# Analysis of MVD and color edge detection for depth maps enhacement

*Master Thesis*

by

**Jordi Bayo Singla**

| | |
|---|---|
| Start date: | 01. January 2010 |
| End date: | 23. June 2010 |
| Supervisor: | Dipl.-Ing. Philipp Merkle<br>Dr.-Ing. Karsten Müller |

# Abstract

MVD (Multiview Video plus Depth) data consists of two components: color video and depth maps sequences. Depth maps represent the spatial arrangement (or three dimensional geometry) of the scene. The MVD representation is used for rendering virtual views in FVV (Free Viewpoint Video) and for 3DTV (3-dimensional TeleVision) applications. Distortions of the silhouettes of objects in the depth maps are a problem when rendering a stereo video pair. This Master thesis presents a system to improve the depth component of MVD . For this purpose, it introduces a new method called correlation histograms for analyzing the two components of depth-enhanced 3D video representations with special emphasis on the improved depth component.

This document gives a description of this new method and presents an analysis of six different MVD data sets with different features. Moreover, a modular and flexible system for improving depth maps is introduced. The idea behind is to use the color video component for extracting edges of the scene and to re-shape the depth component according to the edge information. The mentioned system basically describes a framework. Hence, it is capable to admit changes on specific tasks if the concrete target is respected. After the improvement process, the MVD data is analyzed again via correlation histograms in order to obtain characteristics of the depth improvement.

The achieved results show that correlation histograms are a good method for analyzing the impact of processing MVD data. It is also confirmed that the presented system is modular and flexible, as it works with three different degrees of change, introducing modifications in depth maps, according to the input characteristics. Hence, this system can be used as a framework for depth map improvement. The results show that contours with 1-pixel width jittering in depth maps have been correctly re-shaped. Additionally, constant background and foreground areas of depth maps have also been improved according to the degree of change, attaining better results in terms of temporal consistency. However, future work can focus on unresolved problems, such as jittering with more than one pixel width or by making the system more dynamic.

# Acknowledgments

I am grateful to Philipp Merkle, Dr. Karsten Müller and Proffessor Dr. Thomas Wiegand from Heinrich Hertz Institute (HHI), whose experience oriented me from start to end of this project. I also would give thanks to Professor Ferran Marques, from Politechnical University of Barcelona, who contacted with Dr. Aljoscha Smolic from HHI, the person who aided me to do my master thesis at Heinrich Hertz Institute in Berlin.
I would like to thank the Interactive Visual Media Group of Microsoft Research (USA) for providing the Ballet and Breakdancers data sets, the KUK Filmproduktion GmbH (Germany) for providing the Horse data set, the ETRI (Korea) for providing the Newspaper data set, and Momentum DMT (Turkey) for providing the Synthetic data set.

Thanks to my parents and to the rest of my family for motivating me to have the Erasmus experience. And thanks to people in Berlin with who I shared unforgettable time.

Berlin, 22 June 2010

# Foreword

One of Avatar's curiosities: James Cameron, the film director, wrote the screenplay 15 years before filming. Nevertheless, he did not want to start with this huge project until the technology was totally powerful to develope it.

Berlin, 15.02.2010

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Despite 90% of the population profit from stereopsis, only two-dimensional signals have been historically represented by displays. Since the beginnings of the 20th century many attempts have been made to demonstrate the outstanding possibilities of stereopsis in several 3D displays, but these never achieved the same success than 2D screens. However some 3D productions attained a certain success [2]. Everyone agrees on the fact that *The Power of Love* -directed by Nat G. Deverich and Harry K. Fairall- was the first full-length 3D film shown to paying audience, presented at the Ambassador Hotel Theater in Los Angeles in 1922. Some years before, the Lumière brothers presented the first moving 3D pictures based on the Wheatstone stereoscope at the 1903's World Fair (an entertainment device of the end of the 19th century). To understand how they implemented the stereoscope, some principles of the human visual perception [3] need to be introduced first.

The binocular disparity, also called horizontal or retinal disparity, is the difference in image location of an object or a scene seen by both eyes, due to the horizontal distance between of left and right eye. In computer vision, disparity refers to the same difference but captured by two cameras instead of two eyes. Taking into account that the eye can be modeled as an optical sensor, the detection of the same object in different positions leads to the perception of depth. This human visual capability is known as stereovision and is presented graphically in figure 1.1.

Another visual phenomenon worth mentioning is the parallax. Regarding a scene consisting of an object and a background, the apparent displacement of this object viewed along two different perspectives (or lines of sight) is the aforesaid parallax. In other words, the object viewed from a perspective appears in front of a certain part of the background and viewed from the other perspective the same object appears located in front of a different part of the background. Parallax is measured by the angle (or the semi-angle) formed by these two lines of sight. Thus, closer objects have more parallax (bigger angle) than farther ones (smaller angle). Therefore, parallax is useful to assign relative distances between objects and background.

Joining both concepts, when having disparity with two images, objects that do not change their position while being seen from two different perspectives will be considered as background. Oppositely, those objects that will suffer a displacement relative to this background, will have, in their magnitude, different parallax. This effect will make the objects to stand out more or less, depending on their parallax . As an example, the object in figure 1.1 stands out in front of the white square by combining both views.

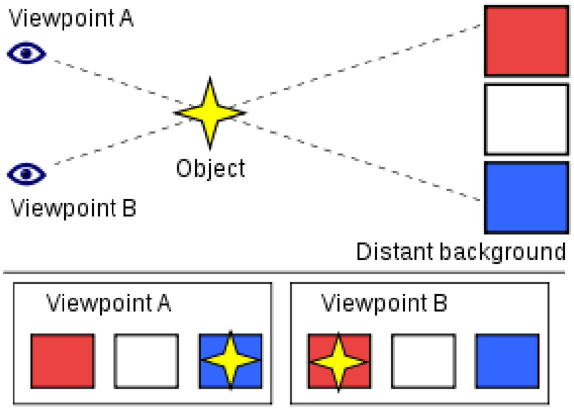


**Figure 1.1:** *Two monoscopic viewpoints (A and B) capture the object in front of different positions of the background. For A, the object appears to be in front of the blue square and for B, the same object appears to be in front of the red square. Figure taken from Wikipedia encyclopedia.*

What Sir Charles Wheatstone built was an arrangement of lenses combined with two pictures of the same object, taken from two different perspectives. Consequently, separating each photography for each eye made the object stand out of the background, and thus providing a 3D impression[1]. Applying this phenomenon to a device, a display must show a compound of two shifted and overlapped images that are representing the same scene. One of these images has to be filtered for each eye, requiring the help of a certain instrument. This method was first implemented with the so-called anaglyph. Here, a color mask (one red and the other cyan) is added to each image and later, those images are superimposed by additive light. The instrument are glasses with color filters (red and cyan), where each glass filters the image with its corresponding added color. This allows to percieve the two views separately. In 1952 M.L. Gunzberg and Arch Oboler released the first color stereoscopic feature, called *Bwana Devil*. The importance of mentioning this film is because it was the first one that used a polarized system, which means that the superimposed images are polarized orthogonally (e.g. linear: one vertically and the other horizontally; circular: clockwise and counterclockwise). In this system the glasses are polarization filters of the same basis (linear or circular), permitting each glass to pass the opposite polarized light. After discovering this technique, producers started creating contents for polarized systems [2].

Both techniques, polarization and anaglyph, are still relevant today. However they were almost forgotten by the general public until the invention of the Imax 3D. It was at the Vancouver's Expo (1986) where Colin Low brought the three-dimensional experience back to the public with the film *Transitions*. Thanks to the Imax experience, general public began to be more interested in the new format of the "seventh art". Meanwhile, 3D productions, videogames business and cartoons started creating their models based on 3D techniques as well.

It is important to remark that all these techniques have been developed according to display's constraints. The recent years, worldwide research activities were started with the aim of developing standards, technologies and production facilities for 3D television (3DTV) and Free Viewpoint Video (FVV). In a first approach, these activities were based on the concept of an end-to-end stereoscopic processing chain (capturing, transmission and display). Here, each view of the stereo pair was treated separately as a unique video stream. Viewing conditions

[1] "Biography of Sir Charles Wheatstone", http://www.acmi.net.au/AIC/WHEATSTONE_BIO.html

and human visual factors had to be taken into account by the camera man during recording, what made 3D productions complicated. Conclusions derived into the need of separating capturing and display geometry by using new methods and 3D video processing tools, yielding to a common, global, flexible and scalable data model that supports the various display types and technologies. This is the Multiview Video plus Depth (MVD) representation. It can be achieved by estimating depth information from the given Multiview Video set from the recording process (meaning 2 or more views) and to use this depth information to create a virtual stereo pair at the receiver side. The depth information is basically the disparity of a pair of images, describing the scene geometry.

In 1998, the European PANORAMA project was one of the first activities on demonstrating the potential of such a depth-based processing approach for stereo adaptation. Meanwhile, the European project ATTEST took this concept and applied it to the requirements of a complete 3DTV processing chain. In this system, the images are transmitted with depth maps providing depth values for each pixel. In general terms, it is broadly accepted that the MVD representation is the groundwork for future 3DTV systems, because it can be built on the existing Digital Video Broadcasting TV infrastructure. Meanwhile, the Motion Picture Expert Group (MPEG) is devoting effort on these tasks and is investigating the needs for standardization in 3D [4].

Regarding 3D contents, recent productions, such as the latter film *Avatar* - directed by James Cameron and winner of three Oscar awards -, attest success. Some other big productions, such as *Alice in Wonderland* - directed by Tim Burton -, are coming to the big screen in a near future. Moreover, first experiments on broadcasting stereoscopic 3D are realized succesfully. This first approach of providing 3D content to the broad public is not based on the MVD representation. However, progress of technologies based on MVD is expected in a near future and thus improving the quality of the 3D experience plus supporting advanced applications.

Accordingly, the work presented here will focus on a system using the MVD representation. The goal of this project is to improve the MVD data obtained by a multiple camera arrangement and which final use would be for a multiview display. In general, depth information estimated from real scences presents distorted silhouettes of the objects in the scene. This is an annoying effect while rendering a virtual view, because pixels around the edge, ones belonging to the background and others to the foreground, are mistakenly recalculated, yielding to a false disparity. Hence, pixels from the background stand out as if they would belong to the foreground and vice versa. With the aim to avoid this distorting effect, edge information from the video data is taken into account for reshaping depth maps' silhouettes.

This work is organized as follows: Chapter 2 describes the MVD representation, how is it obtained, its advantages and deficiencies and the MVD representation from the point of view of an entire 3D processing chain. Chapter 3 presents a comparative statistical analysis of the two components of the MVD representation, introducing analysis methods that are well suited for highlighting the difference between video and depth data. Chapter 4 presents an algorithm for improving the MVD data minimizing the aforementioned deficiencies. In this chapter, visual changes produced by the algorithm are presented. Results of the improved depth maps by applying the new tools introduced in Chapter 3 can be found in chapter 5. Finally, conclusions are drawn in chapter 6.

# Chapter 2

# Multiview Video plus Depth

The human visual system profits from stereopsis to create the depth perception. The brain uses binocular disparity to extract depth information from the two-dimensional images captured by both retinas. Thus, objects are percieved as sets of colors, shapes, textures (those three directly related to the objects' properties) and depths (related to scene's geometry). In a discretized model of the human perception it is also possible to represent a scene by setting a color and a depth value to each pixel, hence representing objects in the scene. Carrying out this idea, the feeling of depth can be achieved with a proper display that renders disparate views to each eye thanks to the depth information. More sophisticated displays need various viewpoints to allow the 3D experience to more than one viewer. While 3DTV allows the impression of depth of the observed scene, FVV permits the interactive selection of viewpoint and direction within a certain operating range. In order to understand the MVD representation better, an overview of these technologies is given first.

FVV provides an experience similar to being in a live show. The capability that allows the user to inspect the scene from several points of view is close to watch, for example, a live concert. This representation offers the same functionality as the one known from 3D computer graphics. Unlike virtual world, real scenes are captured with an array of cameras for FVV. Thus, the multiple camera signals are processed and tranformed to a single scene representation that permits for rendering intermediate views, i.e. selection of arbitrary perspectives and/or directions. Several scene representation formats for FVV are often classified between two extremes [5]. One extreme is the classical 3D computer graphics representation, where the scene geometry is typically described on the basis of 3D wire-frames and meshes. Objects are reproduced using geometric 3D surfaces with an associated texture mapped onto them. The other extreme, usually called Image Based Rendering (IBR), does not use any 3D geometry at all. Here, virtual intermediate views are generated from available real views by interpolation. A technique called Depth Image Based Rendering (DIBR) realizes a solution in between these two extremes. In this case, virtual intermediate views are generated from real views by interpolation from color images and associated per-pixel depth information [6]. In conclusion, DIBR uses scene's geometry information while IBR does not. The main advantage DIBR is the high quality of virtual view synthesis, while avoiding a complex 3D reconstruction. However, it implies dense sampling of the real world with many original views, which generates large amounts of data. Between these two extremes one can find other methods that combine both techniques. As an example, a method can use 3D mesh models (from the first extreme) and view-dependent mapping of multiple textures as acquired from real cameras (from the second extreme). It provides a high quality of rendered views and a bigger viewing zone [1], but for the price of requiring a certain number of cameras, processors and processing time.

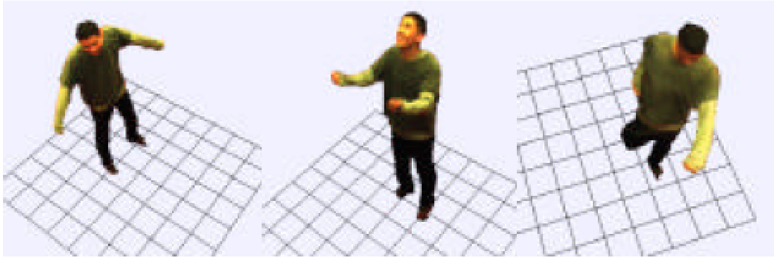


**Figure 2.1:** *Virtual camera flight, rendered views at 3 different times from 3 different viewpoints. Illustration courtesy of [1].*

As previously commented, 3DTV allows the viewer to have the feeling of depth by providing two views, one for each eye. It has also been seen that there are several 3D displays and therefore different 3D processing algorithms. For example, in the case of autostereoscopic displays two virtual views are rendered (one slightly left and one slightly right from the original view) from the color video and associated depth information. Moreover, the rendering process is done at the receiver side, making the structure flexible. Thus, the viewer can adjust the depth impression, such as brightness or color balancing in a classical 2D TV set [1].

**Figure 2.2:** *Illustrative example of depth impression created by a 3D display*

This chapter gives an overview of what do the depth maps represent exactly, how they are computed, and their advantages and deficiencies in section 2.1. Subsequently, the MVD representation from the standpoint of a generic 3D video processing chain, regarding recording, transmission and display processes will be described in section 2.2.

## 2.1 Representation

MVD is a highly flexible 3D representation and, as its name suggests, is a compound of several viewpoints of a scene and each persepective has a color video data plus additional depth information that represents the scene's geometry. On one hand, the Multiview Video is recorded with a multi-camera system. On the other hand, the Depth is estimated with an algorithm whose goal is to generate a depth map for each camera. Such a depth estimation algorithm [4] can be divided in four main modules: the first is for rectification, the second applies disparity matching, the third is for depth map creation and finally a de-rectification module. In order to provide a better understanding of depth maps, an overview of these blocks is presented in the following.

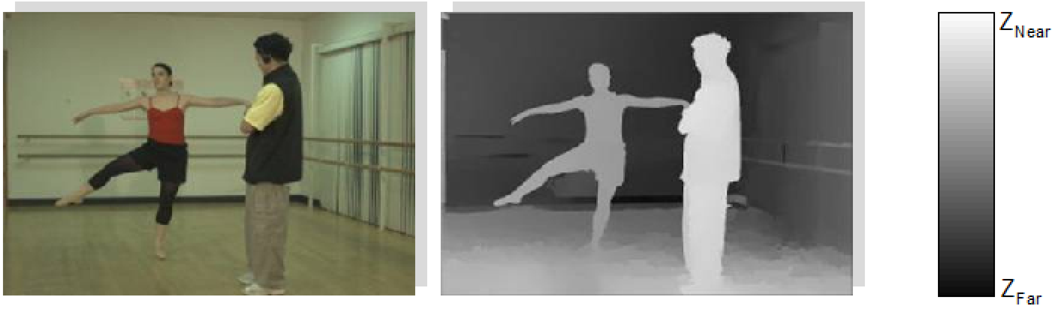
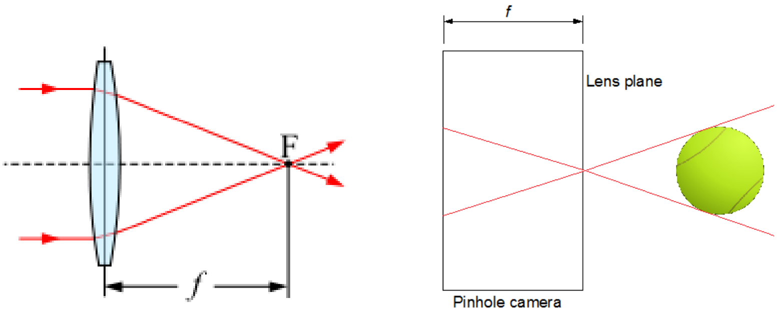


**Figure 2.3:** *Multiview Video plus Depth representation consisting of a set of 2D color video views and their associated depth data*

Prior, it would be good to define the main camera features. The *focal length* of a camera is the distance between the lens plane and its convergence point (see figure 2.4(a)). Referring to a pinhole camera, it is the distance of the hole and the left side of the box (see figure 2.4(b)). For cameras with lens, the hole is filled with a lens. Hence, the light sensor must be placed on the side of the box to capture the light difracted by the lens. Furthermore, the *distance between lenses* is the space between the centers of the two lenses. Finally, the *distance between cameras* or *baseline* (see figure 2.5) is the interval between a camera and its next within an array of cameras (or multi-baseline system).



(a) Lense and focal length (b) Pinhole camera

**Figure 2.4:** *Scheme of a simple pinhole camera. The light passes through a small hole in the middle of the box and is captured in the left side by a film or a sensor. (a) is taken from Wikipedia encyclopedia.*

In a first step, the rectification module works with pairs of adjacent cameras. The rectification process needs the camera parameters because it is based on a rotation of the stereo pair of cameras. The main advantage of this step is that this transformation can be done independently of the depth structure of the scene; just with camera paremeters is enough. This module delivers a rectified camera pair with parallel optical axes, where point correspondences between the two camera images always lie on the same scan line (see figure 2.5). Consequently, the search correspondences can be limited to the horizontal direction, what eases the subsequent processing.

Hence, it provides a unified data structure (the data is independent of the set-up of arbitrary views) and makes the 3D processing flexible. However, there is some loss of information as well. Working in pairs of images leads to have twice data rectification for each camera (from camera $i$ to adjacent cameras $i+1$ and $i-1$), except for the camera borders (cameras 1 and $N$). Thus, this block will render $N-1$ rectified image pairs. I.e. a multi-basline system with 3 cameras would lead to 2 image pairs (one for the pair [cam0, cam1] and another one for the pair [cam1,cam2]).

After rectification, a disparity matching is employed to each rectified pair of views. One solution for this goal is a fast hybrid-recursive-matching algorithm [7]. It delivers extremly smooth and temporally consistent disparity maps, which means highly redundant information and no random noise due to its recursive structure. This algorithm is applied twice to each image pair, once from the left to right view and vice versa. However, this algorithm causes mismatches in critical image areas. These mismatches are detected with a confidence interval, which is obtained by calculating the cross-correlation. This criterion uses a threshold to distinguish between good and false detections, which can be caused by ambiguities in correlation analysis and by occluded areas. The first cause is due to multiple point correspondences found by the matcher. The second cause is due to non-existent point correspondences and consequently they cannot be matched. Then, the rejected disparity values are recomputed by utilizing segmentation-based interpolation, including two techniques: color clustering and change detection. The operators selected for the interpolation are different, depending on segment size and motion content. Hence, a new disparity map is obtained and another consistency check is applied, which will only detect the occluded areas.

Once having the disparity maps well computed, the depth maps are calculated by using the following relation:

$$ZM_{i,j}(u_{i,j},v_{i,j}) = \frac{F_{i,j}B_{i,j}}{|DM_{i,j}(u_{i,j},v_{i,j}) + h_{i,j} - h_{j,i}|} \tag{2.1}$$

Here, $u$ and $v$ are the point coordinates and $ZM_{i,j}(u_{i,j},v_{i,j})$ is the depth map of views $i,j$. The baseline is represented by $B_{i,j}$ and $F_{i,j}$ is the focal length. Finally, $DM_{i,j}(u_{i,j},v_{i,j})$ is the disparity map and $h_{i,j}, h_{j,i}$ are the sensor-shift offsets from the rectification step. Note, that $F_{i,j}$ is usually set to $F_{i,j} = (F_i + F_j)/2$ during rectification. Next, $ZM_{i,j}(u_{i,j},v_{i,j})$ is derectified and merged to a single depth map, associated to camera $i$. Thus, depth maps will correspond to original non-rectified views, which can be easily achieved by the inverse operations used in the rectification step. Cameras 1 and $N$ are directly set to their nearest depth map, while the other views are obtained by averaging their neighbors. For the occluded areas, data is taken from one to the other. Otherwise, data is available from both views and therefore simply averaged. Optionally, each single depth map can be smoothed with a 2D Gaussian low-pass filter, as done in the ATTEST project [8]. It is useful when cameras are relatively close to each other and therefore the size of the occluded areas is small. However, it is a problem for crucial depth information, such as edges, because they become blurred and not sharp as they should be.

