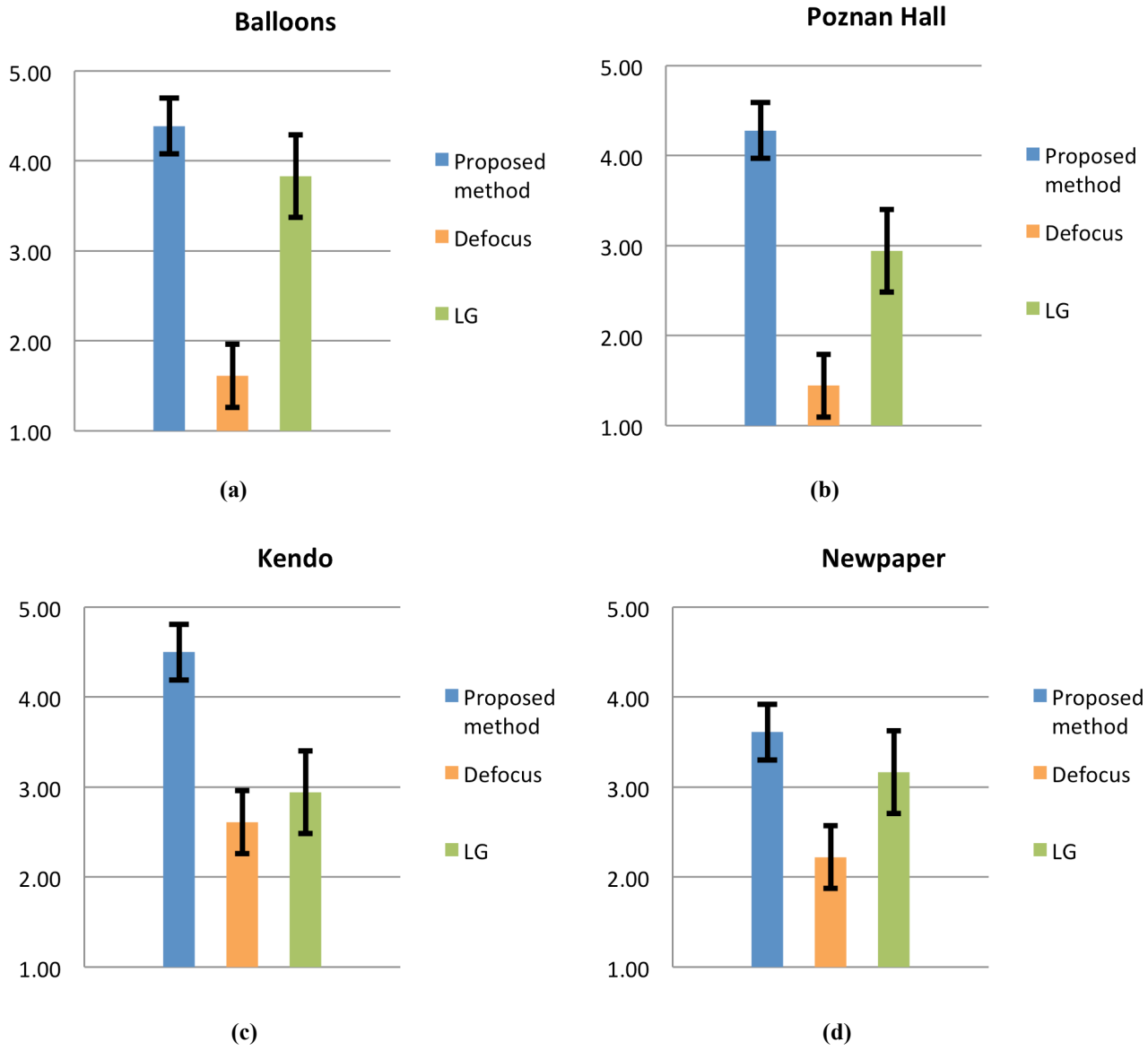


Figure 4.6 Mean opinion scores with 95% confidence interval for 3D visual quality of indoor scenes: (a) Balloons, (b) Poznan Hall, (c) Kendo and (d) Newspaper sequences generated using our proposed framework, defocus, and LG approaches

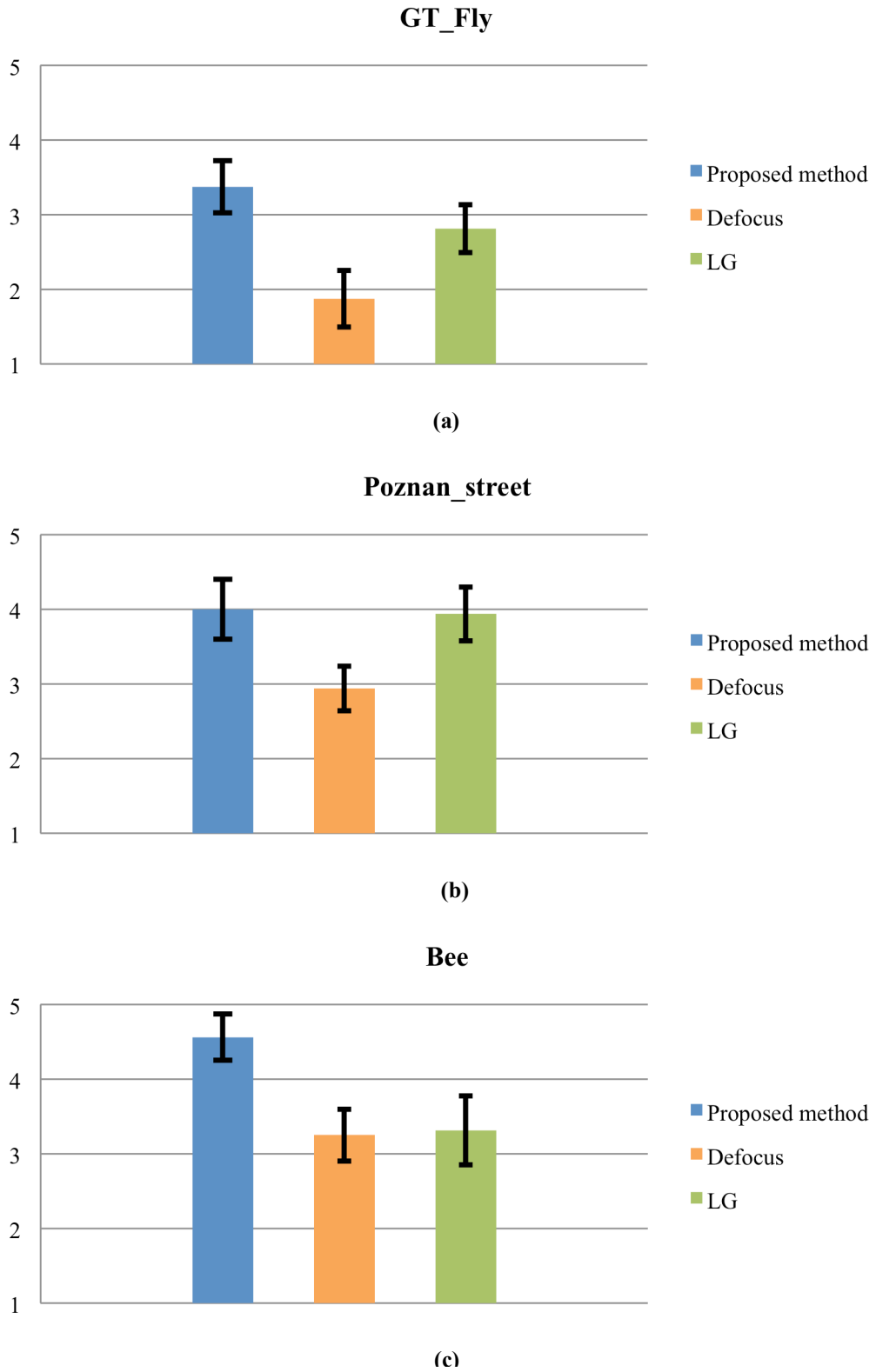


Figure 4.7 Mean opinion scores with 95% confidence interval for 3D visual quality of outdoor scenes: (a) Ghost town fly, (b) Poznan street, and (c) Bee sequences generated using our proposed framework, defocus, and LG approaches