Technically, the depth range of the scene has to be limited to the $z_{near}$ and $z_{far}$ planes (see 2.3). Thus, the range $[z_{near}, z_{far}]$ is quantized to the 8-bit range $[0, d_{max}-1]$[1]. Figure 2.5 exemplifies the inverse quantization, which has the advantage of quantizing the foreground more accurately than the background.

[1] Note: hereafter, the background will be related to lower depth values and, consequently, the foreground will mean higher depth values

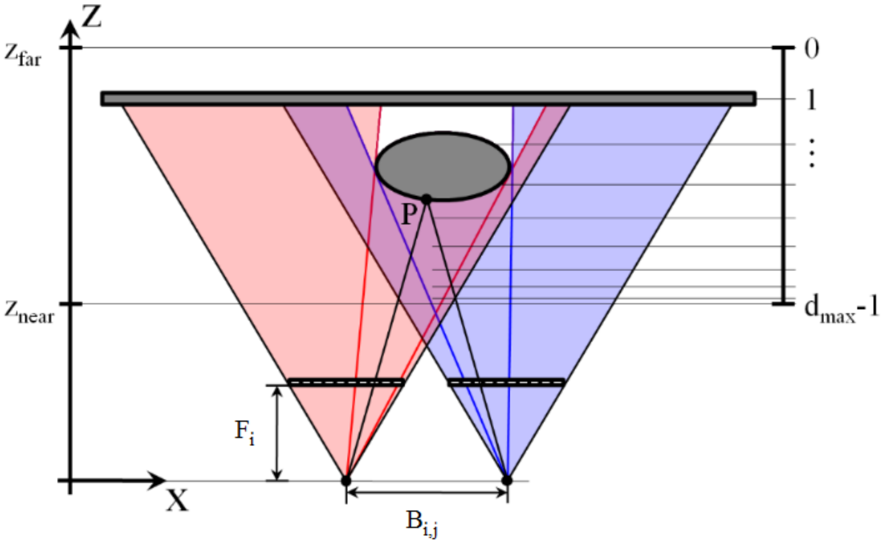
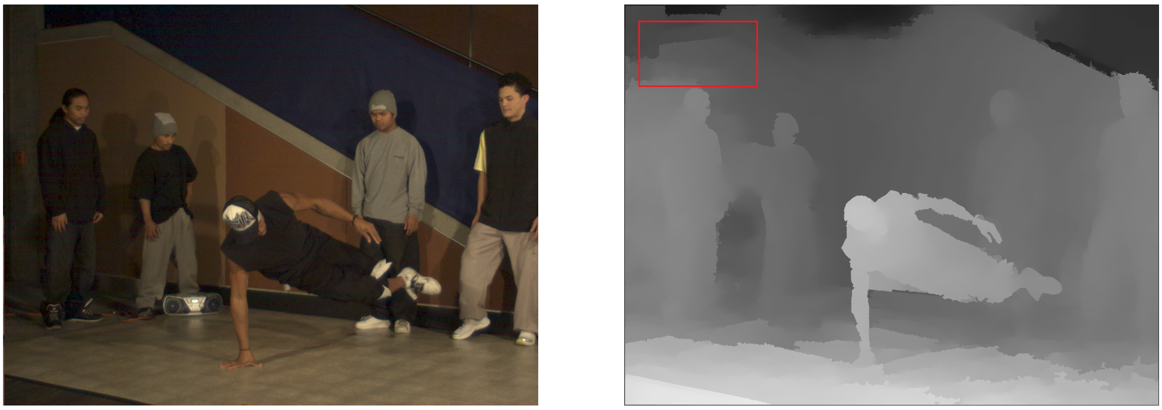


**Figure 2.5:** *Relation between 3D scene, multi-camera geometry and 2D depth maps. Point correspondences converge into the same depth plane. Inverse quantization is applied in order to adapt better to human depth perception.*

As mentioned before, the main problem while computing the depth maps are occluded areas, where data is not available from both views. Figure 2.5 illustrates two occluded areas, one for each camera. For the left camera, a purple-colored triangle behind the object represents an area that just the right camera can capture. Vice versa, the red-colored triangle is the occluded area for the right camera. However, the algorithm is robust enough to complete the whole depth map and to give a consisten output (meaning no empty patches of data). Appart of this fact, if the image is smoothed, like done in the ATTEST project, then all depth transitions at object borders are degraded. However, for rendering it is extremely important to keep all contours in the right position. Hence, this work will focus on adjusting the object borders of the depth maps according to the color information. For this purpose, an edge detector will be needed and, moreover, an algorithm capable to refit the silhouettes of the depth maps according to the edge information.

Another problem not solved in the depth map representation is the non-homogeneity in the background areas. Some sequences, like Breakdancers or Ballet (see Figure 2.6), present a stair-effect in background regions instead of a smooth depth change. This will cause strong visual discomfort while rendering because the objects will not be well placed according to the scene's geometry.

In conclusion, the multiview video plus depth representation is a way to describe a scene, giving several points of view and, for each view, its color information and a depth map associated (tipically with 8 bits per pixel). For coding, the video is usually sent in a YUV format (in the allowed formats 4:4:4, 4:2:2 or 4:2:0) and the depth is sent as a simple luminance channel in a YUV 4:0:0 format [8].



(a) Color from camera 0 at frame 8

(b) Depth data of camera 0 at frame 8

**Figure 2.6:** *Non-homogeneity in the background. Most of the evidence marked at the top of the image.*

## 2.2 The MVD processing chain

The future of 3D systems must be focused on a model such as current TV broadcasting, which consists of three main modules: recording, transmission and display. Moreover, something important to highlight is that 3D systems will have to regard all the display possibilities, because nowadays they can take profit of an all-digital content processing chain. All things considered, 3D algorithms can be easily added to the existing pipeline (recording, transmission and display) without modifying the other processes. Thus, it allows for emerging oportunities in 3D processing, including anti-aliasing, compression and image-enhancement. The following sub-sections show how MVD data is acquired, how is transmitted and how it is displayed on the different 3D display systems.

### 2.2.1 Recording

Cameras have improved significantly in the recent years leading to high quality recording. Having a look at the recent history of 3D, all productions have been made off-line, being storaged and not transmitted and utilizing post-processing tools. Nevertheless, in 2010, first capturing and broadcasting tests have been succesfully done. In the case of capturing the scene, there are several options of recorders, depending on the final use of the data. Thus, the main models of recording to obtain depth maps or MVD data are presented: the camera array set and the 3D-Camera based on the fringe projector technique [9].

**Multi-camera array**

This technique is based on traditional 2D Cameras arranged in a specific way. In the case of just two cameras it will render a stereoscopic vision, but having a higher order leads to the multiview video representation. Usually, cameras are positioned on a circumference arch, but they can also be placed in a linear segment. The number of cameras ranges from two to about 128[2], while in practical systems typically 2-8 cameras are used.

[2] "The Stanford Multi-camera Array", http://graphics.stanford.edu/projects/array/

This kind of set-ups have three main problems, due to the use of different devices. The first one is synchronization. At the moment that cameras start recording the scene, all of them must be synchronized with the same master clock. Thus, in a synchronized multi-camera array system, the frame $F$ of a camera $X$ represents the same time instant $t$ as any other camera $Y$. Another problem is the adjustment of color balancing. All cameras must be calibrated in color, meaning that all cameras must capture and deliver the same tonality for the same recorded color pattern. I.e. a good pattern is the chroma key color pattern -usually green or blue- for compositing two different images together. The third aspect to be aware of is the geometrical camera calibration [10]. This is highly important for the rectification step when computing depth maps (see section 2.1) and for rendering the stereo video pair. Otherwise, the rotation step would not align correctly the optical axes and consequently depth maps would not be well computed. To calibrate the array, an algorithm computes the "pixel error" regarding all cameras (instead of calibrating each camera individually). To achieve this, the algorithm calculates the difference between the observed images of 3D points and what is predicted by the current estimate of model parameters[3]. In summary, it is necessary to have all cameras synchronized to ensure time correspondences. It is important to have all cameras calibrated with the same color balancing for, i.e. point correspondences when computing depth maps. Finally, is crucial to have correct calibration parameters for rectification and de-rectification steps when computing depth maps. The main benefit of this technique is that it renders various points of view, but it carries the inconvenience of a high amount of data to be transmitted. Loosely speaking, if N is the order of the array (or number of cameras), the data to be transmited will be N times a monoscopic view (a 2D camera). To solve this problem compression is needed, for which temporal and spatial redudancy can be exploited.

Once having a Multiview Video set, an algorithm (as the explained in section 2.1) is used to compute the Depth Maps. Hence, MVD data can be obtained from a multi-array camera system plus the mentioned algorithm.

### 3D camera based on fringe projection

The fringe projection technique has its origins in topography. It was presented by Rowe and Welford in 1967 and the basic idea is to project a fringe pattern on an object and view it from another point of view. Here, the projector and the camera are just separated by a distance comparable to the human eyes' distance (around 50mm). Thus, typical problems such as shadowing can be avoided. Moreover, it is well adapted to the human perception. However, this device has a more similar behaviour to a scan rather than to a traditional camera (it needs at least 4s to obtain a depth map from the scene). Some application fields for this type of devide are capturing depth inside small rooms or from a whole 3D scene, a mobile subsystem for scalable topometry (such as a cube where the camera can move in by the three directions X, Y and Z) and augmented reality. Some advantages of this camera are high brightness, large depth of focus and compact design.

In order to obtain MVD data, this camera needs a color video camera that records the same scene (in the case of having just one camera based on fringe projection). The distance between the color video camera and the fringe detector must be considered as well. Moreover, the color video camera have to acquire the scene when there is no light pattern from the fringe projection. Otherwise, the color camera would capture the scene plus the additive light of the fringe projection.

[3] "Robust Multi-camera Calibration", http://www-graphics.stanford.edu/ vaibhav/projects/calib-cs205/

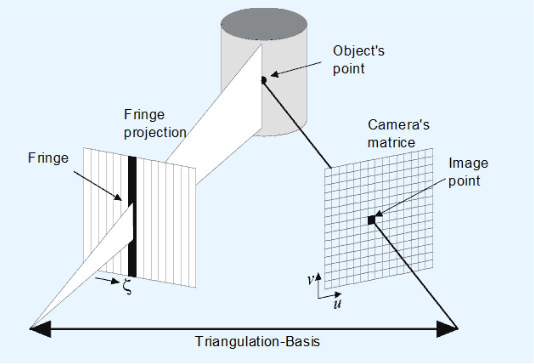


**Figure 2.7:** *Schematic representation of the fringe projection*

Some other devices that are not based on traditional camera systems are depth sensors. By combining color video capture (with a normal camera) with depth acquisition (with a depth sensor) and with an appropiated synchonization module, a MVD data representation can be obtained. Another kind of camera is the so-called stereoscopic video movie camera "3D-CAM". The reader can be referred to [11] for more documentation on this field.

### 2.2.2 Transmission

In terms of telecommunications, transmission is the process of sending and receiving an analogue or digital signal over a capable medium (point-to-point or point-to-multipoint), either wired or wireless. Related to TV systems, historically data has been sent as an analogue signal, in a wireless medium of a point-to-multipoint infrastructure (also called broadcast or multicast). The first big change that the TV system had, was with the transmission of color video signals instead of just the luminance channel. In the recent years, due to the use of satellite communications, TV signals have migrated to the digital format, and also terrestrial emissions with the implantation of the Digital Terrestrial Television (hereafter DTT). Soon, TV channels will succeed with the third big change of TV transmissions, offering to the public the 3D experience.

Recently, TV broadcaster SkySports made history offering the 3D experience with the Premier Leagues' football match Manchester United vs. Arsenal [4]. The reviews pointed out that people really enjoyed the experience, indicating a good acceptance for the global public. Over the next days, this company will open a new channel broadcasting 3D contents. Also another broadcaster, ESPN, foresees to broadcast a total of 25 matches of the upcoming FIFA Wolrd Cup 2010 in South Africa.
They chose to use a stereoscopic system, based on polarized screens and with the need of polarized glasses. The codec used was MPEG-4 and the stereoscopic encode format was side-by-side compressed within a 1080i25 frame[5]. According to the mentioned broadcaster's specifications, this system uses Linear or Horizontal line based encoding (not Quincux based).

Offering 3D contents for live events introduce a point of analysis: given the current TV sys-

[4] As a curiosity, Manchester's supporters said that Manchester United also made history because they won 3D-to-1. http://www.skysports.com/story/0,19528,11096_5889013,00.html

[5] "BSkyB 3D technical specification for PlanoStereocopic (3D) program content", http://introducingsky3d.sky.com/a/bskyb-3d-tech-spec/

tem (where several parameters and procedures are defined), it is necessary to adapt the new amounts of data to this context. Technically, the standard used for this purpose is the worldwide known MPEG-2 (also known as ITU-T H.262) [12]. It is also present in the DTT and satellite transmission, although this last one is migrating to MPEG-4 [13] because of the increase of contents with High-Definition TV (HDTV) resolution. What these codecs offer is a high quality video representation within the existing digital transmission capacities due to diversification of network types and their characteristic formatting and loss/error robustness requirements. The most typical codec of the ITU-T recommendations is the H.264/AVC standard [14] and gives an approach to several networks with different applications (storaging, broadcasting, streaming or video telephony). It is important to remark that in all the ITU-T and ISO/IEC video coding standards the scope of video coding standardization is the decoder, allowing freedom to optimize implementations for specific applications.

It will be seen in section 2.2.3 that some applications and display systems need to receive different points of view to represent 3DTV or FVV. In FVV the 3DVO, a 3D objects that represents the 3D geometry and the texture of an object, is also supported by MPEG-4. Thus, it is possible to be decoded with an appropiate MPEG-4 player, although more refined representations cannot be included in this format. To solve it, an update of the computer graphics part of MPEG-4 called Animated Framework eXtension (henceforth AFX) was added [1]. Then, FVV can be now easily transmitted with the standard MPEG-4 AFX. On the other hand, in stereo-3DTV the sender has to transmit the color information of an stereo video pair. However, with depth enhanced 3DTV (3DTV-MVD) the sender just has to give the color information (for an RGB space uncompressed data it means 3 bytes) plus a depth value (just 1 byte) for each view instead of wasting bandwidth transmitting two color components (3 bytes plus 3 bytes) per view. Thus, with appropiate algorithms the receiver can render the two views allowing the user the 3D experience. In a first approach, it was thought to use H.264/AVC, but afterwards it was seen that a specific Multiview Video Coding (hereafter MVC) presents better results for multiview video (but not for video plus depth). The basic idea is to exploit the temporal redundance (as in H.264/AVC) but also the spatial one (not considered with H.264/AVC and hence called simulcast). Anyway, in the ATTEST programm the transmission overhead for the depth information was between 10%-20%[6] compared to a conventional 2D broadcasting [4]. Note that in order to achieve this solution the camera parameters (intrinsic and extrinsic, such as the focal length or the baseline) are essential and must be transmitted (see section 2.1) as auxiliar information.

For more information on 3D representation and coding refer to [15].

### 2.2.3 Displays

As it has been previously introduced, from parallax stereograms (early 20th century), through polarized 3D and personal 3D cameras (1950s), holography (1960s and 1970s), to 3D films by IMAX (1980s and 1990s) and digital (nowadays), advances in optics, electronic and processing algorithms have resulted in significant improvements in 3D quality and visual comfort, increasing the benefit while decreasing the cost. At present, technology has led to five main different

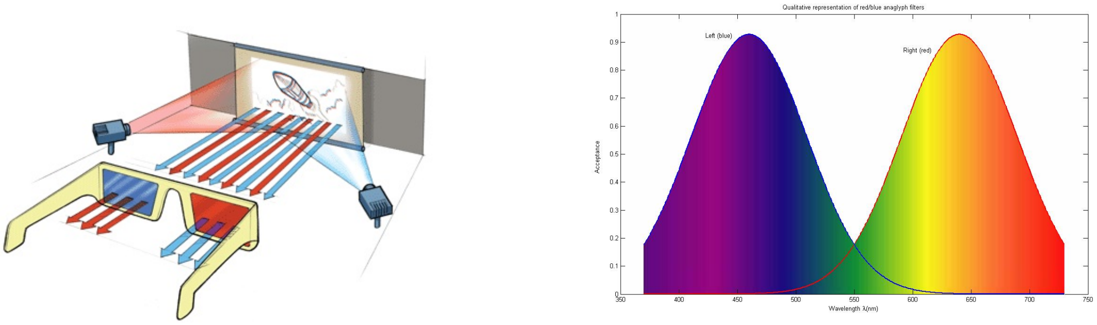
[6] These quantities were achieved under very specific circumstances. 30%-40% are more realistic for practical systems.

3D display technologies: stereoscopic, autostereoscopic, multiview, holographic and volumetric displays. However, not all these technologies benefit from the MVD representation. According to the number of views needed in a MVD data representation, the principles of how the displays work and their intrinsic problems are introduced [2].

### Stereoscopic

Also called glasses-based, stereoscopic systems are widely-used because they can provide the 3D experience at a relative low cost. Moreover, these technologies can be easily applied to large-scale electronic 3D displays. Nevertheless, glasses-based methods were the first ones that allowed people to have the 3D experience.

- **Anaglyph**
  The visual light corresponds to the range (370,730)nm and the most simple division in two bands yields two segments: (370,550)nm and (550,730)nm (see Figure 2.8). Thanks to color introduction in printing and photography these subsets can be used for a mechanism for left/rigth channel separation. This is the basis for anaglyph [16]. As shown in figure 2.8(b), the blue filter curve covers the blue and a part of the green bands, and the red response filter allows passing the other part of the green band and obviously the red one.



(a) Anaglyph glasses

(b) Band-pass light filters

**Figure 2.8:** *Anaglyph filters. (a) Scheme of Anaglyph glasses and display, and (b) Qualitative representation of blue (left) and red (right) band-pass filters.*

One of the advantages of anaglyph is that it is the most printer-compatible 3D system. Because of this, any electronic device capable at representing color or printed material, such as film or paper, permits the three-dimensional representation. Another important point is the price to produce the glasses. It is the least expensive and the most known for the public. On the other hand, this technique has some deficencies. First of all, each eye does not percieve the same spectrum range due to filtering. Secondly, and also the most common in all 3D displays, is the crosstalk between channels; it is inherent that perfect separation of the channels is impossible, adding cross information one to the other. Finally, the limited color representation capability.
However, some improvements could be done. As a primary idea, lenses refined with more frequency selectivity can improve color perception. But the most significant improvement is due to signal processing by optimizing the output (anaglyph images) under its nature constraints (absorption curves of the lenses, spectral density functions of the display pri-

maries and colorimetric features of the human visual system). This solution is proposed by Dubois [17] and basically is, in terms of transmission, a kind of equalization to avoid discoloration.

- **Multi-band filters**
  The main trouble of anaglyph is derived from the fact that each eye captures the light in a single band (it is splited into two bands), a matter that does not allow full color perception and introduces color rivalry between eyes. Thus, this technique divides the spectrum range into two complementary sets of wavelengths, giving a red, green and blue (henceforth RGB) perception to each eye[7]. In Figure 2.9 can be seen that a set of three band-pass filters corresponds to the right eye and the other set to the left. The benefit of using this separation, is that human eye can not distinguish different compositions of the same color, but in order to achieve this phenomenom there must be an accurate color representation and no visual discomforts.

  Nevertheless, as it happens to every technology, there are inconvenients as well. Because of the use of complementary filters, more crosstalk is present. Depending on the accuracy of the filters it can be minimized, as explained before, with some signal processing. Also these glasses contribute to typical colorimetry problems, always present at any display. Finally, note that this technique is relatively new and consequently more expensive than traditional anaglyph or polarized glasses.