Chapter 5: Conclusion and Future Work

5.1 Conclusion

This thesis presents a real-time 2D-to-3D conversion method that generates a high-quality depth map based on the video content. We classify the 2D videos into indoor or outdoor categories using SVM in accordance with the spatial envelope properties of the scene.

Based on the classification results, we estimate the depth map for each video frame using different depth cues. For indoor scenes we use four depth cues: 1) blur, 2) motion, 3) occlusion, and 4) linear perspective, while for outdoor scenes we use three depth cues: 1) blur, 2) haze, and 3) vertical edges.

Using these cues, we first estimate a sparse depth map for the edge-included regions of each frame (we use canny edge detector for identifying these regions). We then apply a joint bilateral filter on the sparse depth map to refine inaccurate depth estimations. Next, we propagate this sparse map to the entire frame in order to obtain the full depth map by solving an optimization problem. Finally, we apply a temporal median filter to the estimated depth map as a post-processing step to ensure temporal consistency and prevent flickering effect.

We compared the depth map created by our method to the original provided (ground-truth) depth maps showing that our results provide realistic depth information.

Further, our experimental results from subjective tests show that the proposed scheme outperforms the defocus estimation method and the existing state-of-the-art 2D-to-3D technique offered by the LG OLED displays.

5.2 Future work

The work in this field (i.e., depth map estimation) is not exclusive to 2D-to-3D conversion application. MPEG is currently exploring a new technology called FTV (Free-viewpoint Television) that enables the user to view a 3D world by freely changing the viewpoint. This technology, unlike regular 3D, exploits arbitrary camera arrangements including arc and 2D parallel settings. FTV applications include: Super Multi-view Video (SMV), Free Navigation (FN), and 360-degree 3D video. High quality depth data are essential in these applications.

For SMV applications, hundreds of dense views, which are configured in an arc (Figures 5.1 and 5.2) or linear (Figure 5.3) arrangements, are rendered to provide a very pleasant glasses-free 3D viewing experience with wide viewing angle, smooth transition between adjacent views, and some walk-around feeling on foreground objects. Therefore, a high quality depth map is desired for such application.

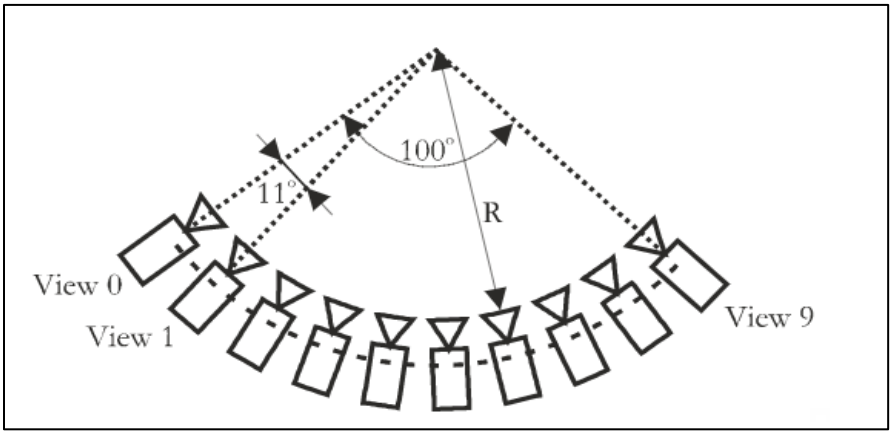


Figure 5.1 Arc Camera Arrangements for SMV [60]

Figure 5.2 Arc camera setup used in production of "Poznan Test" sequence [60]

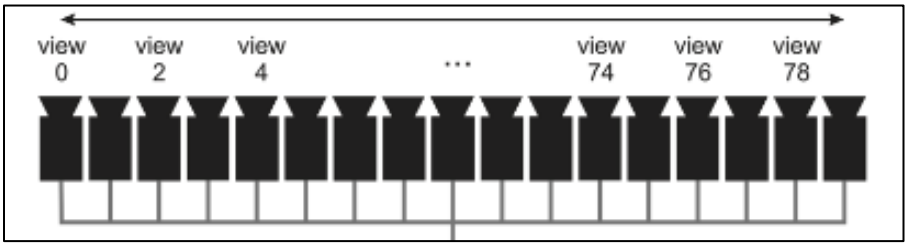


Figure 5.3 Linear Camera Arrangements for SMV [60]

With FN systems, the viewer virtually navigates in a scene by choosing by himself/herself the trajectory of virtual viewpoints. The source of data for FN is a sparse number of views (i.e., 6 – 10 views) with arbitrary positioning and wide baseline distance between each view, as shown in Figure 5.4. The input views require a high quality depth data in order to render the arbitrary virtual viewpoints.

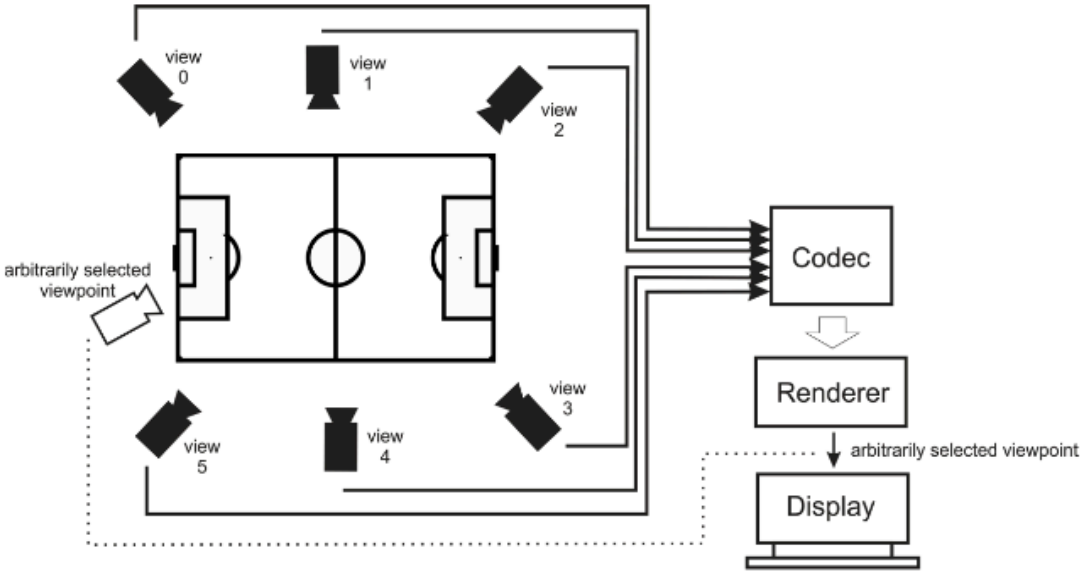


Figure 5.4 Linear Camera Arrangements for SMV [61]

Another FTV application is 360-degree video as shown in Figure 5.5. Such video covers Free Navigation requirements, not only for the 360-degree panoramic all-around viewing, but also for stereo parallax, i.e., when having a micro-movement of the head forward/backward or left/right around the center of the 360-degree circle, the occlusions/disocclusions around the object edges should change as one would experience in visualizing the real world. Consequently, novel views have to be synthesized constantly from the original image and depth data that will be transmitted.