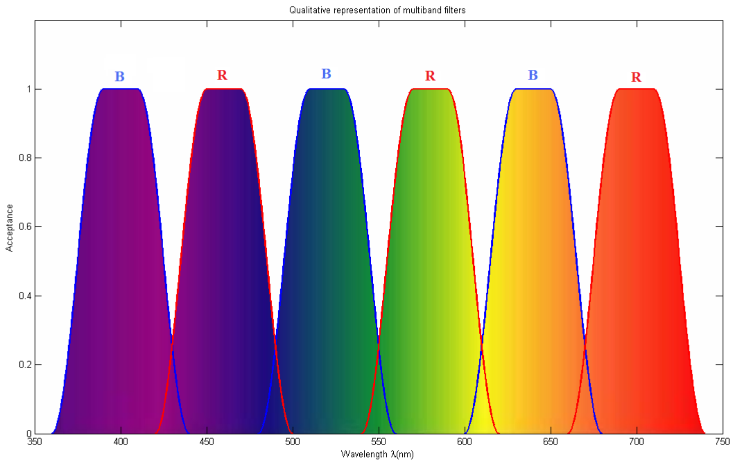
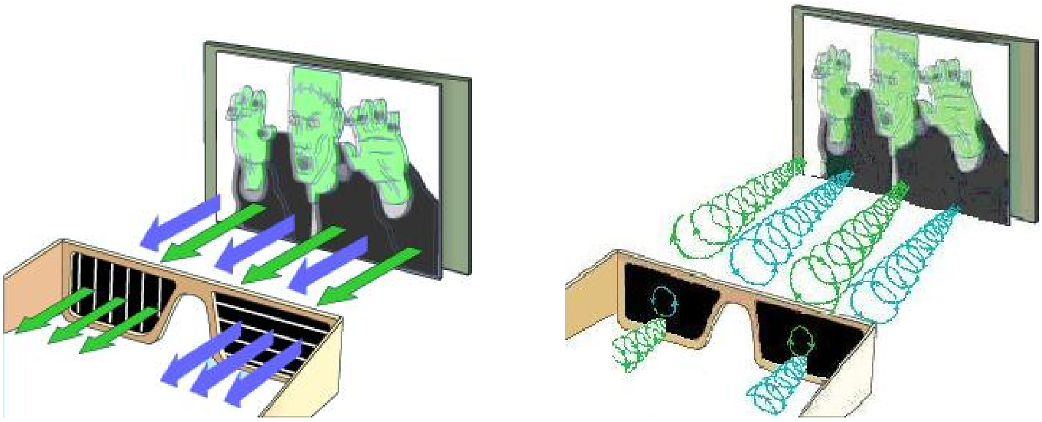


**Figure 2.9:** *Complementary sets of wavelengths according to RGB space for right (red) and left (blue) views.*

- **Light Polarization**
  In the way of resolving full-color representation troubles of anaglyph systems [16], polarized glasses permit to see the whole color gamut. In that case, lenses are polarized orthogonally (see Figure 2.10 as example). Besides, a non-depolarizing screen is needed to ensure that polarization is maintained during projection.
  However, the perfect calibration of this orthogonality is practically impossible, leading to

[7] In the RGB space, red, green and blue channels are orthogonal, thus all colors can be represented as a combination of the basis' vectors.

crosstalk. Moreover, if linear polarization is used, rotation causes more crosstalk; the more the viewer moves (rotates) his head, the more optical distortion is present. Therefore, it is inconvenient to be totally aligned with the screen if linear polarization is utilized. Anyway, this fact can be facilely solved using circular polarization, due to their invariance against rotation. Nevertheless, it increases the cost of manufacturing as circular polarization glasses are more expensive than linear ones. By the side of the projector, other features are required. Due to light blocking by filters, projectors have to increase their brightness. Second, the alignement of the two projectors must be carefully attended. This can be mitigated by using only one projector at double refresh rate and with a dynamic modulator polarity switching in synchronism with interleaved images. This solution can be adopted by TV sets, inasmuch as they just use a source of emitting light. Notwithstanding, these solutions do not erase all the problems, because crosstalk is always present. Note that using circular polarization makes the crosstalk to have a shift-invariant behaviour, while using linear polarization makes the problem almost insolvable. In order to minimize the crosstalk, signal processing solutions are needed. Furthermore, by increasing the brightness not all colors will have the same gain, because of the filter response.



(a) Linear Polarization

(b) Circular Polarization

**Figure 2.10:** *Scheme of Polarized glasses. Lenses block the orientation marked. (a) Linear polarization, and (b) Circular polarization*

- **Light shuttering**
  Instead of using light polarization, a mechanism that blocks light is applied by fast switching glasses (working in synchronism with the screen), that become transparent when the intended image is displayed and opaque when the view is not the desired. Hence, this mechanism requires a display working at double refresh rate (at least). I.e., if the normal refresh rate is 60Hz it must work at 120Hz (as said, at least). The main benefits of this technique are full Cathode Ray Tube (CRT) color gamut, full CRT spatial resolution. Though, the synchronism based method carries some disadvantages, such as expensive cost of Liquid-crystal shutters (LCS) manufacturing or the omnipresent optical crosstalk in these glasses-based systems. However, the crosstalk in CRT/LCS compound system has not the same nature as, for example, light polarization. It is produced by screen phosphor persistence, LCS extinction characteristics (the afterglow of one image into the another) and LCS asynchronism (also called timing errors). Once more, the solution proposed to reduce the crosstalk is somehow an equalization, doing a pre-distortion of the image. Afterwards, with the display's lacks, the system is able to recover the original

image.

As it has been seen, each model has its benefits and inconveniences. It is also important to mention that each technology is based on the deficiencies of the previous and thus it tries to yield what the previous cannot offer by their nature. On table 2.1 benefits and advantages of each method are summarized.

| **Technique** | **Benefits** | **Disadvantages** |
|---|---|---|
| Anaglyph | Least expensive<br>Applicable to any color image | Color rivalry & no full color perception<br>Optical crosstalk |
| Multi-band filters | Full-color representation | More expensive<br>Optical crosstalk |
| Light Polarization | Less expensive<br>Full-color representation | Non-depolarizing screen needed<br>Crosstalk |
| Light Shuttering | Full CRT color gamut<br>Full CRT spatial resolution | Most expensive<br>Crosstalk |

**Table 2.1:** *Summary of benefits and disadvantages of stereoscopic displays, with special regard to crosstalk and color perception.*

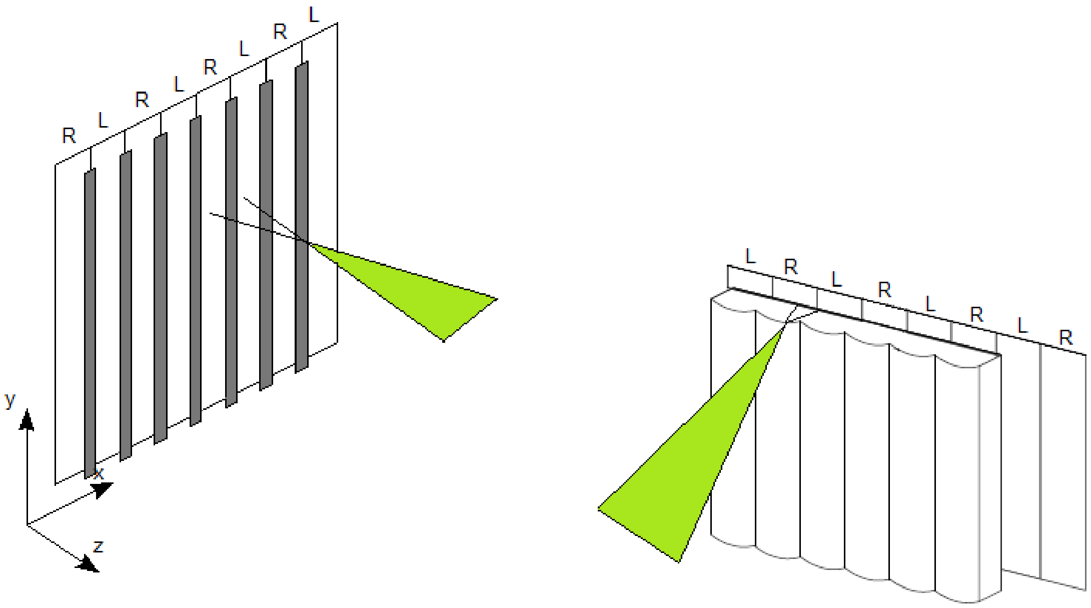
### Autostereoscopic

Potential technical problems such as projector missalignment or viewing discommodity along with public discomfort (the need to wear glasses and cost and care of them) led to a glasses-free model for the 3D experience. Since the beginnings of the 20th century, techniques were based on spatial multiplexing of right and left images interacting with a light-directing mechanism capable to exhibit each view to the viewer's eyes. The most well-known multiplexing techniques are the parallax barrier displays and the microlens displays. Here the main characteristics of both methods are presented.

- **Parallax-barrier displays**
  This technique involves an arrangement of opaque slits distributed and spaced horizontally, placed close to a pixel-addressable screen, such as a Liquid Crystal Display (hereafter LCD) or plasma (see 2.11(a)). Hence, if this slit's layer is carefully aligned with pixel distribution, each slit contributes as an approximate pin-hole projector, rendering two different perspectives to the viewer's eyes.

- **Microlens displays**
  Also just called lenticulars, a layer of narrow and thin semicylindrical microlenses is set in front of the screen at approximately one focal length. Thus, light passes through these lenses and it is directed towards the viewer's eyes (see 2.11(b)). If the lenses are precisely aligned with pixel columns, the viewer (standing at the correct distance) will be able to have the 3D experience. However, lens aberrations reduce contrast of images from adjacent pixels and cause optical crosstalk.

Therefore, by addressing (parallax-barrier) or deflecting (lenticulars) the light to some points of the space creates the viewing zone and, as it suggests, is the subset of the space where the right eye percieves the right image and the left sees the left image. The size and shape of the



(a) Parallax barrier

(b) Lenticular

**Figure 2.11:** *Scheme of Stereoscopic glasses-free displays and their viewing zone. (a) Parallax-barrier and, (b) Lenticular displays. Consider that each set labeled as R (right) or L(left) is a RGB pixel representation.*

viewing zone is directly related to the geometry of the technique. For example, taking into account the coordinates reference from figure 2.11, the viewing zone is invariant on the Y axis but not on the X and the Z axis. Then, watching the screen out of the viewing zone could cause the depth-inverted effect, where right eye sees left image and vice versa. This may cause visual discomfort because correct (called orthoscopic) and inverted (pseudoscopic) view zones are adjacent, also considered as optical crosstalk. Table 2.2 summarizes the possible combinations of percieved images [8]. By using these techniques, both present a reduction to the half of horizontal spatial resolution, delivering half of the information shown in the screen to the left and the other half to the right eye. Thus, both images have to be horizontally sub-sampled and consequently a half-band horizontal lowpass filter must be supplied in order to avoid aliasing.

| **Right eye** | **Left eye** | **Theoretical result** |
|---|---|---|
| Right | Left | Correct 3D experience |
| Left | Right | Inverted depth perception |
| Right | Right | Approximately 2D view of right data |
| Left | Left | Approximately 2D view of left data |

**Table 2.2:** *Possible combinations of percieved images and their theoretical results*

Introducing band guards with no pixel information between left and right views is the most common method to mimize crosstalk. However, it reduces the viewing zone and a new set of combinations should be added to table 2.2. Simply consider that it may deliver black image data to the eyes.

[8] The position of left and right is suposed to be adjacent, otherwise the result is not correct.

Finally, note that these techniques are applicable for LCD or plasma screens, but not for CRT displays, as the light patterns on a CRT screen suffer from jitter, which it is not accepted by the viewer.

### Multiview

The two stereoscopic techniques introduced previously, present the views independent of the viewer's position, a fact that can lead to confusion about the shape of an object. In other words, if the viewer changes the point of view the shape of an object does not. Thus, it is a good improvement to allow the viewer to have motion parallax, which is a complement to stereopsis that models better the human visual system. The way to implement this idea is based on displaying several viewpoints of the scene. This method has two variants: active multiview displays, where the position of the viewer is tracked for rendering the according view of the scene; and passive multiview displays, where regardless the viewers' position several (ideally all) the views of the scene are displayed.

- **Active Multiview**
  As it has been commented, that the display provides two images by tracking the viewer's location. The optical mechanism could be one of those previously described. Though, this method has one inconvenience: due to head tracking, practical systems are restricetd to a single user, which is a weakness considering television systems. Another fact worth mentioning is that depending on the motion pattern of a viewer's head temporally irregular view multiplexing takes place. Fortunately, this problem can be solved with a temporal antialiasing filter.
- **Passive Multiview**
  The main motivation to do one more step is to support multiple viewers and to avoid head tracking. By presenting several points of view at the same time, this problem can be solved. The light is then directed to different viewing zones, at the expense of reducing spatial resolution. If a viewer is in a correct distance to the screen he can have the real FVV experience. In terms of advance, it is a huge progress to have motion parallax, to support multiple viewers and to eliminate glasses for users' comfort. The passive multiview displays use lenticular systems to diffract the light and they provide horizontal parallax only, meaning that viewers can move from side to side, but not vertically. The full-parallax displays are currently weak, because the pixel density of LCD and plasma displays is not high enough. Moreover, the number of viewers is restricted according to the order of the horizontal and spatial resolution. In other words, the spatial resolution gets divided by a factor of N viewers. This causes an annoying imbalance if the number of viewers is great, but can be solved by slanting the lenses. It resolves the sub-sampling and pixels appear distributted more uniformly. Nevertheless, it introduces a problem of alisasing because of irregular sub-sampling.

Another method worth mentioning is the projectional multiview display. It is based on multiple projectors combined with optics that shape individual view zones. Like the other techniques, the fact of showing multiple views makes this technology susceptible to aliasing artifacts. The most common kind of aliasing is the so-called interperspective. It occurs when the spatial frequency of a displayed 3D point in a scene is higher than the display's resolution. Visually, it splits a continous blurred part of the image into fragments and can be prevented during recording or corrected while editing.

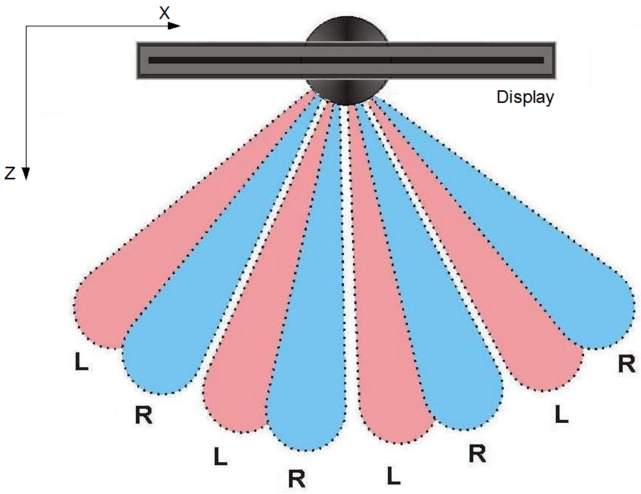


**Figure 2.12:** *Outline of a mutliview display with 4 possible points of view and their viewing zones*

The influence of the MVD representation in the display module can be analyzed in accordance with the number of views needed. In the case of the displays previously introduced, the stereoscopic and the autostereoscopic systems just need one view (meaning one video component and its per-pixel depth information), while the multiview systems need at least one perspective (in the case of the passive multiview, more than one video plus depth views). For the first subset (one view), the MVD is not totally exploited. Rendering the second view from video plus depth will always show artifacts, because no information for filling the disocclusions is available. By using two views, it leads to a better quality, because the second video plus depth view can be used to fill disoclusions with real information. Nevertheless, it is also interesting from the standpoint of transmitting, inasmuch as the overall of data transmitted is less than the two video signals (left and right). In both cases, the rendered views will be a pair compound by the original view and a virtual one, which is obtained by shifting the original view (to the right or to the left, depending on the original view). For the other subset (more than one view needed) the MVD is used to render two images (left and right) for each viewpoint. It should be clear that this process of rendering must be done at the reciever side, thus, in 3DTV, the display will create the virtual views through the color information and its depth map associated.

For a global scope of display technologies the lecture of [2] is honestly recommended.

# Chapter 3

# Analysis of MVD

Multiview video plus depth (MVD) data consists of color video plus depth sequences of one to N views. Color component is typically captured with a mutli-camera array system (as seen in section 2.2.1). Depth component is obtained from color video set with an algorithm that at least needs two views (as presented in section 2.1). As depth maps represent the scene geometry, their features differ from color video data. The main motivation of this chapter is to show the most significant differences between color video and its associated depth map.

To show statistical features of the given data set, known analysis techniques will be used in this chapter, such as the one-dimensional histogram (section 3.1) and the spectrum of an image (section 3.2). However, other analysis tools are recquired to obtain insights into similarities and differencies between color and depth components. Therefore, a new method called Correlation Histogram is introduced to illustrate the main characteristics of the so-called depth maps (section 3.3). This method is also useful to distinguish between color and depth characteristics. The results of this method will be shown and discussed in section 3.4. Finally, the Correlation Histogram method will later be used to test the output of the solution proposed in Chapter 4.

## 3.1 Histogram

A histogram of a grey-level image is a discrete one-dimensional function that shows the amount of occurrences of its grey values. In the case of this work, the range of grey values is [0, 255], corresponding to 1 byte per pixel. Formally, a histogram is described as:

$$H(r_k) = n_k \tag{3.1}$$

where $r_k$ is the $k$th grey level and $n_k$ the number of pixels with grey level $r_k$. Thus, a histogram will have 256 bins $H(r_k)$. Generally, a histogram is normalized by dividing each bin by the total number of pixels of the image ($N$). A constant image will have the same value $r_k$ for all its pixels, what means in a normalized histogram, $H(r_k) = 1$ and $H(r_l) = 0$ for $l \in [0, 255]$ and $\neq k$. Loosely speaking, it gives an intuitive distribution of probabilities of grey occurences in the given image. It must be considered that, for a sequence, if the scene does not change, this distribution is fairly approximated. Otherwise, actions with the camera (like a panning, a zoom in/out or directly a change of the scene) will introduce a lot of changes in the scene. The description of the used MVD data set can be found in the Appendix.

Histograms of color images representing real scenes typically shows soft transitions from one value to the next (see figure 3.1). Real images contain several various textures, with different color tonalities. Textures are associated to the physical object's properties. Formally, texture is defined as the spatial variation in pixel intensities. Hence, for smooth textures (which are the most common, i.e. clear sky, human skin, etc.) the variation of one pixel to its neighbors will not differ in exceed, while for rough textures pixel intensities will differ more. Mathematically, if texture is modeled with a sine function, smooth textures have a lower amplitude than rough ones.

On the other hand, depth maps can vary according to the way in which they have been obtained. Moreover, in the case of disparity maps not the whole range of values is used, and therefore there will be a higher concentration in particular values. However, the shape of a histogram of a depth map (or a disparity map) reveals more information of the scene's geometry than the color image (see figure 3.1). Easily explained, the input of the color histogram is the luminance channel, showing how the luminance is distributed in the scene. On the other hand, the input of a depth histogram is grey values whose are directly related to the scene's geometry (see figure 2.5). Consequently, the background and the foreground can be effortlessly identified in a depth histogram.

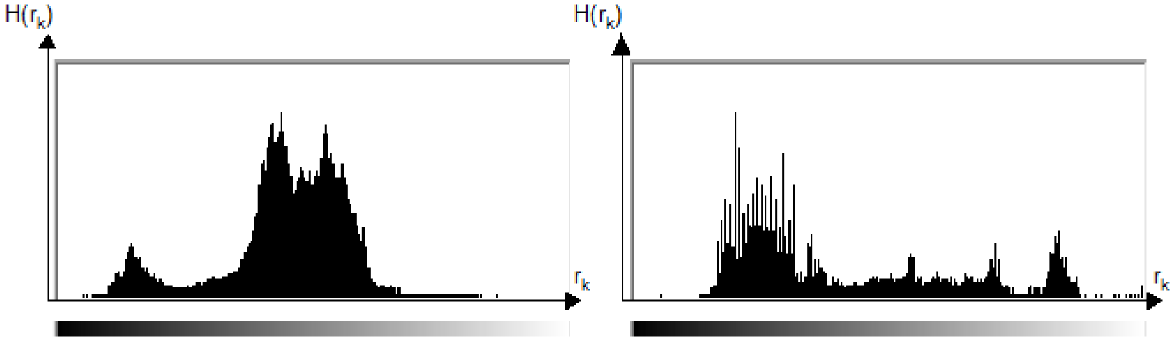


**Figure 3.1:** *Example of two histograms, corresponding to first frame of camera 0 of ballet sequence. Color component (left) and depth component (right).*

One of the reasons to use image histograms is because some characteristics, like the range of values that is being used in the representation, are easy to exbtract. It also permits to extract statistical values, like percentiles, or other ones for concrete algorithms and processes [18]. Unfortunately, despite histograms are really good at showing global features of images, they are not able to manifest local effects. Moreover, histograms are not capable to show the intrinsic relations of the image (i.e. edge transitions). For this reason and according to this work's scope, a tool which brings qualitative features will be needed.