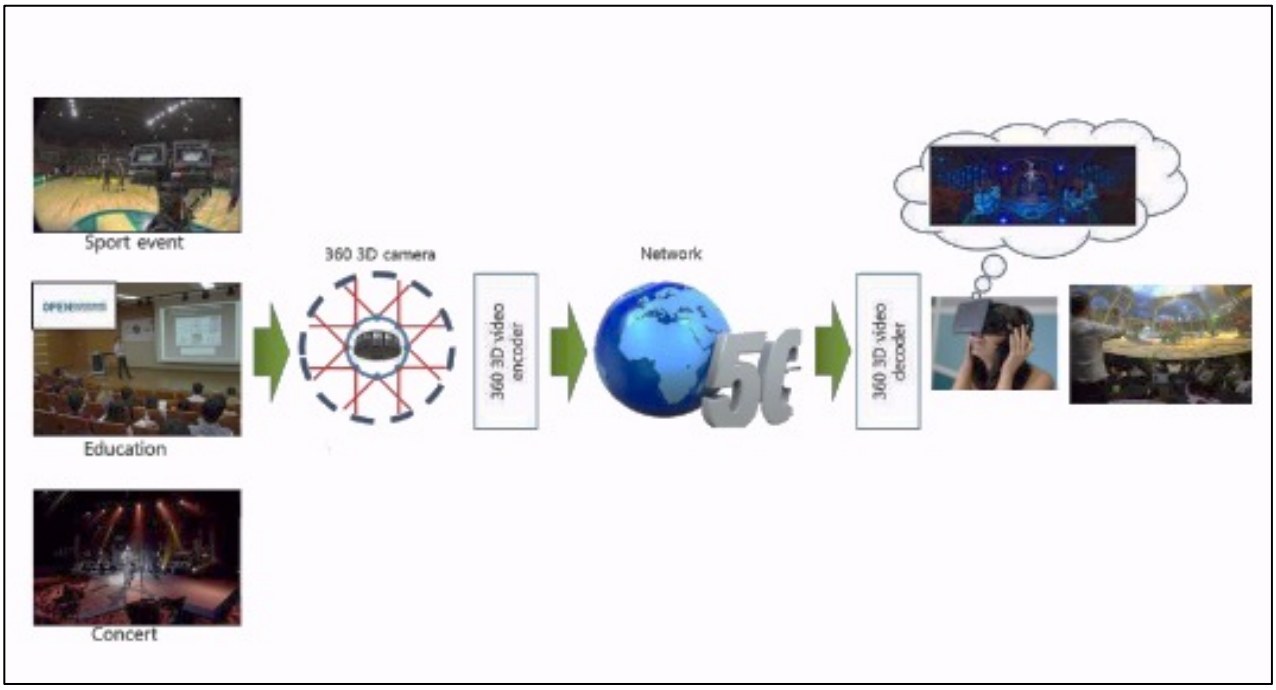


Figure 5.5 360-degree 3D application system [62]

In conclusion, the depth map estimation technique developed in this thesis could be the foundation of a more general approach that extends its use to consider dense and arbitrary camera arrangements.

Bibliography

[1] L. Zhang, C. Vázquez, and S. Knorr, “3D-TV content creation: Automatic 2D-to-3D video conversion,” *IEEE Trans. Broadcast.*, vol. 57, no. 2 PART 2, pp. 372–383, 2011.

[2] J. M. Dominic, “Recent Trends in 2D to 3D Conversion: A Survey,” *www.ijraset.com Issue IV*, vol. 2, 2014.

[3] M. T. Pourazad, P. Nasiopoulos, and A. Bashashati, “Random forests-based 2D-to-3D video conversion,” in *2010 IEEE International Conference on Electronics, Circuits, and Systems, ICECS 2010 - Proceedings*, 2010, pp. 150–153.