## 3.2 Spectrum

The spectrum of a signal (or function) refers to a certain range of frequencies. To determine this range of frequencies, the Fourier transformation is applied to the signal. This transformation reveals periodicities in a function (or signal). In more detail, the output of this transformation is a sum of sine and/or cosine functions of different frequencies of the spectrum, each multiplied by a coefficient or weighing function (this sum takes the name of Fourier series for the first case and Fourier transform for the second one). The Fourier transform is used in numerous branches in mathematics, science and engineering. Specifically, in signal processing it is one of the most powerful tools for analyzing and processing. Therefore, in image processing it is also used because images can be interpreted as signals. Although the scope of this section is not to explain the mathematics of this operator in detail, some basies need to be introduced first. Thus, the one-dimensional Fourier transformation ($F(u)$) and its inverse are defined as:

$$F(u) = \int_{-\infty}^{\infty} f(x) \cdot e^{-j2\pi xu} dx \tag{3.2}$$

$$f(x) = \int_{-\infty}^{\infty} F(u) \cdot e^{j2\pi xu} du \tag{3.3}$$

Where $j$ is the imaginary number $\sqrt{(-1)}$. Extending these formulae to the two-dimensional case ($f(x,y)$) leads to:

$$F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y) \cdot e^{-j2\pi(ux+vy)} dxdy \tag{3.4}$$

$$f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u,v) \cdot e^{j2\pi(ux+vy)} dudv \tag{3.5}$$

According to image processing interests, the discrete version of the 2D Fourier transform and its inverse is needed. For a discrete two-dimensional function, $f[m,n]$ with $m$ in the range $[0, M-1]$ and $n$ in the range $[0, N-1]$, the pair of equations is:

$$DFT[k,l] = \frac{1}{\sqrt{MN}} \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f[m,n] \cdot e^{-j2\pi(\frac{km}{M}+\frac{ln}{N})} \tag{3.6}$$

$$IDFT[m,n] = \frac{1}{\sqrt{MN}} \sum_{k=0}^{M-1}\sum_{l=0}^{N-1} DFT[k,l] \cdot e^{j2\pi(\frac{km}{M}+\frac{ln}{N})} \tag{3.7}$$

with $k$ and $l$ in the ranges $[0, M-1]$ and $[0, N-1]$, respectively. Consequently, the output is a "map of frequencies" with the same resolution as the original image. However, in all those cases the transformation is from the real to the complex domain ($f : R \longrightarrow C$) and therefore the complex output can be represented with a sum of the real and the imaginary parts, or with a complex vector which has a modulus and a phase. Here the modulus will be used to represent the spectrum. The modulus reveals frequency components, although the main and most important information of the image is "stored" in the phase [19], but its representation is totally chaotic.

The modulus representation is a 3D plot, but it can be represented with a 2D image with resolution $M \times N$. The content of this image is $F(u, v)$ and, as similarly done in topography, the mentioned value is mapped with a color palette. Another change to enhance the modulus representation is the shift of the image in horizontal and vertical direction, which can be achieved due to exponential properties:

$$F\{f(x, y) \cdot (-1)^{(x+y)}\} = F(u - M/2, v - N/2) \tag{3.8}$$

$F\{\cdot\}$ represents the Fourier transformation of the content in between brackets. Particularly, $F(0, 0)$ is the energy (or mean value) of the modulus image. Once the shift is done, this point is moved from the corner to the center of the image. Thus, low frequencies are located in the center of the representation and high frequencies in the corners. Finally, the last processing step done for a better representation is a logarithmic scaling:

$$Mod = log(1 + |F(u, v)|) \tag{3.9}$$

Hence, high values (usually low frequencies) are scaled in a lower range and facilitates the contrast with low values (usually high frequencies). The reason to add 1 to the modulus before applying the logarithmic scaling lies into computer's exigencies (if $|F(u, v)| = 0$ the logarithm returns $-\infty$, what it is not possible to represent).
Now that the representation is defined, it is interesting to predict what can be seen in a spectrum. Abovementioned, the modulus representation is a "map of frequencies" of the image. Thus, rough textures and edges, whose are both high frequencies, can be identified in the corners of the frequency domain (see figure 3.2). Here, variations are manifested as edges as well, but oriented perpendicularly to the edges of the image domain (because it is the direction in which intensity changes).

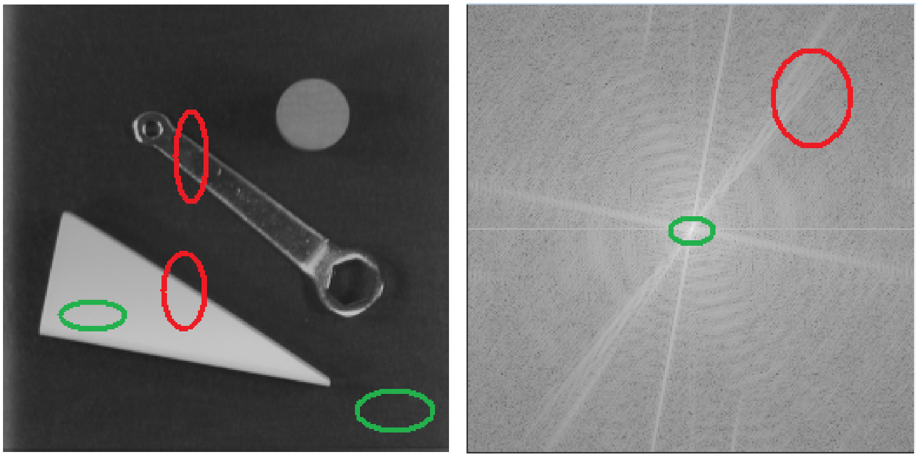


**Figure 3.2:** *Example of spectrum representation in a greyscale. Edge transitions and rough textures are located out of the center of the representation (red mark), while constant regions are easily detected in the center of the spectrum (green mark).*

Spectrum analysis is good at disclosing scene's geometry from color images. However, the texture in terms of high frequency, acts like additive noise and leads to vague interpretations. Unlike the histogram, the spectrum is able to reveal intrinsic features of the image, but it is not consistent enough to conclude if, afterwards, an image has been changed or not. With respect to analysis and processing of MVD the task therefore is to find a tool capable to reveal general features (as the histogram can) and also able to reveal intrinsic characteristics (as the spectrum does), separating the nature of high frequencies (rough textures and edges).

## 3.3 Correlation Histogram

In order to show similarities and differences between color and depth components of MVD, a well-suited method, called correlation histogram, is presented. As its name suggests, this tool shows the correlation between pixels. Moreover, it represents a two-dimensional histogram, that gives an idea of how the correlated information is distributed.

Unlike classical histograms, correlation histograms use an array of bins $H(k, l)$. Here, each bin represents the number of specific pixel pair combinations with grey values $k$ and $l$. As an example, if the bin $H(22, 33) = 102$, the image being analyzed contains 102 pixel pairs with values $k = 22$ and $l = 33$. As the statistical analysis covers the video luminance or the depth information, both with 8 bits per pixel, an array of 256x256 bins is necessary. Figure 3.3 illustrates the two types of correlation histograms (called spatial and temporal) used for analyzing the video and the depth component of the MVD sequences. Note, that references taken are orthogonals (the scalar product of these three vectors is equal to 0) as it is represented in figure 3.3.

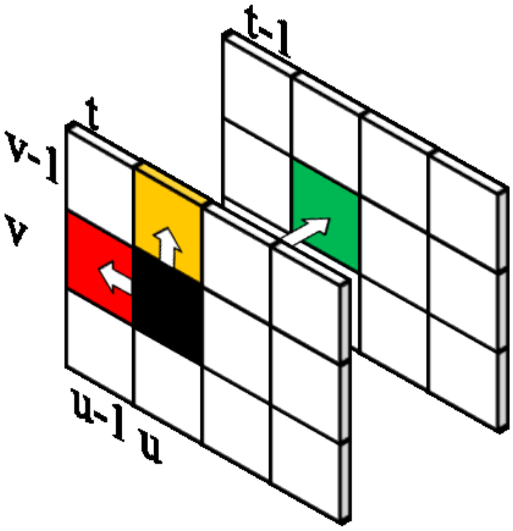


**Figure 3.3:** *Two types of pixel relations used for correlation histograms: spatial (red and orange) and temporal (green)*

To represent the correlation histogram a 3D plot would be needed. However, as it is similarly done with the spectrum representation, an image of 256x256 pixels is enough (one pixel corresponds to one bin). Then each value $H(k, l)$ is represented by a pixel with an intensity value according to a color map. Again, a logarithmic scaling is done in order to show a better contrast between the most and the least repeated values:

$$H^{'}(k, l) = log(1 + H(k, l)) \tag{3.10}$$

In the following two sections, these two types of correlation histogram are described in detail:

### 3.3.1 Spatial correlation histogram

In the case of spatial correlation histogram two orthogonal references are chosen (see Figure 3.3). This leads to two independent correlation histograms, one for "horizontal" and one for "vertical" correlation, which are defined as:

$$H_s^v(k,l) = n_{sv} \tag{3.11}$$

$$H_s^h(k,l) = n_{sh} \tag{3.12}$$

with $n_{sv}$ the number of pairs of pixels with $I(u,v,t)$ having the grey value $k$ and $I(u,v-1,t)$ having the grey value $l$ and $u$ and $v$ being the coordinates of image $I$ at time instant $t$. Similarly, in equation 3.12, $n_{sh}$ is the number of pairs of pixels with $I(u,v,t)$ having the grey value $k$ and $I(u-1,v,t)$ having the grey value $l$. Once the image is analyzed, the resulting vertical and horizontal spatial correlation histograms are averaged in order to obtain a unified data representation:

$$H_s(k,l) = \frac{H_s^v(k,l) + H_s^h(k,l)}{2} = n_s \tag{3.13}$$

Note that, the first row and the first column of an input image have no vertical and horizontal references, respectively, and are therefore not taking into account for analysis.

The chosen combination of pixel pairs (one above and one left) can be discussed. The main reason of this combination is due to that it reveals intensity changes in both directions. In this case, both references are orthogonals and by combining both in a same representation it is ensured that step pixel transitions in any direction are detected. Moreover, both pixel references are at the minimum euclidean distance regarding spatial variations (i.e. by chosing diagonal references the distance between pixels being analyzed is $\sqrt{2}$). Another possibility would be to join both orthogonal references into a unique spatial correlation histogram. Hence, the correlation histogram would be:

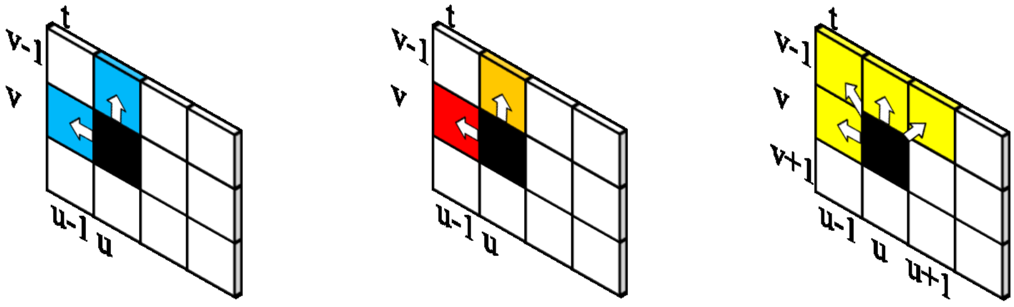
$$H_s(k,l) = n_s \tag{3.14}$$

in which $n_s$ is the number of pairs of pixels with $I(u,v,t)$ having the grey value $k$ and $1/2[I(u-1,v,t) + I(u,v-1,t)]$ having the grey value $l$ (see figure 3.4(a)). However, this idea do not represent the exact pixel transition in the output. As an example, a vertical transition with $I(u,v,t) = 100$, $I(u-1,v,t) = 50$ and $I(u,v-1,t) = 100$ results $k = 100$ and $l = 75$ (see figure 3.5). Therefore, the transition from value 50 to value 100 is represented in the output with a point located in coordinates $(100, 75)$.

Finally, another discarded combination is to set the "coding-neighborhood" as the second element ($l$) of the pair (see figure 3.4(c)):

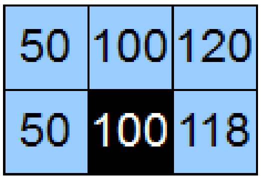
$$H_s^{cod}(k,l) = n_{sc} \tag{3.15}$$

with $n_{sc}$ the pair of pixels with $I(u,v,t)$ having the grey value $k$ and $1/4[I(u-1,v,t)+I(u,v-1,t)+I(u-1,v-1,t)+I(u+1,v-1,t)]$ having the grey value $l$. This combination introduces a new concept of diagonal pixels. However, this method has the problem that averages more adjacent pixels than the elected one. Taking four adjacent pixels distorts intensity changes. Following the example presented in figure 3.5, here $k = 100$ and $l = 80$. Hence, values are not mapped in the output according to pixels transitions (as it occurs with equation 3.14).



(a) (b) (c)

**Figure 3.4:** *Different combinations for a spatial correlation histogram*



| | | |
|---|---|---|
| 50 | 100 | 120 |
| 50 | 100 | 118 |

**Figure 3.5:** *Example of a pixel and 5 pixel neighbors. Black marked pixel corresponds to $k$ element of the pair.*

As aforesaid, histogram analysis lacks of intrinsic relations between pixels. Additionally, a spectrum does not map rough textures and edges in different parts of the modulus representation. In contrast to that, the spatial correlation histogram has the ability to show intensity changes from a pixel to its previous ones (in both orthogonal directions), which means, edges in the scene can be well detected. With the aim to allow for a better understanding of the output, some examples are presented in 3.6.

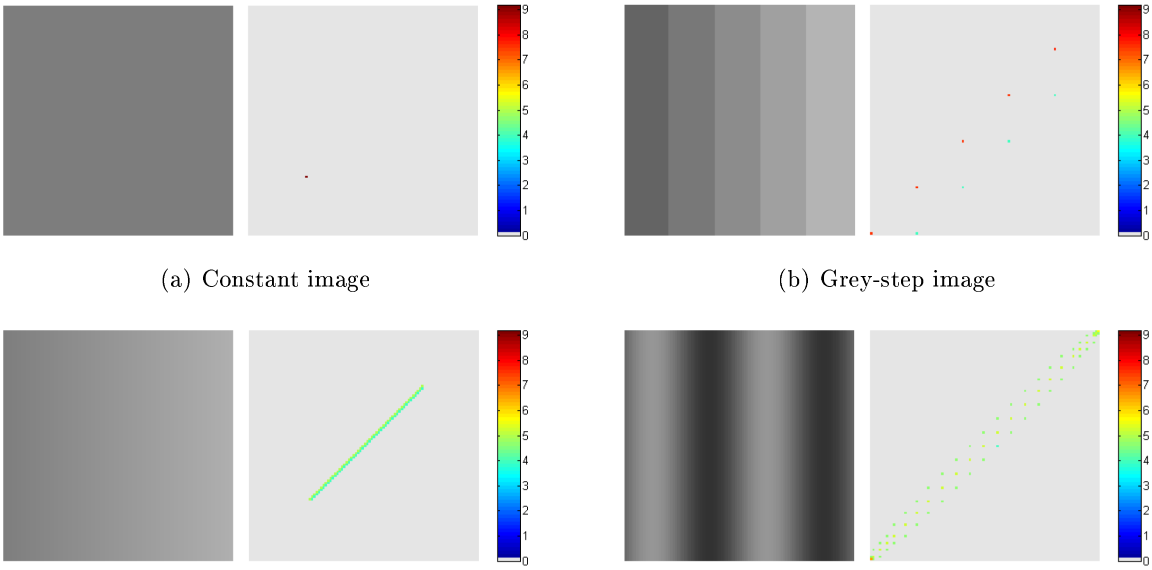
Four synthetic images are used to investigate the spatial correlation histogram output. All have a resolution of 100x100 pixels and their values are in the range $[0, 100]$, leading to a correlation histogram with 100x100 bins[1].

The first and the most simple case is an image with a constant value of 25, as shown in figure 3.6(a). As there is no spatial pixel variation ($I(u, v, t) = I(u-1, v, t)$ and $I(u, v, t) = I(u, v-1, t)$ $\forall u, v$), the output is $H_s(k, k) = 100 \times 100 = 10000$ iff $k = 25$ and equal to zero for any other case.

Similarly, for the step function image in figure 3.6(b) the output has red points along the diagonal (consequence of constant regions with different values) and blue points below it, resulting from horizontal edges (i.e. from 0 to 20, from 20 to 40, etc.). In the case of having softer transitions that keep constant in one direction, they are again mapped onto the diagonal, while grey changes are located closer to this line. In particular, figure 3.6(c) shows the result of a horizontal gradient image (with a range of $[26, 75]$) where maximal pixel variations are of one pixel difference and therefore a line is located one pixel below the diagonal[2].

[1] For a better printed version visualization, figures had been re-scaled to the range [100,200]

[2] Problems at displaying could not distinguish a yellow line (diagonal) and a green line (line below the diagonal)



(a) Constant image

(b) Grey-step image

(c) Gradient image

(d) Sinusoid image

**Figure 3.6:** *Examples of four spatial correlation histograms applied to basic images (constant (a), grey-step (b), gradient (c) and sinusoid (d)).*

Finally, a horizontal sinusoid presents no changes in vertical direction, which leads to diagonal values different to 0. and the horizontal variations are represented by values above and below the diagonal.

In accordance with the given definition of the pair of pixels and the presented examples, the spatial correlation histogram representation can be differentiated in the following groups (see figure 3.7):

1. Diagonal. In this part of the output the pairs of values are similar. That means that soft textures are represented along and around the one-pixel-width diagonal line. Flat regions are located here in the case of depth maps.

2. Quadrant I. This subset is the one corresponding to low values. On one hand they will mean dark regions (color) and on the other one background areas (depth).

3. Quadrant IV. Dual to the quadrant I, high values belong to this part. Bright luminance values (color) are located here and foreground regions (depth) or objects will be here represented.

4. Quadrants II and III. Both quadrants are set in the same group because both represent the interaction of low and high values (and vice versa). In the case of color component it means rough textures while for the depth it means interaction between foreground and background (Quadrant II) and vice versa (Quadrant III).

Due to the orthogonal references taken, edges oriented in any direction can be detected by a spatial correlation histogram. As a kind of histogram, it is good at showing how the information is distributed (in this case, the correlation of the pair of pixels). Notwithstanding, this tool does not represent the edges orientation. It is able to reveal changes of grey intensities but not the

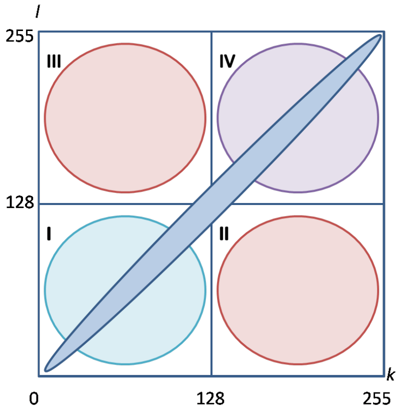


**Figure 3.7:** *Generic output of a Correlation Histogram with four quadrants plus a diagonal region.*

direction of them nor their location in the scene. Consequently, this tool is non-injective (several different inputs can lead to a same output), although in practice would be nearly impossible to find a pair of natural sequences with the same output.

### 3.3.2 Temporal correlation histogram

Unlike the spatial correlation histogram, where pixels pairs belong to the same image, the temporal correlation histogram needs two frames to be computed. Formally, it is defined as:

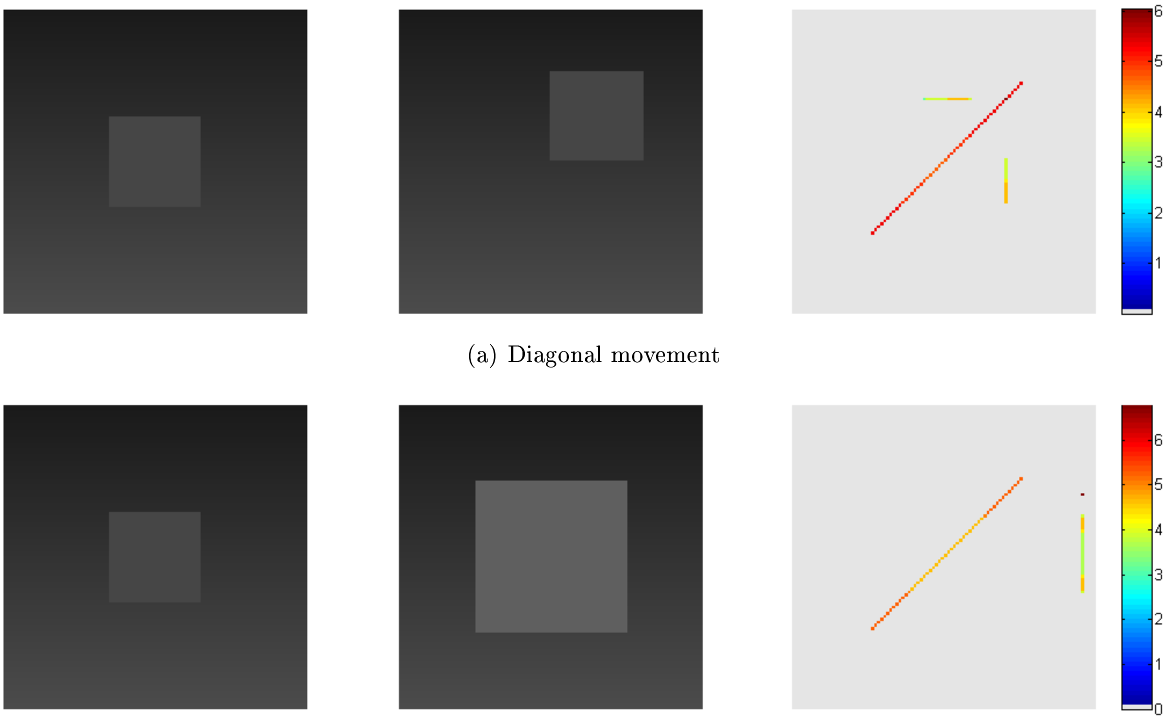
$$H_t(k,l) = n_t \tag{3.16}$$

with $n_t$ is the number of pairs of pixels with $I(u,v,t)$ having the grey value $k$ and $I(u,v,t-1)$ having the grey value $l$ and $u$ and $v$ being the spatial coordinates for each frame (see figure 3.3). Note, that the first frame lacks of a previous reference and is therefore not taken into account for analysis. Thus, if a sequence has $F$ frames, $F-1$ temporal correlation histograms can be computed from them.

The motivation to chose this combination is because it is the closest temporal dependency. In a discreet domain, it is the minimum euclidean distance regarding temporal dependency.
Another context for a possible use of this tool is with coded sequences. The intention would be to analyze the differences between frames in the order they are transmitted, instead of the order they are represented. A good test would be with a sequence encoded with an IPPP[3] structure, where temporal references are in one direction. Hence, the result of such temporal correlation histogram would show the degree of change between pairs of images that have been used for predicting.

Spectrum analysis is good in detecting constant areas, but it does not give a clear idea of their consistency. In contrast, temporal correlation histograms can be used for detecting static regions of a sequence and determine if they are consistent or not. With the aim to allow for a better understanding of the output, some examples are presented in figure 3.8.

[3] In this sequence frames are coded in the Intra mode (I) and Predicted mode (P)



(a) Diagonal movement

(b) Depth movement

**Figure 3.8:** *Examples of two different temporal correlation histograms. From left to right, frame at instant $t-1$, frame at instant $t$ and temporal correlation histograms outputs. (a) represents a diagonal movement (constant depth value). (b) represents a depth movement (variation in the $Z$ coordinate).*

Both figures represent the output of a temporal correlation histogram for synthetic image pairs. The total range of grey values is $[0, 100]$ and therefore the resolution of the correlation histogram is 100x100 bins (one bin per pixel). For both examples at instant $t-1$, a box (with value 70 and a size of 30x30) is centered in a 100x100 pixel resolution scene with a vertical gradient in the range $[26, 75]$.
For the first case, the box is displaced from the center to the top-right corner (changing its $U$ and $V$ coordinates) but remaining at the same depth (thus, depth values belonging to the box keep constant). The output shows the overlapped regions, where the diagonal corresponds to the coincidence of the background and a quarter of the box with itself, and the other two lines are the interaction of the box and the background.
For the second one, the box size and depth value are increased, representing the movement from a farther to a closer position (with depth value is 95). Looking at the output, those values that correspond to superimposed pixels that keep constant in time (only the background) can be seen in the diagonal. The vertical line ($x = 95$) correspond to the area of the box that is covering the background and the red point is the overlap of the box with itself, what means $H_t(95, 70) = 30 \times 30 = 900$.

Note, that for a temporal correlation histogram, a sequence with no motion means that $I(u, v, t) = I(u, v, t-1)$ and therefore $H_t(k, l) = 0$ if $k \neq l$ and $H_t(k, k) = n_t$ with $k$ in the range of grey values of the given image $I$. In this case, all values are concentrated to the diagonal. In both examples in figure 3.8 the background is temporally consistent, as low values keep on the diagonal and are not dispersed.

In summary, the temporal correlation histogram output can be divided in the following groups (see figure 3.7):

1. Diagonal. It contains the values which did not change, meaning constant values (all those $I(u, v, t) = I(u, v, t - 1)$). Thus, constant foreground and background is represented by this line (depth) and also those regions that do not have a change of luminance (color).

2. Quadrant I. Subset corresponding to low values. Here the temporal variation of the background in depth maps is identified. In the case of color component, changes in pixels of low luminance are here.

3. Quadrant IV. Subset corresponding to high values, what means temporal variations of the foreground. Similarly to quadrant I, changes of bright pixels are represented here for color component.

4. Quadrants II and III. According to depth maps, it is the interaction between background and foreground. Objects comming from the back to the front of the scene (by varying, at least, the $z$ component) belong to quadrant II, while objects moving from the front to the back belong to quadrant III. Note, that effects of moving objects in horizontal or vertical direction can not be predicted, as it depends on background's content. On the other hand, pixels that change from dark to bright values correspond to quadrant III and pixels that change from bright to dark values are located in quadrant II.

As aforesaid, this tool is useful for detecting temporal consistency. In the case of depth maps, temporal correlation histograms can be used for analyzing background and foreground regions. Note, that this analysis must be done for certainly static background and foreground. Otherwise, some false detections could be comitted. It means that the data that is being analyzed must be known when analyzing background and foreground areas (basically if there is motion or not in those regions). All sequences used in this work present a static background in general. However, constant areas (in a spatial sense) like the floor or the background present sharp changes of depth intensities, when these changes should be ideally a static gradient of a grey level range (see Ballet or Breakdancers sequences). Moreover, even if these changes of depth intensities are sharp, they must be static areas (in a temporal sense) as there is no motion in the background and the floor. As it happens with the spatial case, this tool is also non-injective.

## 3.4 Analysis results

A total set of six MVD sequences (see Appendix), with five natural sequences and one synthetic, have been analyzed. Depth maps of natural sequences are obtained from color information with different methods. However, they present the inherent imprecissions which are typical of the estimating algorithms. Hence, the synthetic sequence will be used as reference because it contains ground truth depth information. All results are equalized regarding different resolutions, number of cameras and number of frames of the test data sets. Table 3.1 summarizes these data set properties:

| Sequence | Type | Num Cameras | Num Frames | Resolution |
|---|---|---|---|---|
| Ballet | Color | 8 | 100 | 1024x768 |
| | Depth | 8 | 100 | 1024x768 |
| Book | Color | 3 | 100 | 1024x768 |
| | Depth | 3 | 100 | 1024x768 |
| Breakdancers | Color | 8 | 100 | 1024x768 |
| | Depth | 8 | 100 | 1024x768 |
| Horse | Color | 2 | 140 | 1920x1080 |
| | Depth | 1 | 140 | 1920x1080 |
| Newspaper | Color | 2 | 200 | 1024x768 |
| | Depth | 2 | 200 | 1024x768 |
| Synthetic | Color | 11 | 250 | 1024x768 |
| | Depth | 11 | 250 | 1024x768 |

**Table 3.1:** *Summary of the properties of the test data set.*

In this section, 3 of 6 MVD sequences have been chosen for analyzing. These three sequences are the Breakdancers, the Horse and the Synthetic. As previously mentioned, the Synthetic is taken because it is a reference due to ground truth depth data. Breakdancers and Ballet have been computed with the same algorithm and therefore it is interesting to take one of them for analyzing (Breakdancers shows more characteristics than the Ballet set). The Horse sequence presents features that are totally different to the rest of sequences and it is also the only one with different resolution. Hence, the chosen sequences are diverse by presenting different number of frames, cameras and resolutions.

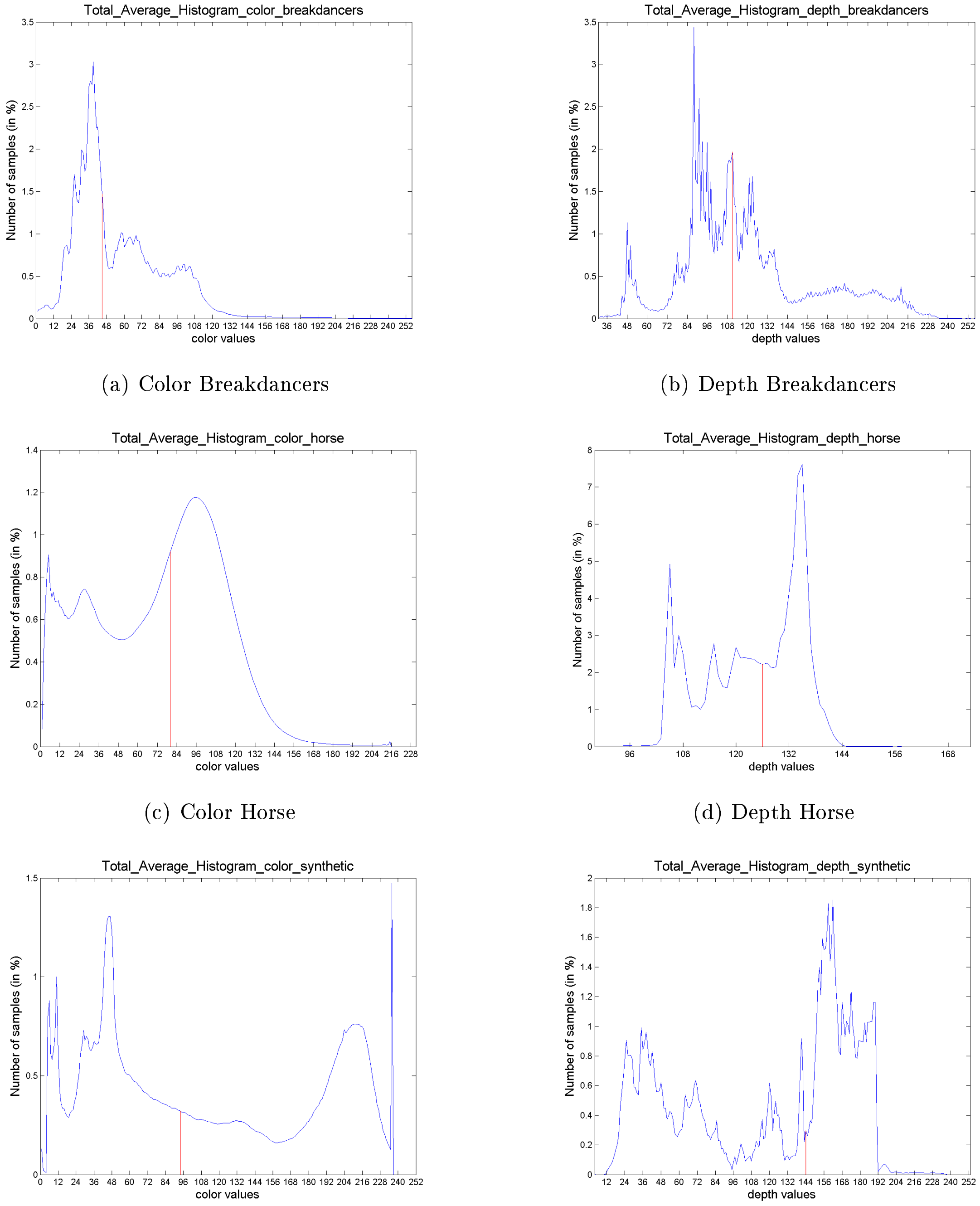
The methods explained in this chapter have been used to analyze the data set summarized in the table above. In order to show statistical differences between color and depth information of the MVD representation, the histogram analysis is the first one applied and discussed in 3.4.1. Then, the spectrum can be found in 3.4.2. Finally, the new correlation histogram method is used to analyze the data set and the spatial as well as temporal results are discussed in 3.4.3.

### 3.4.1 Histogram

First, histogram results for video and depth are analyzed in order obtain insights into similarities and differences of the color and depth characteristics. As mentioned in section 3.1, this method is useful to reveal statistical features of the given data set. Hence, histograms are good for showing general features, such as the range[4] of values and their distribution. Figure 3.9 shows the histograms obtained for the 3 of the 6 MVD sequences, distinguishing between color and depth.

Having a quick look at the results confirmes that color component (left) have smoother curve profiles than depth maps (right). This difference can also be recognized in the rest of sequences (see A.3). What is more, there is no strong correlation between color and depth histograms of the same sequence. Consequently, regarding the approximated distribution of probabilities, it is

[4] The range is defined as the interval composed by the first and the last bins of a histogram that are different to zero.



(a) Color Breakdancers

(b) Depth Breakdancers

(c) Color Horse

(d) Depth Horse

(e) Color Synthetic

(f) Depth Synthetic

**Figure 3.9:** *Histograms of 3 of the 6 MVD sequences, corresponding to color and depth.*

not beneficial to extrapolate values from one component to the other, confirming their different nature. It is attested that a value of one component (i.e. color) cannot be extrapolated from the other (i.e. depth) because both values differ on their representation (luminance is incorrelated with scene geometry representation).

Now analyzing color and depth components separately, the presented sequences almost use the whole range for the color. Specially for the horse and the synthetic sets, some bright values are

not being used. The shape of the histogram, or the approximated distribution of probabilities, are also different between them. Breakdancers present a high concentration in dark values (from 0 to 128 approximately) distributed in three main bell shaped regions with diverse visible peaks. As an example, this range of values correspond to dark clothes or the wall of the background. The Horse sequence also presents three main bell shaped regions which are concentrated in the range from 0 to around 144, but just the first one has more than one peak. In this case, this range of values mainly correspond to the figure of the animal. By contrast, the Synthetic data set presents four main bells plus four sharp peaks (located at around values 0, 8, 12 and 236). Here is difficult to identify the source of these regions because this sequence uses a big set of colors. However, luminance values in this sequence are more homogeneous by covering the range [11, 236] and having a median value of 94, while for the Horse and the Breakdancers the median is lower (80 for the first and 45 for the second). For these two sequences, the 95% of the values covers the ranges [0, 110] (Breakdancers) and [0, 129] (Horse). All these statistics can be found in percentile tables A.1 and A.2.
For the rest of the sequences, Ballet sequence presents a bigger concentration in the middle values with two main regions, with more than 85% for the main bell shaped, while the other two (Book and Newspaper) are using almost the whole range with a more uniformly distributed probability of occurence, which is ranging between 0.2% and 0.4%.

Regarding the depth component, five of the six provided depth maps use the whole range of values, which is not the case for the Horse sequence, which uses a small dynamic range (approximately [90, 157]). As afore mentioned, this data set uses a different estimating algorithm than the Breakdancers. In this sequence is easy to differentiate between foreground (figure of the animal and bottom of the image) and the rest. Having a look at the histogram, the biggest peak corresponds to the horse and the part of grass that is closer to the camera (bottom of the image). The Breakdancers sequence presents a high concentration in low and middle values with sharp peaks, representing a 80% for the range $[0, 150]$ (see table A.2). This range of values correspond to background areas (wall and young men standing up). From this value to the brightest, values tend to be distributed uniformly. Contrary, the Synthetic sequence accumulates almost 50% of the distribution in the interval [144, 192] with sharp peaks. Low and middle values have a broad range but they are not distributed uniformly as Breakdancers bright values. A good measure to highlight the "concentration" of values regarding the used range is presented:

$$\eta = \frac{\Pi(95) - \Pi(5)}{range} \tag{3.17}$$

where $\Pi(x)$ is the percentile $x$ of the histogram and assuming the term "concentration" as the 90% of the values, excluding a 5% of the low and a 5% of the high values. Hence, the concentration of these 3 sequences is summarized in the following table:

| **Sequence** | $\Pi(5)$ | $\Pi(95)$ | **Range** | $\eta$ |
|---|---|---|---|---|
| Breakdancers | 54 | 200 | 226 | 0.646 |
| Horse | 105 | 137 | 67 | 0.478 |
| Synthetic | 26 | 187 | 237 | 0.679 |

**Table 3.2:** *Concentration of used depth values*

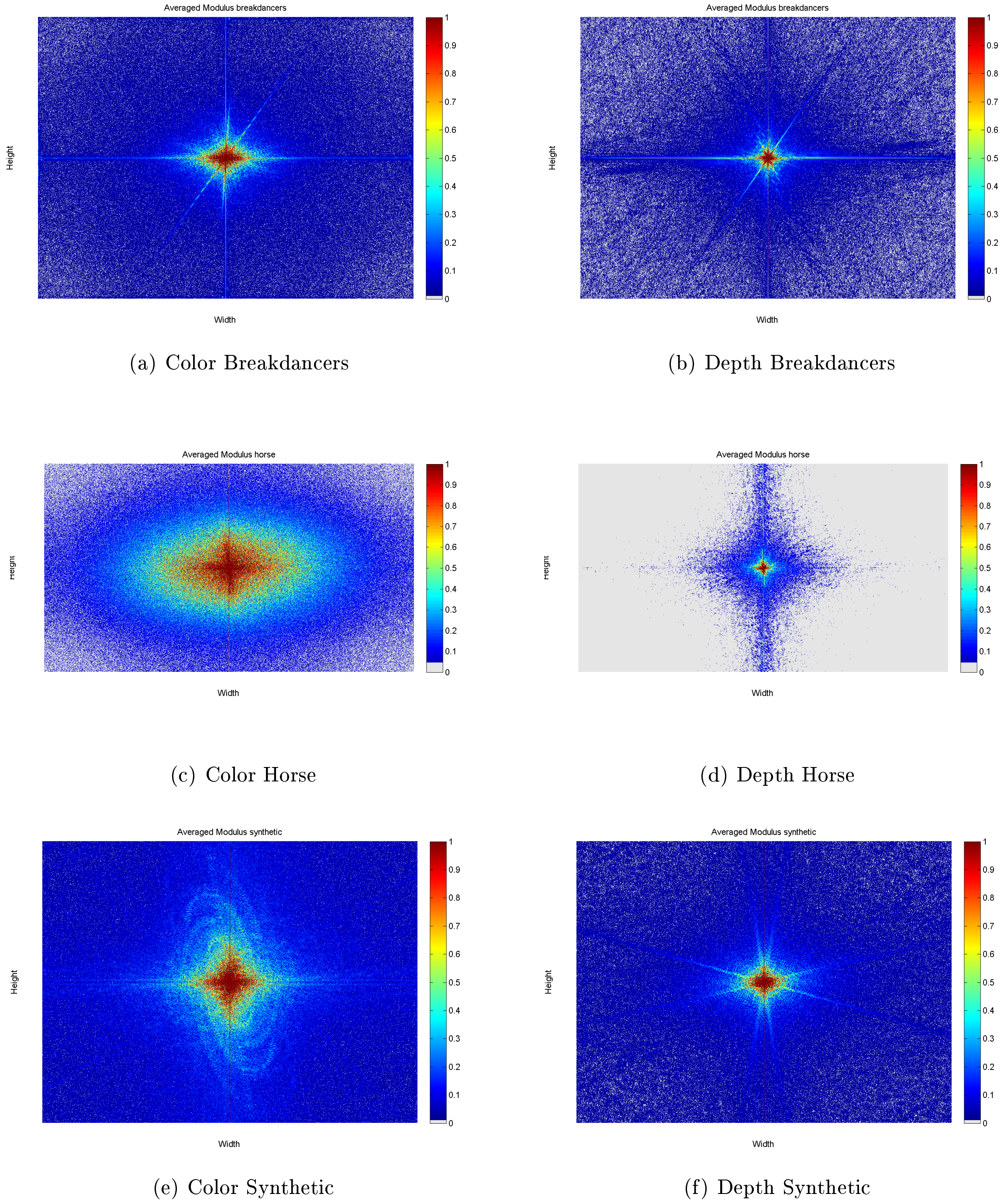
It is worth mentioning that although the Horse set uses a small dynamic range, its concentration does not differ in exceed from the other two sequences. As depth maps give an idea of the scene geometry, their essential target is to be an accurate representation for the algorithm that renders the stereo color video pair. In summary, the quality of the depth map does not depend on its range of values if the information is well used later and provides a good rendered stereo pair.
The other 3 MVD sequences have a concentration of 0.728, 0.705 and 0.617 (for Ballet, Book and Newspaper) which are similar to the Breakdancers and the Synthetic sequences. However, note that, for Book and Newspaper sets certain available values are not being used, what prevents to relate the idea of "utilization" of available values with the afore defined term "concentration".

### 3.4.2 Spectrum

While histogram analysis is good for deriving statistical characteristics, the spectrum representation allows to observe some intrinsic dependencies. The main goal of analyzing from the standpoint of frequency domain is to highlight similarities between edges present in the color component and edges from silhouettes in the depth component. As mentioned in section 3.2, just the modulus is taken into account for the analysis, as it is impossible to derive conclusions from the phase. According to the modulus representation technique explained in section 3.2, figure 3.10 shows the modulus of both components of 3 MVD sequences.

First of all, Breakdancers and Synthetic sequences have the same resolution while the resolution of the Horse sequence is bigger (see table 3.1). For both sequences, there is coincidence of line directions present in color and depth outputs, although it is clearer for Breakdancers than the synthetic one. However, for the Horse set, it is really difficult to attest similarities between color and depth spectrums. For the remaining sequences, whose modulus representation can be found in A.4, there is also coincidence of line patterns between color and depth components. Note, that there is no exact match of direction and intensity of line patterns in color and depth outputs, though it gives an intuitive idea of spatial variations. While depth maps present a high concentration of low frequencies, color tends to have a spreader frequency range (color video has more texture than depth maps).