[4] K. Moustakas, D. Tzovaras, and M. G. Strintzis, “Stereoscopic video generation based on efficient layered structure and motion estimation from a monoscopic image sequence,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 8, pp. 1065–1073, 2005.

[5] Q. Wei, “Converting 2d to 3d: A survey,” *Int. Conf. Page*, 2005.

[6] A. Barjatya, “Block Matching Algorithms For Motion Estimation,” *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 225–229, 2004.

[7] Y. Feng, J. Ren, and J. Jiang, “Object-based 2D-to-3D video conversion for effective stereoscopic content generation in 3D-TV applications,” *IEEE Trans. Broadcast.*, vol. 57, no. 2 PART 2, pp. 500–509, 2011.

[8] K. Shimono, W. J. Tam, and S. Nakamizo, “Wheatstone-Panum limiting case: occlusion, camouflage, and vergence-induced disparity cues.,” *Percept. Psychophys.*, vol. 61, no. 3, pp. 445–455, 1999.

[9] S. Battiato, A. Capra, S. Curti, and M. La Cascia, “3D stereoscopic image pairs by depth-map generation,” in *Proceedings - 2nd International Symposium on 3D Data Processing, Visualization, and Transmission. 3DPVT 2004*, 2004, pp. 124–131.

[10] C. Yu-Lin, C. Wei-Yin, C. Jing-Ying, T. Yi-Min, L. Chia-Lin, and C. Liang-Gee, “Priority depth fusion for the 2D to 3D conversion system,” in *Three-Dimensional Image Capture and Applications*, 2008, vol. 6805, pp. 1–8.

[11] C. Zhu, Y. Zhao, L. Yu, and M. Tanimoto, *3D-TV System with Depth-Image-Based Rendering*. New York, NY: Springer New York, 2013.

[12] D. Scharstein and R. Szeliski, “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms,” *Int. J. Comput. Vis.*, 2002.

[13] C. Fehn, R. D. La Barré, and S. Pastoor, “Interactive 3-DTV-concepts and key technologies,” *Proc. IEEE*, 2006.

[14] P. Harman, J. Flack, S. Fox, and M. Dowely, “Rapid 2D to 3D Conversion,” *Int. Soc. Opt. Photonics*, vol. 4660, pp. 25–27, 2002.

[15] R. Phan and R. Rzeszutek, “Semi-automatic 2D to 3D image conversion using scale-space random walks and a graph cuts based depth prior,” *Image Process. (ICIP),* 2011.

[16] P. Harman, “Home based 3D entertainment-an overview,” in *Proceedings 2000 International Conference on Image Processing (Cat. No.00CH37101)*, 2000, vol. 1, pp. 1–4.

[17] C. Tomasi and T. Kanade, “Shape and motion from image streams: a factorization method.,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 90, no. 21, pp. 9795–9802, 1993.

[18] Q. Deng, Y. Zhang, and S. Li, “The overview of 2D to 3D conversion,” *ICEIEC 2013 - Proc. 2013 IEEE 4th Int. Conf. Electron. Inf. Emerg. Commun.*, pp. 226–229, 2013.

[19] D. Kim, D. Min, and K. Sohn, “A stereoscopic video generation method using stereoscopic display characterization and motion analysis,” *IEEE Trans. Broadcast.*, vol. 54, no. 2, pp. 188–197, 2008.

[20] L. Zhang, B. Lawrence, D. Wang, and A. Vincent, “Comparison study on feature matching and block matching for automatic 2D to 3D Video Conversion,” in *Visual Media Production, 2005. CVMP, The 2nd IEE European Conference on*, 2005, pp. 122–129.

[21] M. T. Pourazad, P. Nasiopoulos, and R. K. Ward, “Generating the depth map from the motion information of H.264-encoded 2D video sequence,” *Eurasip J. Image Video Process.*, vol. 2010, 2010.

[22] M. T. Pourazad, P. Nasiopoulos, and R. K. Ward, “Conversion of H.264-encoded 2D video to 3D format,” in *ICCE 2010 - 2010 Digest of Technical Papers International Conference on Consumer Electronics*, 2010, pp. 63–64.