Having a look at the color video spectrum of the MVD sequences, the Synthetic set presents non-linear patterns, unlike it happens to the rest of the data. It also has weak lines, but what makes it different to the rest are the spirals around low frequencies. These spirals are due to elliptical and round shapes in the scene (such as the tables, or the railtracks on the floor, or the texture of the couch and courtains). In this sequence, low frequencies in both directions (vertical and horizontal) are distributed similarly. However, a high frequency response can be seen for constant values regarding horizontal variation ($|F(0, f_y)|$ with $f_y \in [0, 767]$ is high). In the same way, the Horse sequence presents a high amount of frequency response for $f_x = 0$, indicating that intensity changes occur in the vertical direction (i.e, the beginning of the trees, changing from grass to forest). Nevertheless, it has the widest frequency range. On the other hand, the Breakdancers sequence has a high concentration of low frequencies and a horizontal variation bigger than the vertical one. The dominating line corresponds to the transition of the stairs and the railing (variation in pixel intensities through the normal vector of this line). Oppositely, Ballet has more variation in the vertical direction than the horizontal one. The other two sequences, Book and Newspaper, have a wide range of low and middle frequencies.



(a) Color Breakdancers

(b) Depth Breakdancers

(c) Color Horse

(d) Depth Horse

(e) Color Synthetic

(f) Depth Synthetic

**Figure 3.10:** *Modulus representation of 3 of the 6 MVD sequences, corresponding to color and depth.*

Nevertheless, just Newspaper has less presence of high frequencies with a frayed shape on both corners.

As previously said, depth maps lack rough texture and consequently their spectrum has less high frequencies. This fact can be seen in the depth spectra of Breakdancers and Horse sequences, where high frequencies are not present. In the case of the Horse set, is worth mentioning that the amount area with no frequency (grey) is related to the dynamic range of the depth map. In other words, the wider the dynamic range, the more possibilities of high frequencies. As an example, if the dynamic range is low, the amplitude of the sinusoid will be low, although its frequency would be high. However, the spectrum of the synthetic sequence reveals high frequency components. The cause of these high frequencies can be related with gradient areas, like the floor, whose Fourier series results an infinite sum of sinusoids.

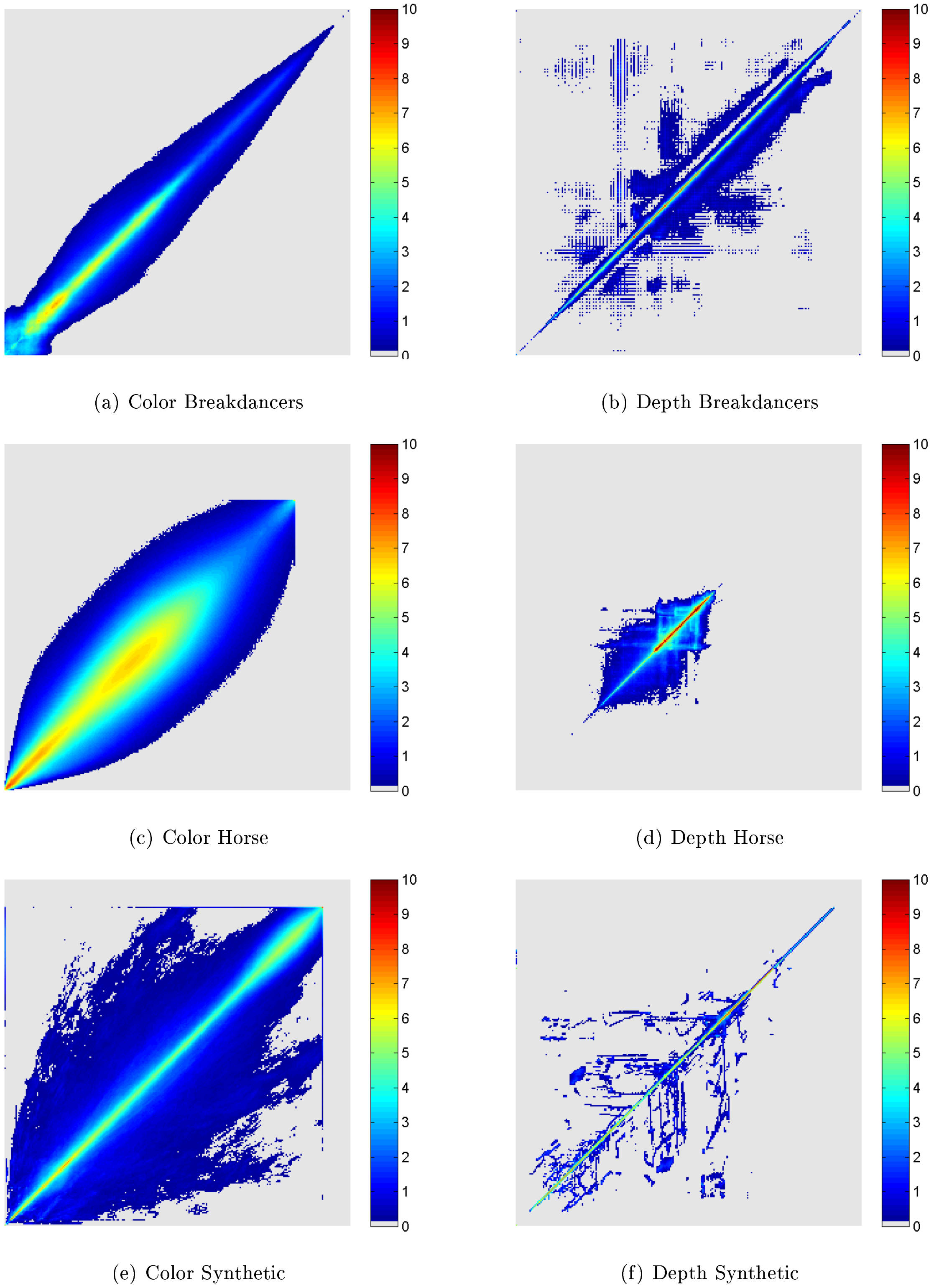
The remaining two sequences, Book and Newspaper, have less high frequency components than the ballet. For this last one, $|F(0, f_y)|$ and $|F(f_x, 0)|$ are both high, meaning constant regions regarding vertical and horizontal variations.

### 3.4.3 Correlation Histograms

Both presented methods, spatial and temporal correlation histograms, are finally used to show the main differences between color and depth components of MVD. The pairs of pixels that have been analyzed are the ones explained in subsections 3.3.1 and 3.3.2. As previously explained, spectrum analysis does not illustrate strong correlations between edges present in depth and color components. Moreover, it is difficult to decide, if the spectrum of an unknown input belongs to depth or to color component. Some general features of color and depth modulus representation have been highlighted. Nevertheless, these features are not relevant enough to analyze the MVD sequences in more detail that will be used subsequently.

First, spatial correlation histograms are analyzed for both MVD components and in a next step temporal corration histograms. Breakdancers, Horse and Synthetic outputs of the aforesaid spatial correlation histogram tool are illustrated in Figure 3.11. Note, that bins far from the diagonal represent sharp and bins near the diagonal represent flat pixel transitions, respectively. Looking at the results, color outputs show the typical and well-known characteristics with a more compact distribution along the diagonal, which is directly related to soft textures. On the other hand, depth results are considerably different, as depth correlation histograms are much more frayed with isolated areas, hence representing sharp edges between foreground and background objects. It is confirmed again that for the color MVD component, values change softly from a pixel to its neighbours, while depth has high pixel transitions. What is more, wide bell shaped regions in histogram outputs are represented as wide areas around the diagonal and sharp peaks are represented as thin areas around the diagonal with high values (tending to red).

Considering both components separately, the color output of the MVD set presents a generic pattern for all natural sequences with a high concentration around the diagonal, while the Synthetic sequence has a wider and more frayed shape around the diagonal.
Related to histograms again, the dynamic range of the sequence can be seen here as well. Specially for the Horse and the Synthetic ones, the luminance gamut is perfectly delimited by a corner with a "saturation" value, which is the upper extreme of the interval.
The remaining sequences, which can be seen in A.5, present the described characteristics. Note, that the range of values is practically the maximum for the Book and Newspaper sequences, but not for the Ballet, as it was also commented in subsection 3.4.1.



(a) Color Breakdancers

(b) Depth Breakdancers

(c) Color Horse

(d) Depth Horse

(e) Color Synthetic

(f) Depth Synthetic

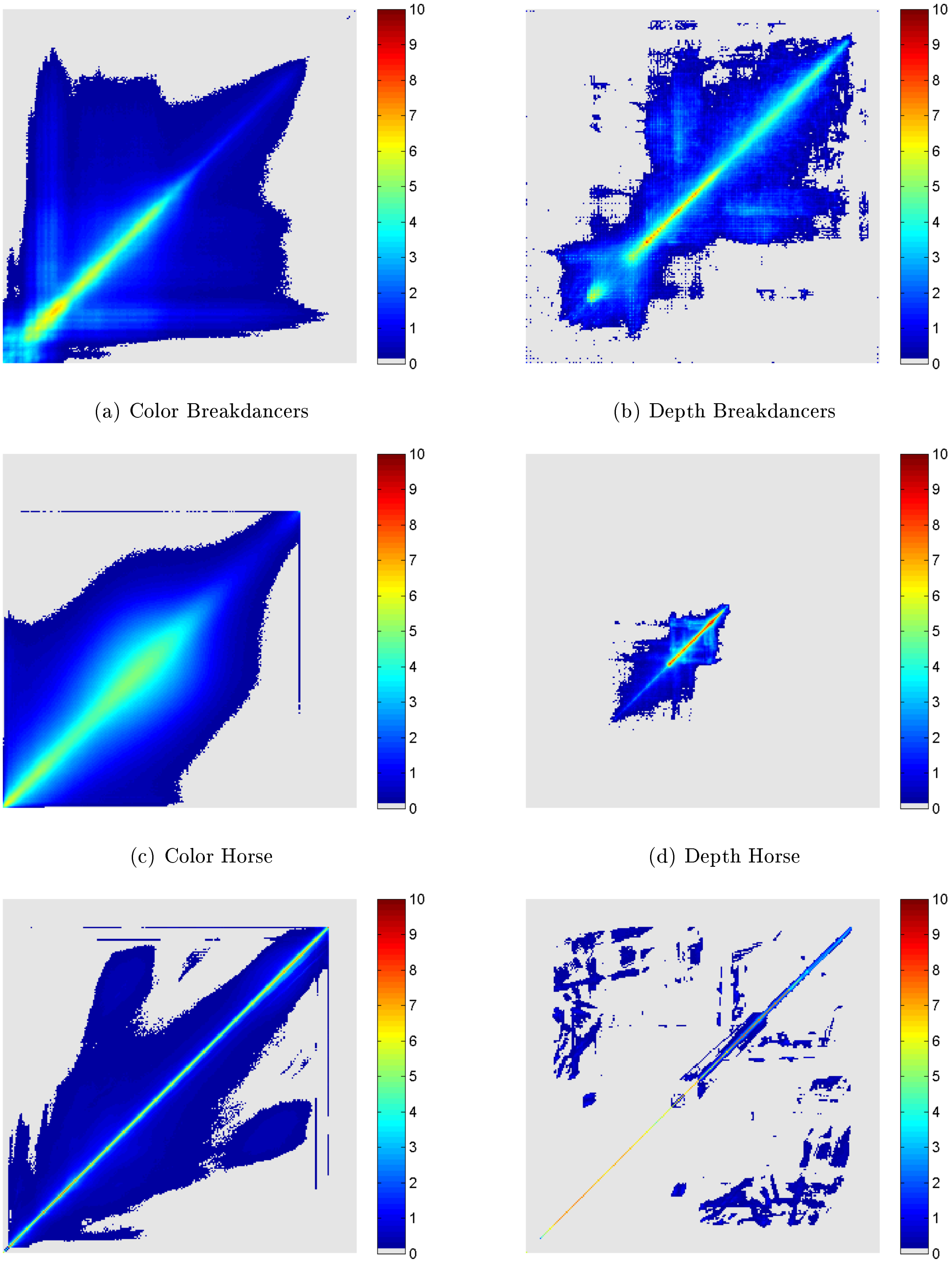
**Figure 3.11:** *Spatial Correlation Histograms of 3 MVD sequences, corresponding to color and depth.*

Refering to the examples in figure 3.6, depth maps should present characteristics comparable to the constant and the grey-step images (figures 3.6(a) and 3.6(b)). Hypothetically, spatial correlation histograms should present a high density of values along the diagonal and its surrounding bins (due to gradient regions as well) and some clear vertical and horizontal "lines" caused by having edges in the scene. This is the case for the Synthetic sequence describing the suposed behavior perfectly. However, the results of natural sequences, like the Breakdancers or the Horse, are frayed with isolated areas. Remember that the desired ouput are linear patterns. Here, the inherent errors and imprecisions of depth estimation are relevant. Nevertheless, it is important to highlight the symmetries around the diagonal in all cases (due to entire shapes in front of the background). As an example, steep pixel transitions from background to foreground (left to right or bottom-up) lead to points in the second quadrant and transitions from foreground to background lead to points in the third quadrant. Mathematically, it can be understood as $H_s(k, l) = n_A$, where $k$ and $l$ are the grey values of foreground and background, respectively. In the other sense, it would be $H_s(l, k) = n_B$ where $k$ and $l$ represent the afore mentioned grey values. Note, that it is difficult to have $n_A = n_B$ regarding all sequences, because of a non-constant background and distorted silhouettes. As a comment, take notice of the range of values that are being used for depth maps, a statistic that can be subtracted directly with this tool instead of using a complementary histogram.
The rest of the sequences (figure A.5) also show the described characteristics. However, Book and Newspaper sequences do not use some available values (as it was highlighted in subsection 3.4.1) and do not give a clear idea of line patterns and/or areas (because of their discontinuities in values).

Finally, temporal correlation histograms are used to analyze the MVD data set. As explained in subsection 3.3.2, the taken pair of pixels is a frame and its preceeding. Figure 3.12 illustrates the output of this tool for Breakdancers, Horse and Synthetic sequences. Spatial correlation histograms for color component present more compact distribution along the diagonal than temporal correlation histograms. However, the Synthetic sequence is an exception: it has wide areas around the diagonal in both outputs. This is due to the presence of a big set of colors. In the case of depth component, the characteristic differences are essentially the same. Temporal correlation histograms tend to have wider regions and less isolated patches.

Basically, the characteristic differences are the same as for the spatial correlation histograms. As it could be foreseen, temporal correlation histograms for the color component have a major concentration along the diagonal and its surrounding. For the depth component a straight diagonal line with no distortion in the first quadrant might be expected. As explained in subsection 3.3.2 and according to the sequence description in the appendix, background is suposed to be static. Having a look at figure 3.12, this behavior can be observed for the Synthetic sequence, but not for the natural ones. Again, the static background is not well estimated and therefore depth maps have temporal inconsistencies. Note, that in both cases there is a high level of symmetry relative to the diagonal.

Analyzing the outputs per component, temporal correlation histograms for the color component have some common characteristics. As previously said, values tend to be concentrated along and around the diagonal. On the other hand, sequences with more motion (i.e. Breakdancers) than others (i.e. Synthetic) present wider areas with more intensity. Nevertheless, the Horse sequence (slow motion) presents a wide result. Remember that this sequence is the only one



(a) Color Breakdancers

(b) Depth Breakdancers

(c) Color Horse

(d) Depth Horse

(e) Color Synthetic

(f) Depth Synthetic

**Figure 3.12:** *Temporal Correlation Histograms of 3 MVD sequences, corresponding to color and depth.*

that has been recorded outdoor. Therefore, effects like the wind can vary rough textures in a temporal sense (i.e. the grass or the hair of the horse). In this case, a pixel that belongs to a rough textured area can get different values due to external causes. Consequently, temporal correlation histogram gives an intuitive idea of the grade of motion of the sequence. Note, that if a sequence has external causes, like flickering, or changes in the ambient light, these effects will have therefore an impact in the temporal correlation histogram output.

As aforementioned, the correlation histogram of natural sequences is more frayed and presents continuous patches, instead of linear patterns and isolated (or less frayed) areas. It is imporant to highlight that symmetries regarding the diagonal are present again. This fact can be described locally by having objects appearing and disappearing. Ideally and refering to a local effect, an object (i.e. a first frame) that moves (second frame) and goes back to the initial position (third frame), causes an analysis such that references for first histogram are $I(u, v, t-2)$ and $I(u, v, t-1)$, and $I(u, v, t-1)$ and $I(u, v, t)$ for the second. Thus, bin values of both outputs are the same if one of them swaps its axis (because $I(u, v, t-2) = I(u, v, t)$). However, this effect cannot be interpreted in a global context, but rather gives an instinctive idea of the amount of symmetry expected. Therefore, exact symmetries cannot be expected.

Comparing the Synthetic output with the natural sequences exposes that temporal consistency is a problem with depth estimation algorithms. As previously commented, the first quadrant shows the amount of temporal variation of low values (background). Ideally, when having a static background, and as it happens to the Synthetic sequence, the desired output is a straigth line, meaning no temporal variation of those values that compound the background. Nevertheless, this first quadrant for natural sequences is unfortunately covered by a region and not a line. What is more, in the case of the Breakdancers sequence, this area is really big, revealing a high temporal inconsistency.
For the remaining sequences, it is worth mentioning that temporal consistency can be also seen, despite the non-use of available values.

# Chapter 4

# Improvement of MVD

It is well-known that image processing is aiming at two different, external and internal, targets: the first one is directly oriented to human perception by making up and improving pictorial information, and the second one focuses on data storage, coding and transmission [20]. In the field of 3D video with depth-enhanced representations, depth maps are used to represent the third spatial dimension. They typically utilize just 1 byte per pixel for the depth information, what is good for transmitting (less data on the channel than transmitting a stereo video pair). Although image processing contributed a lot to improve the visual perception, there is still a lack of extracting high quality depth information from a given data set (see Figure 4.1). Instead of improving algorithms that create depth maps, the main goal of the color video edge extraction method is to use the color information to, somehow, reconstruct the exact profile of silhouettes present in the scene, as these are the determining factor for the quality of rendered views. To carry out this idea, it is needed to have a tool to extract the edges from the video data, a processing tool to match this information with depth values and a reconstructor subsystem to reshape the depth map according to the new profiles.

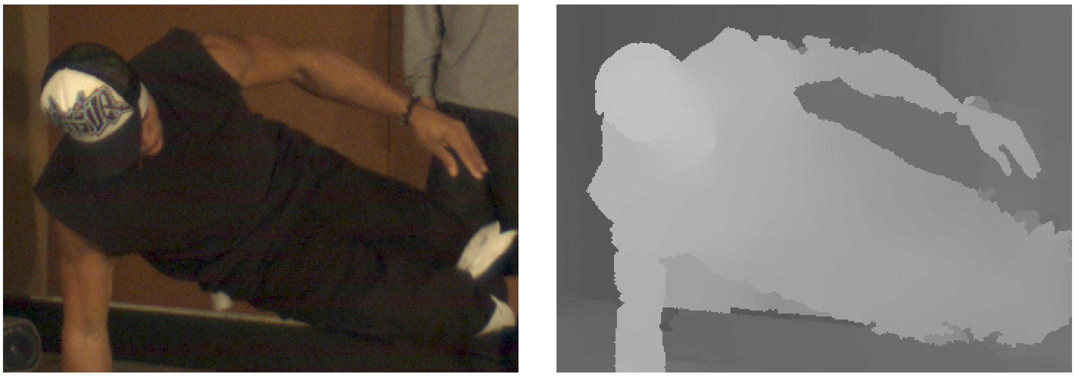


**Figure 4.1:** *Sample of Color plus Depth representation of Breakdancers sequence. Representative defective depth profile comparing to color edge.*

In accordance with the hierarchy of MPEG codecs, the implemented system works with 4 different layers. Figure 4.2 shows these layers from top to bottom: a ***sequence*** (layer 1) is composed by ***groups of images*** (layer 2). Each ***image*** (layer 3) is formmed by squared ***blocks*** (layer 4) which consists of a certain number of ***pixels*** (minimal image information). It has been designed in a way that a superior layer can use intrinsic operations of a lower layer, but not vice versa. Thus, the algorithm receives two input sequences (video and depth) and delivers one ouput (processed depth). The main processing pipeline takes the two components and reads in frame by frame. Matching this abstraction with the system hierarchy, it leads to 2 different subsets working in different layers: the Color Edge Detection subsystem working at the 3rd layer and the Depth Map Fitting subsystem operating at the 3rd and 4th layer.

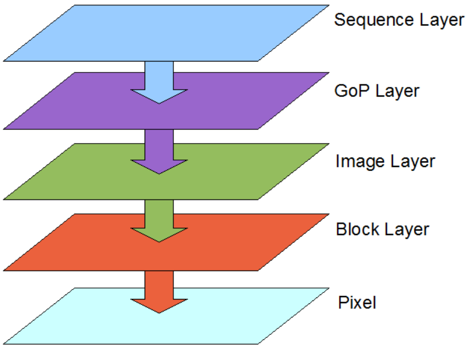


**Figure 4.2:** *Scheme of the hierarchical top to bottom layers. Each layer is composed by the layer below*

This chapter is organized as follows: section 4.1 gives an overview of the aforementioned subsets and a detailed explanation of their corresponding submodules. Section 4.2 presents a study of the correlation between extracted edges from color and depth. Results are presented and discussed in section 4.3 according to the system structure and its configuration options.

## 4.1 Extraction of color edges for depth map improvement

Having a look at the block diagram of the system presented in Figure 4.3, two independent main modules and their working layers can be distinguished. In general terms, the first subset is in charge of preparing the edge information ($E$) from the color component ($C$). The second subsystem, also in the 3rd layer, smooths the depth map ($D_b$). Furthermore, but now in the 4th layer, it elaborates a list of suitable blocks to be processed. Blocks that are not taken into account for processing are intended to be background or smoothly textured foreground. One must consider that the ground truth is the depth map itself, and consequently if a block of the depth map is not suitable to be improved it should be skipped instead of wasting time on computing something not improvable. Later, this criterion to establish the possibility of a block to be improved will be concretized. Then, this second subset finds blocks ($B_i$) of different sizes, which are divided by an edge and thus separating two regions. Depth values that are computed by a reconstructor module are set to each region. Finally, the last task of this subsystem is to compose a new depth map ($D_E$), getting on one hand the improved blocks ($B_i$) and on the other hand the blurred depth blocks (the corresponding blocks in $D_b$ according to the list of suitable blocks).

Due to multiple modules and submodules composing the system, several configuration options are available. In order to evaluate the impact due to input parameters and/or the chosen criterions of some submodules on the output, three working options have been defined and tested for the whole subsystem. These options have only influence on the second module. Next, the complete system and its modules are presented.

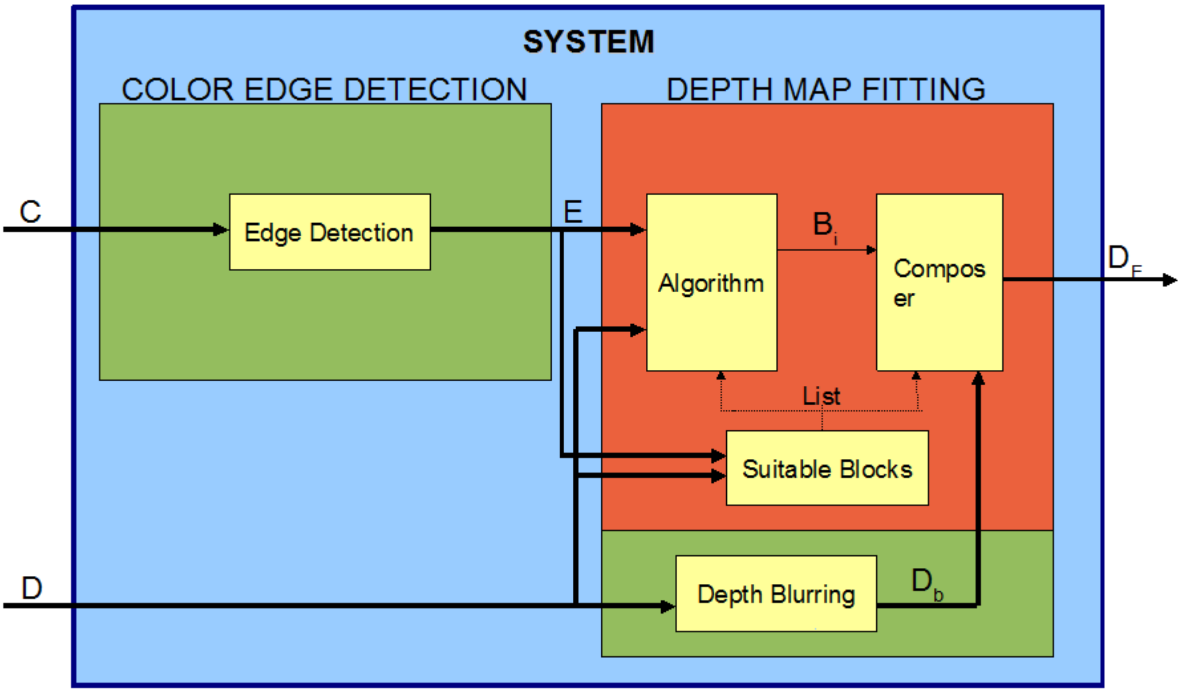


**Figure 4.3:** *Schematic diagram of the system working in the third layer (green) and in the fourth layer (red). Thick arrows represent intput/output images, the thin one input/output blocks and the discontinuity one metadata (the list).*

### 4.1.1 Edge Extraction

The main question regarding the extraction of the edge mask is: which edge detector among the various existing solutions is better adapted to this process requirements?. To answer this question, some history of edge detectors is presented.

Since the beginning of the 1970s, some specialists on image processing started to research focusing on edge detectors. In terms of image processing an edge corresponds to a discontinuity in the intensity surface. Therefore, a tool labeled as edge detector must serve to simplify the analysis of images by reducing the amount of data to be processed, while preserving useful structural information about object contours.

Localization and robustness with respect to noise are some of the objectives of edge detectors. Several methods have been developed with the aim of achieving these objectives. Some of these proposed methods are simple derivatives [21], [22], optimality criteria [23], curve fitting [24], junction restoring [25], [26] or linear operators [27]. In general terms, one could divide the different edge detectors in three groups: the first uses an intensity gradient of the image, the second performs a post-processing algorithm based on recovering scene topology and the third compares different fittings. They all have in common a certain pre-processing step; an image, from the signal processing point of view, is composed by the sum of information plus additive Gaussian noise. Thus, a filtering step to reduce the noise, which normally would be a Gaussian operator, is needed. The main reason to use a Gaussian filter is because it is computationally inexpensive. However, the size of the Gaussian kernel must be set as an external parameter.

Over the years, algorithms became more complex than a simple derivative. The fact of having a decision criteria in a part of the algorithm requires the systems to introduce more external parameters, in some of them called thresholds or scales, depending on the functionality of each one. This matter can be easily solved by just setting a constant value to each external parameter, but this would lead to sub-optimal results, because sequences differ in characteristics (amount of motion, texture, etc.).

Arriving at this point, the question of which edge detector should be used is posed again. The most obvious answer would be "the best one", but some other requirements must be considered. One of this requirements is the ease to find public source code. While surfing the net, the most popular and casted detector is the Canny. It is also implemented in the OpenCV project, which library has been used to develop this project.
Furthermore, a study from 1996 reveal that Canny and Nalwa-Binford edge detectors are the best ones [28]. Nalwa-Binford has the advantage that setting external parameters to a constant value does not affect to results quality as it happens for other edge detectors. However, Canny presents better results when parameters are chosen adaptively. In addition, it is found that Canny's thresholds could be set dynamically [18]. Considering all these points, the Canny edge detector is the elected one to be used in the presented system.

Now that the core component of this subsystem is clarified, the other submodules will be introduced. Figure 4.1.1 shows the scheme of the edge detection subset:

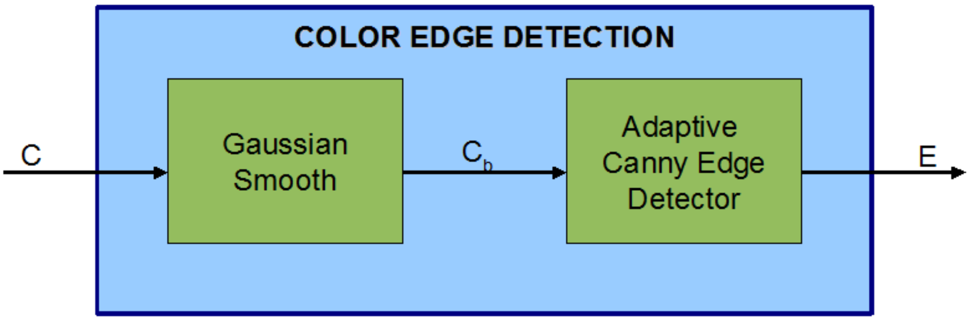


**Figure 4.4:** *Scheme of the edge detection subsystem. The Color image (C) is blurred ($C_b$) for a better output (E) of the Adaptive Canny Edge Detector submodule.*

As it can be seen, these two aligned processes are necessary for extracting the edge mask. Next, it is explained how these submodules work:

- **Pre-processing**
  As aforementioned, the main motivation is to reduce rough texture while preserving contours. It means that it is interesting to smooth spatial variation in pixel intensities (texture), when these intensities are lower than a high spatial variation in pixel intensities (edge). Although reducing texture is important for the next submodule (it will prevent false edge-detections), most important issue here is to keep up the edges. This trade-off could have been achieved by using a bilateral filter [29]. However, it needs more external parameters that depend on the texture of the image. Consequently, a texture pre-analysis would be needed in order to set the values properly. However, texture analysis is not the

target of this work. Moreover, bilateral filter works better with CIE Lab color representation.
From a signal standpoint, an image is compound by useful data and random noise. Thus, the best way to reduce the noise is with a low-pass filter, tpically with a Gaussian shape. Moreover, the computational cost of implementation of this filter is not expensive. Consequently, it saves a lot of computational cost for further steps. A Gaussian smoothing from the OpenCV library is used and the kernel size required for the Gaussian filter is set to a constant value. The values accepted by the function are odd numbers and the value is set individually for each sequence.

- **Canny Algorithm**
  Thanks to the OpenCV project it was not necessary to implement the Canny algorithm. Even so, its functionality must be explained. According to Canny's publication [23], the edge detector is composed of 4 steps. First of all, the input image is smoothed with a Gaussian filter to reduce noise. This task is already done by the previous submodule. Then it finds the image gradient to detect regions with high spatial derivates (here an aperture size for the Sobel derivative operator is required). The third step is the so-called *nonmaximum suppression.* It tracks along the regions gathered in the previous step and suppresses all those pixels that are not at the local maximum of the gradient direction. Finally, it again tracks along those remaining pixels and applies a *Hystheresis Thresholding.* If a value is below the lower threshold ($T_L$) then it is rejected. The value will be accepted as an edge point if it is above the high threshold ($T_H$) and if the value is between the $T_L$ and $T_H$, it is labeled as edge iff it is connected (with a path) to a value over $T_H$.
  After testing several inputs for the Canny edge detector it has been found that the best input is the luminance channel (Y). The theoretical justification is very simple. Using just one color channel would exclude some relevant color information for edge detection. It is also sub-optimal to use the 3 RGB channels by combining them with boolean operations (e.g. if $R == G == B \Rightarrow$ output $= 1$, or e.g. if $(R + G + B)/3 > 50\% \Rightarrow$ output $= 1$). Another possibility, also sub-optimal, could be to average the 3 channels, resulting in a non-standard luminance. The reason why all these possibilities are sub-optimal is that they do not consider the nature of the light and how the human eye absorves it [30]. Therefore, the input for the Canny algorithm module will be the standard luminance defined as:

  $$Y = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B \tag{4.1}$$

  with R, G and B in the range [0,255]. It would also be interesting to cite some discarded options for the input of the Canny algorithm module. It has been seen that, in general, edge detectors do not work good on dark image areas. Having in mind the nonmaximum suppression of the algorithm, one evaluated option was to use subdivisions of the images as input for the edge detector, making the edge detector to work better in darker parts of the image. Unfortunately this idea did not improve the Canny output.
  Nevertheless, the main inconvenience of the Canny algorithm is the need of three external parameters. Two external parameters are the low and high thresholds $T_L$ and $T_H$, respectively. In order to have a dynamic behavior and an independent treatment for each image, thresholds for the Canny algorithm are defined addaptative for each image. The main reason is that for the same scene there is not always the same ambient light in each frame and consequently the sharpness of some edges could be distorted. Basically, a histogram of the spatial derivatives is included between the second and third steps explained above (as it is similarly presented in [18]). As explained in section 3.1, histograms show, how the information is distributed. Hence, it is a good tool, as reveals the amount

of "strong" spatial derivatives. Hence, these spatial derivatives will always be succesful detected. Vice versa, using a histogram for spatial derivatives is good because reveals the "weak" ones. Therefore, the histogram will help to erase a lot of noise. Specifically, the histogram is computed with 64 bins. Then, thresholds are chosen according to a rule: $T_L$ is the bin (later scaled from [0,63] to [0,255]), which is equal or higher than the 80% of the image resolution. $T_H$ is set to 2.5 times the first threshold. The values for the first threshold and the ratio are arbitrary, but after several tests it was proven that they were most suitables for the given data set. There are also some recommendations, like in [31], that suggest a ratio high:low between 2:1 and 3:1. By setting the proposed values, this rule is accomplished and it is ensured that $T_L$ and $T_H$ are in the range [0,255]. Further, the aperture size parameter is directly set to 3, as higher values do not give satisfactory outputs.

The output of this module is a binary image with the same resolution as the input. This binary image is one of the two inputs for the next processing step. However, the edge mask presents some imperfections. As aforesaid, edge detections are preferred over missing structural edge information. Consequently, there can be texture areas, not totally smoothed, that are represented as edges (see Figure 4.5). Then, the next module will have to detect these false cases and try to remove as many as possible. Further investigations could focus on preserving the correct location of the edges while smoothing high textured areas. The use of a bilateral filter is a good option. In order to set parmeters dynamically, a texture analysis must be done. Nevertheless, there is also some other inevitable cases where dark areas present false edge detections.

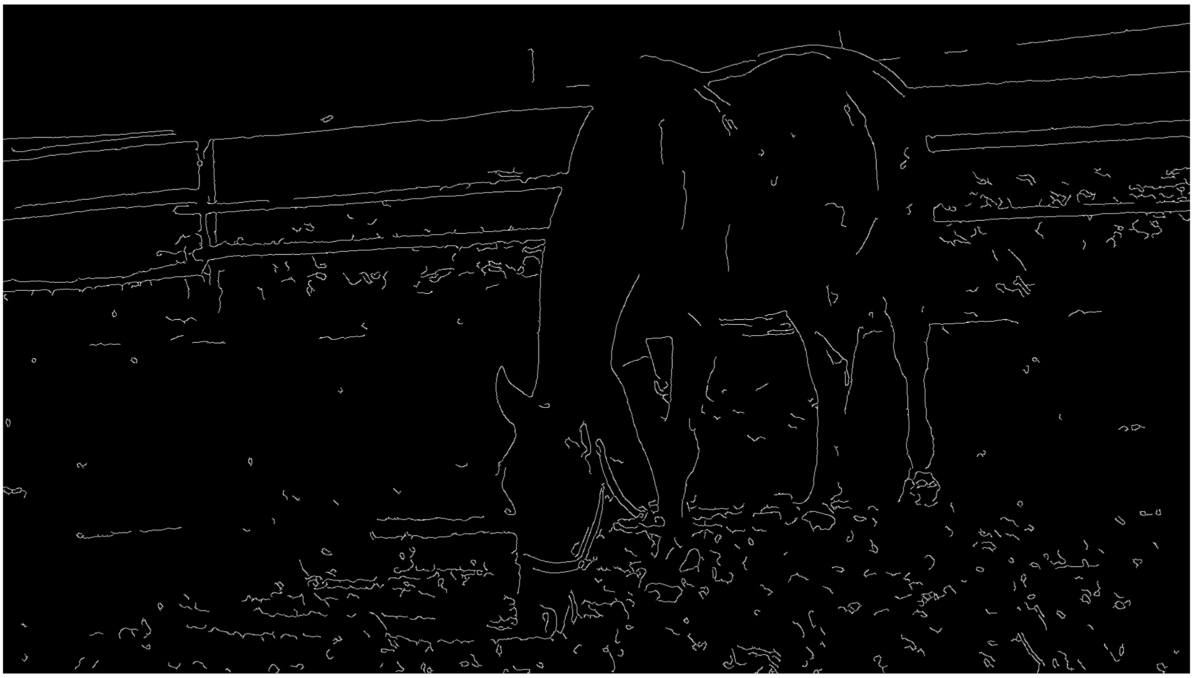


**Figure 4.5:** *Canny output for Horse sequence. Grass (rough texture) is detected as edge information.*

### 4.1.2 Depth enhancement due to edge information

This subsystem is in charge of extracting a list of blocks from the depth maps, which are capable to be processed and to use the edge mask obtained from the color information to improve the silhouettes of the scene. To achieve this improvement, the depth map fitting subsystem treats each suitable block independently. For this purpose, each block is analyzed in order to detect if it contains two (and only two) regions of pixels divided by one (and only one) edge line. Here, a region must contain at least one pixel and the edge line does not belong to any region. If this is not the case, the block is splitted in a quadtree structure and the subblocks are consequently analyzed again, as if they were a normal block, but the number of pixels being a quarter part of the original one. If the block presents two regions, then the subsystem re-ffits the depth map according to the region information obtained from the color. Once those parts of the images, that are suitable to be improved, are processed, the new depth map is built. Figure 4.6 illustrates the 4 submodules and their working layers can easily be differentiated:

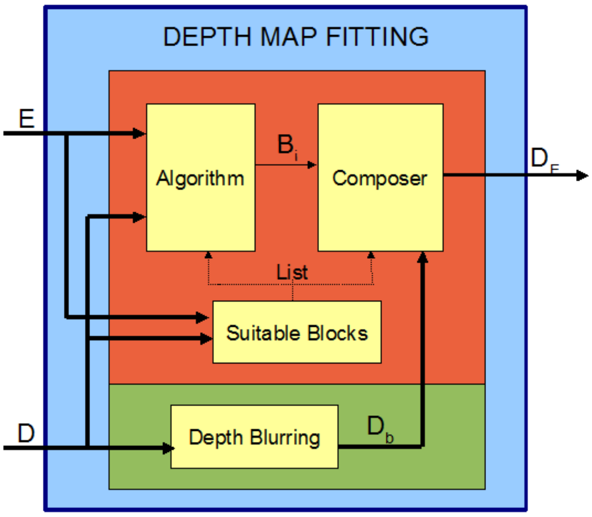


**Figure 4.6:** *Block diagram of the depth map fitting subset and according working layers: the green area corresponds to the image layer and the red one to the block layer.*

The reason for creating such a list in this subsystem is to improve all blocks according to their characteristics. Hence, for effectively improving blocks, an order of different actions will be set. Here, the first open question appears: how to distinguish between types of blocks. The improvement step and the rebuild process are the core of this subset. The functionality of each module is detailed below together with their working options, if available:

- **List of suitable blocks**
  First of all, it is important to define what should be understood as a suitable block to be processed. In this context, a depth block can either contain a homogeneous part of the image, like the foreground or the background, or contain a non-homogeneous region, like an interaction between the foreground and the background. Hence, there is a distinction between homogeneous and non-homogeneous blocks, where the first group contains those not suitable to be processed by the algorithm module. The other group, where an edge (or more than one) is present, contains those qualified to be processed. In terms of processing, the homogeneous blocks will present a sharp histogram, with the main characteristic of

a low variance. Contrary, the other set might yield a less uniform histogram, resulting in a higher variance. Mathematically and regarding to computation time, the calculus of the variance is a good method to have a threshold for differentiating types of blocks. For this, the common formulae of the mean value (4.2) and the variance (4.3) are:

$$\bar{x} = \frac{1}{N} \cdot \sum_{i=0}^{N-1} x_i \qquad x_i \in B_j \tag{4.2}$$

$$\sigma^2 = \frac{1}{N} \cdot \sum_{i=0}^{N-1} (x_i - \bar{x})^2 \qquad x_i \in B_j \tag{4.3}$$

with $N$ being the number of pixels of the current block and $x_i$ the grey value of a pixel that belongs to block $B_j$.
In the processing chain, the variance is first computed and then compared to a constant threshold. If the variance is higher than the threshold (dispersion of values), this block will be considered as suitable to be processed. Otherwise it will be skipped for the following submodules.

Three different options are used for setting up the parameters in this module. In Option1 and Option2 the threshold is set to a constant value. For the first option, the algorithm is launched for eleven values (from 20 to 30 in steps of 1), in order to evaluate the differences in terms of statistics (number of processed blocks, discarded, etc.) when chosing a constant threshold. For the second, the algorithm is launched just once but according to the optimal threshold of the given sequence. This optimal threshold is obtained from the correlation between color edges and depth maps presented in section 4.2.

However, another option to distinguish between suitable and non-suitable blocks for processing is simply the edge detection itself. In this case, the input of the system is the edge mask. Here, blocks with no edge are labeled as non-suitable. Otherwise, if just a single pixel of a block has an edge value, the block is labeled as suitable. This is the criterion used in Option3.

- **Algorithm**
  This part of the process is the core of the complete system. The main motivation of this work (improving the silhouettes of the depth maps) is basically implemented in this block. Thus, the goal of this submodule is to deliver an improved depth block to the next. As one can imagine, this is extremely optimistic and hence not always this improvement will be achieved, in which case the block will not suffer any alteration.
  This module is a recursive algorithm based on a quadtree structure. It means that, the algorithm generates four calls to itself by changing some input parameters. This operation starts with the "maximum block layer size", a constant parameter that depends on the resolution of the frame. According to another constant parameter ("number of allowed recalls"), this process of re-calling itself by dividing the block in quarters can be done three times. A block can be subdivided and a recursive function can be executed for each of these subblocks. As an example, if a block (which size is 64) is subdivided, the algorithm recalls itself four times (for blocks which size is 32). For each subblock, this operation is repeated if is needed. This operation can be done up to three times, meaning that the smallest block size in this example is 8.