[23] S. Zhuo and T. Sim, “Defocus map estimation from a single image,” in *Pattern Recognition*, 2011, vol. 44, no. 9, pp. 1852–1858.

[24] H. Sun, Z. Zhao, X. Jin, L. Niu, and L. Zhang, “Depth from defocus and blur for single image,” in *2013 Visual Communications and Image Processing (VCIP)*, 2013, pp. 1–5.

[25] J. Ens and P. Lawrence, “Investigation of methods for determining depth from focus,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 2, pp. 97–108, 1993.

[26] A. Saxena, S. H. Chung, and A. Y. Ng, “Learning Depth from Single Monocular Images,” *Adv. Neural Inf. Process. Syst.*, vol. 18, pp. 1161–1168, 2006.

[27] X. Wang, A. Berestov, J. Wei, and X. Tu, “2D to 3D image conversion based on image content,” U.S. Patent 8,520,935, 2013.

[28] H. R. Sanderson, S. R. Pegg, A. Baik, Y. J. Jung, J. W. Kim, and D. Park, “Method and apparatus for generating depth information of an image,” US20130235153A1, 2013.

[29] M. Jeong, N. Kumar, A. Sharma, A. MUSTAFA, K. Sehgal, K. NANJUNDAIYER, N. Krishnan, R. PAM, A. LADE, A. BANNE, B. NEYYAN, P. BHAGAVATHI, and R. THARAYIL, “Method and apparatus for converting 2d video to 3d video,” US 20140210944 A1.

[30] S. Paris and F. Durand, “A fast approximation of the bilateral filter using a signal processing approach,” *Int. J. Comput. Vis.*, vol. 81, no. 1, pp. 24–52, 2009.

[31] M.-H. Hung, J.-S. Pan, and C.-H. Hsieh, “A Fast Algorithm of Temporal Median Filter for Background Subtraction,” *Ubiquitous Int.*, vol. 5, no. 1, 2014.

[32] A. Oliva and A. Torralba, “Modeling the shape of the scene: A holistic representation of the spatial envelope,” *Int. J. Comput. Vis.*, 2001.

[33] S. Canu, Y. Grandvalet, V. Guigue, and A. Rakotomamonjy, “Svm and kernel methods matlab toolbox,” *Percept. Syst. Inf.*, no. INSA de Rouen, Rouen, France 2, no. 2, 2005.

[34] A. Torralba and A. Oliva, “Semantic organization of scenes using discriminant structural templates,” *Vision, 1999. Proc. …*, 1999.

[35] C. Hsu, C. Chang, and C. Lin, “A practical guide to support vector classification,” 2003.

[36] J. Xiao, J. Hays, K. Ehinger, and A. Oliva, “Sun database: Large-scale scene recognition from abbey to zoo,” *Comput. Vis.*, 2010.

[37] M. Domański, T. Grajek, K. Klimaszewski, M. Kurc, O. Stankiewicz, J. Stankowski, and K. Wegner, “Poznan Multiview Video Test Sequences and Camera Parameters,” *ISO/IEC JTC1/SC29/WG11 MPEG*, 2009.

[38] “National Institute of Information and Communications Technology.” [Online]. Available: http://www.nict.go.jp/. [Accessed: 20-Jan-2015].

[39] C. Swain, “Integration of monocular cues to create depth effect,” *Acoust. Speech, Signal Process. 1997.*, 1997.

[40] H. Sun, Z. Zhao, X. Jin, and L. Niu, “Depth from defocus and blur for single image,” *Image Process. (…*, 2013.

[41] X. Huang, L. Wang, J. Huang, D. Li, and M. Zhang, “A depth extraction method based on motion and geometry for 2D to 3D conversion,” *3rd Int. Symp. Intell. Inf. Technol. Appl. IITA 2009*, vol. 3, pp. 294–298, 2009.

[42] “Image Processing Toolbox: Houph Transform - User’s Guid (R2006a),” *The MathWorks, Inc.*, 2006. [Online]. Available: http://www.mathworks.com/. [Accessed: 01-Aug-2015].

[43] P. Felzenszwalb and D. Huttenlocher, “Efficient graph-based image segmentation,” *Int. J. Comput.*, 2004.

[44] N. Yang, J. W. Lee, and R. Park, “DEPTH MAP GENERATION USING LOCAL DEPTH HYPOTHESIS FOR 2D-TO-3D C ONVERSION,” vol. 3, no. 1, pp. 1–15, 2013.

[45] R. C. Gonzalez, R. E. Woods, L. Mcdowell, T. Galligan, and P. P. Hall, *Digital Image Processing*, Third edit. Upper Saddle River, Pearson Education, 2010.

[46] K. Han and K. Hong, “Geometric and texture cue based depth-map estimation for 2D to 3D image conversion,” *2011 IEEE Int. Conf. Consum.*, 2011.

[47] A. Ayvaci, M. Raptis, and S. Soatto, “Sparse occlusion detection with optical flow,” *Int. J. Comput. Vis.*, 2012.

[48] T. To, S. Nayar, R. Ramamoorthi, and P. Hanrahan, “Basic Principles of Surface Reflectance.”

[49] Y. Nesterov, “A method for unconstrained convex minimization problem with the rate of convergence O (1/k2),” *Dokl. an SSSR*, vol. 269, no. 3, pp. 543–547, 1983.

[50] Y. Nie and K. Ma, “Adaptive rood pattern search for fast block-matching motion estimation,” *Image Process. IEEE Trans.*, 2002.

[51] L.-M. Po, X. Xu, Y. Zhu, and S. Zhang, “Automatic 2D-to 3D Video Conversion technique based on depth-from motion and color segmentation,” *IEEE 10th Int. Conf. Signal Process.*, pp. 1–4, 2010.

[52] A. Z. Chitade and S. K. Katiyar, “Colour based image segmentation using k-means clustering,” *Int. J. Eng. Sci. Technol.*, 2010.

[53] N. Otsu, “A threshold selection method from gray-level histograms,” *Automatica*, 1975.

[54] K. He, J. Sun, and X. Tang, “Single image haze removal using dark channel prior,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 12, pp. 2341–2353, 2011.

[55] J. Canny, “A computational approach to edge detection.,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 6, pp. 679–698, 1986.

[56] A. Levin, D. Lischinski, and Y. Weiss, “A closed-form solution to natural image matting,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 228–242, 2008.

[57] E. Hsu, T. Mertens, S. Paris, S. Avidan, and F. Durand, “Light mixture estimation for spatially varying white balance,” *ACM Trans. Graph.*, vol. 27, no. 3, p. 1, 2008.

[58] “View Synthesis Software Manual, release 3.5.” MPEG, 2009.

[59] ITU-R Recommendation BT.500-13, “Methodology for the subjective assessment of the quality of television pictures.” 2012.

[60] M. Domański, A. Dziembowski, M. Kurc, A. Łuczak, D. Mieloch, J. Siast, and K. Wegner, “Poznan University of Technology test multiview video sequences acquired with circular camera arrangement–‘Poznan Team’ and ‘Poznan Blocks’ sequences.,” *ISO/IEC JTC1/SC29/WG11 MPEG2014 M 35846.*, 2014.

[61] G. Lafruit, K. Wegner, and M. Tanimoto, “raft Call for Evidence on FTV.,” *ISO/IEC JTC1/SC29/WG11 MPEG2015 N 15095.*, 2015.

[62] G. Bang, G. Lafruit*, G. soon Lee, and N. Ho Hur, “Introduction to 360 3D video application and requirements for FTV discussion,” *ISO/IEC JTC1/SC29/WG11 MPEG/M37351*, 2015.