In order to provide a deeper insight to the procedure of this module, Figure 4.7 illustrates the processing structure:

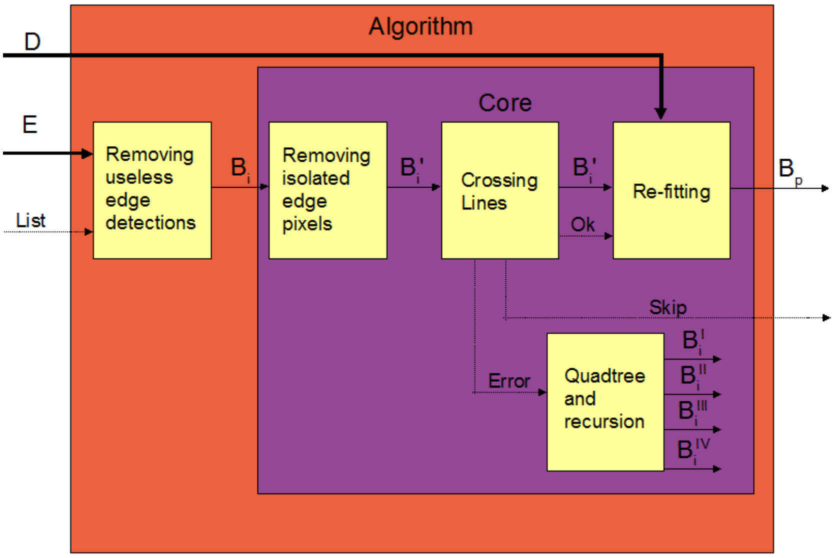


**Figure 4.7:** *Scheme of Algorithm module. It needs 3 inputs, 2 are in the 3rd Layer (E and D) plus metadata (List). It delivers blocks (4th Layer) plus labels (Metadata).*

Before explaining all submodules in detail, some definitions are previously needed.

1. **Neighborhood**
   In the case of pixel domain, a neighborhood of a pixel is the number of pixels immediately close to it. I.e., a 4-pixel neighborhood are the upper, lower, left and right pixels of the current one.
2. **Maximum distance allowed**
   Positions are indexed with natural numbers in a discreet domain. In the case of image processing, pixels of an image are usually indexed with two coordinates, namely $u$ and $v$. Thus, the maximum distance allowed is defined as the euclidean distance between a pair of pixels. I.e., for a 8-pixel neighborhood the maximum distance allowed is $\sqrt{1+1} = \sqrt{2}$.
3. **Connectivity**
   Condition of a pair of pixels such that their distance is below or equal to the maximum distance allowed.
4. **Initial and end sets of points**
   Set of points which are connected. All points must belong to one of the 4 borders of the block. Each point must be connected, at least, with another point of the set. Initial ($Ini$) and end ($End$) sets of points are a part of an edge in a block. They are unconnected, meaning $Ini \bigcap End = \emptyset$ (see green pixels in figure 4.8).
5. **Path**
   Set of points which are connected. The $Path$ must connect the $Ini$ and the $End$ set. Because of that, any point of the set can not belong to a border, meaning $Ini \bigcap Path = \emptyset$ and $End \bigcap Path = \emptyset$ (see white pixels in figure 4.8).
6. **Isolated line**
   Set of connected points such that they have no connection to a $Ini$, $End$ or $Path$ set.

This algorithm works with an 8-pixel neighborhood, resulting in a maximum distance allowed of $\sqrt{2}$. Next, submodules are explained according to the definitions.

- Removing useless edge detections
  At the end of section 4.1.1 the imperfection of detecting the edges is mentioned. However, those regions with high texture in color component are not considered, if they represent foreground or background (low variance) in depth maps. Taking into account the information from the list of suitable blocks, it is useless to process blocks that are not going to be improved and therefore some computation time will be saved. Thus, this submodule, that has the edge mask as input, deletes small edges in blocks iff these blocks are labeled as suitable blocks[1]. In terms of signal processing the noise is being reduced while information is being preserved in this submodule.

- Removing isolated edge pixels
  This submodule erases edge pixels that are located on block borders ($x = 0$, $y = 0$, $x = bs - 1$ and $y = bs - 1$) and do not have another edge pixel within maximum distance allowed. Note, that on block borders, the neighborhood has three pixels less than a normal 8-pixel neighborhood for regular positions, except corners that have 5 pixels less (although the maximum distance allowed in both cases is still $\sqrt{2}$). This process avoids having $Ini$ or $End$ sets of points with just one element and no connection to a path.

- Crossing lines
  Once having the block with less possible noise, the crossing lines submodule is in charge of finding a unique line that crosses the block. It means, that there is one and only one $Ini$ and $End$ sets, and that they are connected by a unique $Path$. This $Path$ must contain the total number of pixels. Therefore, blocks that presents lines with branches (see figure 4.8) are discarded. This process follows three elementary steps: the first is to check if there are a valid $Ini$ and $End$ set; the second is to verify if a path that connects both sets exists; the third one checks the absence of an isolated line. This submodule produces three metadata outputs, what are labels for the current block. The "Ok" label is set to a block that passes these three checks. The "Skip" label is for those blocks that do not contain any edge pixel. As it can be foreseen, just the first check is the one able to set "Skip" for a block (if $Ini = \emptyset$ and $End = \emptyset$). The last label, the "Error", is for those blocks that do not pass all three checks.

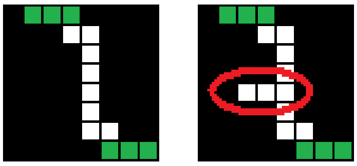


**Figure 4.8:** *Example of a correct path (left) and an incorrect path with a branch (right). In both cases $Ini$ and $End$ sets are marked with green*

[1] Note, that this submodule is not part of the recursion and therefore is applied once to each block of the suitable blocks list.

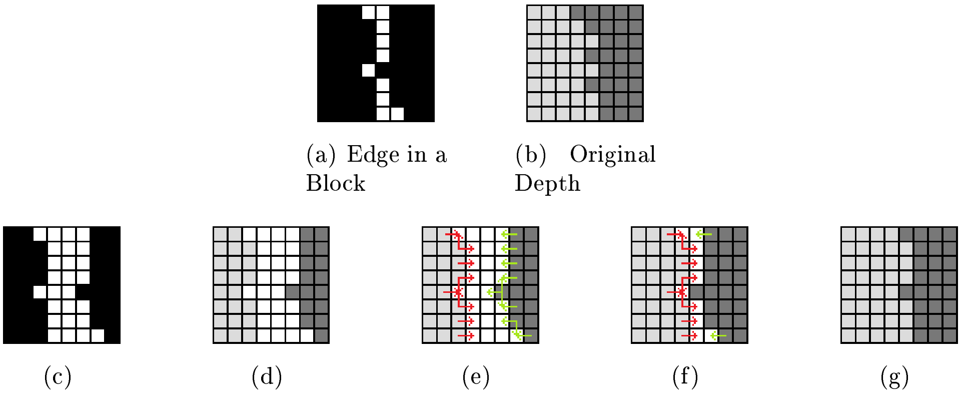
- Quadtree and recursion

  If a block is marked with the "Error" label, this submodule splits the current block into four new blocks, whith half the side length of the parent block. However, coordinates must be recomputed according to each child block. Then the recursive function is re-called for each child, and the core algorithm (except the Removing useless edge detections submodule) is applied again. According to the constant parameter "number of allowed recalls", a suitable block can be splitted up to 3 times. I.e. if a suitable block has 40x40 pixel, the algortihm can re-call up to blocks with 5x5 pixel.

- Re-fitting

  The input of this submodule is a block composed by two regions, which are separated by a line. The readjustment process must be done over depth transitions, what in color edge detection means over the edge line detected. The typical problem that depth maps present in contour areas is some jittering over the desired profile (see figures 4.9(a) and 4.9(b)). Normally the width of this jittering is one pixel, but in some cases it is more than one (and therefore a noisy contour). It is reasonable, to create an "unknown" area around the edge (from the edge mask) to, afterwards, reconstruct the correct profile according to color information. The easiest way to implement this idea is by doing a dilation (see figure 4.9(c)) of the line with a cross structuring element (one pixel above, one below, one right, one left). Thus, the unknown area is a 3-pixel wide line. Then, the block gets divided in three different regions, namely A, B and Line. For regions A and B, values from the depth map are directly copied to the reconstructed depth block (figure 4.9(d)). The reason to copy values directly is because the output should be as similar to the input as possible, but with reshaped contours and with smoothed flat areas where no edge is present. Hence, the depth information is preserved for regions A and B. In order to avoid miss-reconstructions, the variance of region A and region B are computed to ensure flat regions. If region A and/or B are not flat (high depth transitions), then the block is discarded and consequently not improved. Here, the threshold used to discriminate is the same as the one used in the List of suitable blocks module for Option1 or Option2. Mathematically, if $\sigma_A^2 < Th$ and $\sigma_B^2 < Th$, then the block is accepted. Finally, the Line region is reconstruced as it is illustrated in figure 4.9(e). To achieve this, pixels adjacent to region A that belong to the Line region are computed by averaging pixels from A considering a 4-pixel neighborhood (note, that the maximum number of pixels to average is 3). I.e., in a vertical line, pixels of the left side of the Line region will take the left adjacent pixel (because this is the only pixel that not belongs to the Line region in a 4-pixel neighborhood). The same is done for pixels adjacent to B. Once adjacent pixels to A and B are computed, they are labeled as part of A or B region, according to their adjacency. The last step is to set correct values for the pixels of the real edge line. Similarly done with a 4-pixel neighborhood, the maximum value of the 4 neighbors (regarding only to those that belong to region A or B, as line region are unknown values) is the selected value to be set (see figure 4.9(f)). Once it is done, the block can be regarded as improved (figure 4.9(g)).

Despite the fact that this submodule has several subsets, it is totally independent of the working option. It works purely as a set of actions/operations independent of the criteria to decide if a block is suitable or not, or the way a block is smoothed or not. Therefore, it makes the structure robust, as it does not change its procedure.


(a) Edge in a Block (b) Original Depth

(c) (d) (e) (f) (g)

**Figure 4.9:** *Scheme of Re-fitting submodule.*

- **Depth blurring**
  In order to solve the background problem presented in section 2.1, it is thought to use a filter to blur the depth image. Thus, all those blocks labeled as background or foreground could feature the mentioned problem and can be improved with a rough smoothing. This smoothing is implemented, with a Gaussian filter again, which kernel size is 9. Another option, that was discarded, was to use a Median smoothing filter. The main problem of using this filter is that the output presents a block-effect, which is a problem for the rendering process (it may cause effects like having boxes standing out of the background while it should be flat). However, the Median filter showed good local results, but for the global goal of this process a continuous blur is preferred. Note that this process uses a Gaussian filter and, as previously explained, it is fast and easy to implement.
  In Option1 a smoothed depth map is created in order to provide improved depth inforamtion for the depth composer (for those blocks labeled as not suitable to be processed). A hierarchycal smooth is thought for the other working options in order to provide improved depth information for those blocks that have been labeled as "Skip" in the previous submodule. Thus, a set of four smoothed depth maps with a Gaussian filter is prepared for the depth composer submodule, whose kernel sizes are 9, 7, 5 and 3 (three more smoothed depth maps according to the constant parameter that allow the reinvokation of the Algorithm submodule).

- **Depth Composer**
  This submodule is intended to embed the solutions provided by the Algorithm and the Depth blurring submodules. Working in the Block Layer, the Depth Composer assembles the output with all the output blocks of the Algorithm submodule. Then, all those blocks that might correspond to the background or the foreground, are taken from the Depth blurring submodule's output. Thanks to the List of suitable Blocks, these remaining blocks which not belong to this list are labeled as background or foreground and therefore they complete the improved depth map. Hence, the output of the whole system has an improved background and foreground, and silhouettes which correspond better to the real scene.
  In Option1, when composing the new depth map, all those blocks that have been labeled as not suitable are taken from the smoothed depth map (with a Gaussian kernel of 9). However, the "Skip" blocks from the Algorithm submodule are copied directly from the original depth map, without any change. As it can be seen, this is a less forcing working option. On the other hand, Option2 and Option3 have the same behavior as Option1,

but the "Skip" blocks are copied accordingly to their block size and their corresponding smoothed depth map. Table 4.1 summarizes the relations:

| Number of pixels per block | Kernel size |
|---|---|
| $Max\_block\_size^2$ | 9 |
| $Max\_block\_size^2/4$ | 7 |
| $Max\_block\_size^2/16$ | 5 |
| $Max\_block\_size^2/64$ | 3 |

**Table 4.1:** *Relation of kernel size and block size*

As it has been mentioned in the description of the modules, three different working options have been chosen. The modules that depend on these options are the Suitable Blocks, the Depth Blurring and the Composer. In summary, the first one elaborates a list of suitable blocks according to variance criteria (Option1 and Option2) or according to edge presence (Option3). The second module uses a single smoothed regular background and foreground for non-suitable blocks in Option1 or a hierarchycal smoothed regular background and foreground for "Skipped" blocks in Option2 and Option3. Finally, the Composer copies original depth values for "Skipped" blocks in Option1 or takes the corresponding smoothed block by the Depth Blurring module accordingly to its hierarchy in Option2 and Option3. Table 4.2 organizes those set-ups by working option:

| | **List of suitable blocks** | **Depth blurring** | **Depth composer** |
|---|---|---|---|
| **Option1** | Calculus of variance | Single depth blur | No change if "Skip" label |
| **Option2** | Calculus of variance | Hierarchical depth blurring | Depth values from hierarchical smoothing |
| **Option3** | Edge criteria | Hierarchical depth blurring | Depth values from hierarchical smoothing |

**Table 4.2:** *Working options and the corresponding block's set-up.*

Similarly as the Color Edge Detection subset, this subsystem has inherent problems. Analyzing each module, the List of suitable blocks and the Composer modules do not introduce any problem. However, the Algorithm is based on some criterias that can be discussed. On the one hand, the Removing isolated edge pixels submodule acts without considering the possibility of deleting a real edge. One reason for this is the use of a fixed grid, meaning non-shiftable. Then, a pixel belonging to a border of a block could be, i.e., the peak or the end of a curve of the adjacent block. Therefore, the edge information is being slightly distorted. However, there is a trade-off between preserving the edge information entirely or, in contrast, deform it a bit and benefit of the behavior of the algorithm. Conclusions have led to the second option. On the other hand, the Re-fitting submodule is based in a blind faith in one pixel-width jittering of the depth maps. This jittering is measured from the edge that is in the edge mask to the farther depth value of the corresponding depth transition (it is equal to 2 in the example presented in 4.9). In that case, the dilation takes no effect for providing the "unknown" area (look at top and bottom pixels that keep invariant in figure 4.9), because the high jittering remains invariant (the one pixel-width jittering is corrected). In order to set a correct "unknown" area, it would be possible to compute the area between the real color edge and the depth transition. In this case, larger scale jittering would also be covered. Finally, the Depth blurring submodule might be too aggressive with a big kernel size. This becomes a problem afterwards, when composing final depth maps. In such a case, background would be distorted by blurred values comming from an edge transition. Further investigations could focus on finding a filter that smoothes the blocks roughly and that is aware of high depth transitions in order to avoid mixing background and foreground values.

## 4.2 Correlation between edges in color video and depth maps

All this work is based on the correlation between depth transitions and silhouettes extracted from the color component. It has been explained previously that a block or region with depth transitions means high variance. If this block is analyzed with a histogram, it leads to isolated regions of values, such as figure 4.10 illustrates.

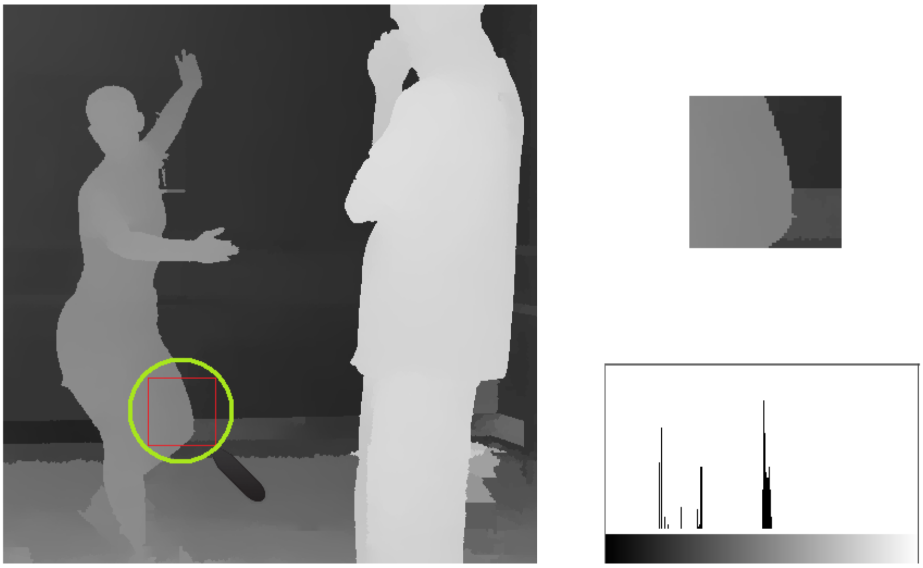


**Figure 4.10:** *Example of a block with high variance. On top right a zoom of the marked block and its corresponding histogram, with grey values accumulated around 4 different mean values.*

A histogram with this kind of distribution leads to a high variance, revealing a correlation between depth transitions and a statistical parameter, namely the variance. Taking into account that depth maps are the signal for describing the 3D geometry of the scene, it is reasonable to match the following conditions: if a block or region has a high variance then the edge mask should also contain an edge (because it represents structural information of the scene); if a block or region of a depth map has a low variance (flat regions), then the edge mask might contain an edge or not (because of rough texture detected as an edge, as it happens with the Horse sequence). These two conditions lead to a third one that should always be satisfied: it is not possible to have a depth block or region with high variance and no edge in the corresponding edge mask. Hence, two conclusions can be extracted. The first is that depth maps might be inexact. The second one is the weakness of the edge detection at yielding the edge mask. However, after several tests, it has been verified that edge mask contain the whole structural information plus noise caused by rough textures.

In that sense, this study of correlation between edges and variance of depth maps reveals non-homogeneous regions when no edge is detected. Loosely speaking, it highlights irregular background regions of depth maps. Specially Ballet or Breakdancers sequences contain such regions, where instead of having a gradient of grey values in background areas (because they are flat and not irregular), depth maps show a kind of wavy and step line regions (see A.1(b) and A.1(f)). This is one of the inherent problems of depth maps estimated from natural sequences.

This correlation study intends to justify the difference between the working options. As aforementioned, they are organized from the passive to the active sets of actions. The first and the second option have the same number of suitable blocks, because their criteria is the same. Nevertheless, the third option has more suitable blocks due to more blocks with edges than blocks with a variance higher than the selected threshold. This correlation is also useful to select the threshold for the List of suitable blocks and the Algorithm modules.

In order to verify that the aforesaid third condition is accomplished, the correlation between high variance and extracted edges is analyzed. To achieve this, two one-dimensional arrays are created, namely $V_d$ and $E_m$. $V_d$ contains the variance (floating point values) of blocks that corresponds to their 3rd subblock layer (16x16 or 10x10 pixels depending on sequence's resolution). I.e., $V_d[0] = 37$ means that the top-left block of the depth frame has a variance of 37. The other array, $E_d$, contains boolean values, being true if an edge pixel is found in the corresponding block or false when there is no edge pixel. Then, a third one-dimensional array is generated ($R_{V,E}$), which elements are:

$$R_{V,E}[k] = V_d[k] \cdot \overline{E}_m[k] \tag{4.4}$$

where $\overline{E}_m$ is the logical negation of vector $E_m$, meaning true (equal to 1 in terms of a value) for no edge found and false (equal to 0 in terms of a value) for an edge found. Consequently, the resulting array contains two different types of elements. One set of elements is the variance of those blocks that do not contain an edge. The other set is equal to zero. Ideally, all those elements different to zero should be low values (meaning depth flat regions).

Hence, all sequences have been analyzed in order to derive an appropiate threshold for each sequence. First, $V_d$ and $E_m$ have been computed for each frame. Then, they have been merged into a single $V_d$ and $E_m$ one-dimensional arrays regarding to number of frames of each sequence. For the case of $E_m$, if an element is higher than the half of the number of frames, is set to 1. It means that there is an edge in a certain block for more than the 50% of the frames. Oppositely, if an element of the array is lower than the half of the number of frames it is set to 0. Consequently, it leads to a boolean vector and equation 4.4 can be applied. Once $R_{V,E}$ is obtained, another one-dimensional array is created, namely $Acc_R$. This array contains the number of elements of $R_{V,E}$ which are higher than a certain value (remember that these values refer to variances). I.e. $Acc_R[30] = 18$ means that there are 18 elements of $R_{V,E}$ with a value (variance) higher than 30. Consequently, the plot of this new array is monotonically decreasing and the first element contains the total number of blocks that have been analyzed for each sequence. Figure 4.11 shows the plot of $Acc_R$ for all sequences. The plots have been normalized with respect to the to number of analyzed blocks. For the Horse sequence, the number of blocks that have been analyzed is 20736 and 3072 for the remaining five sequences.

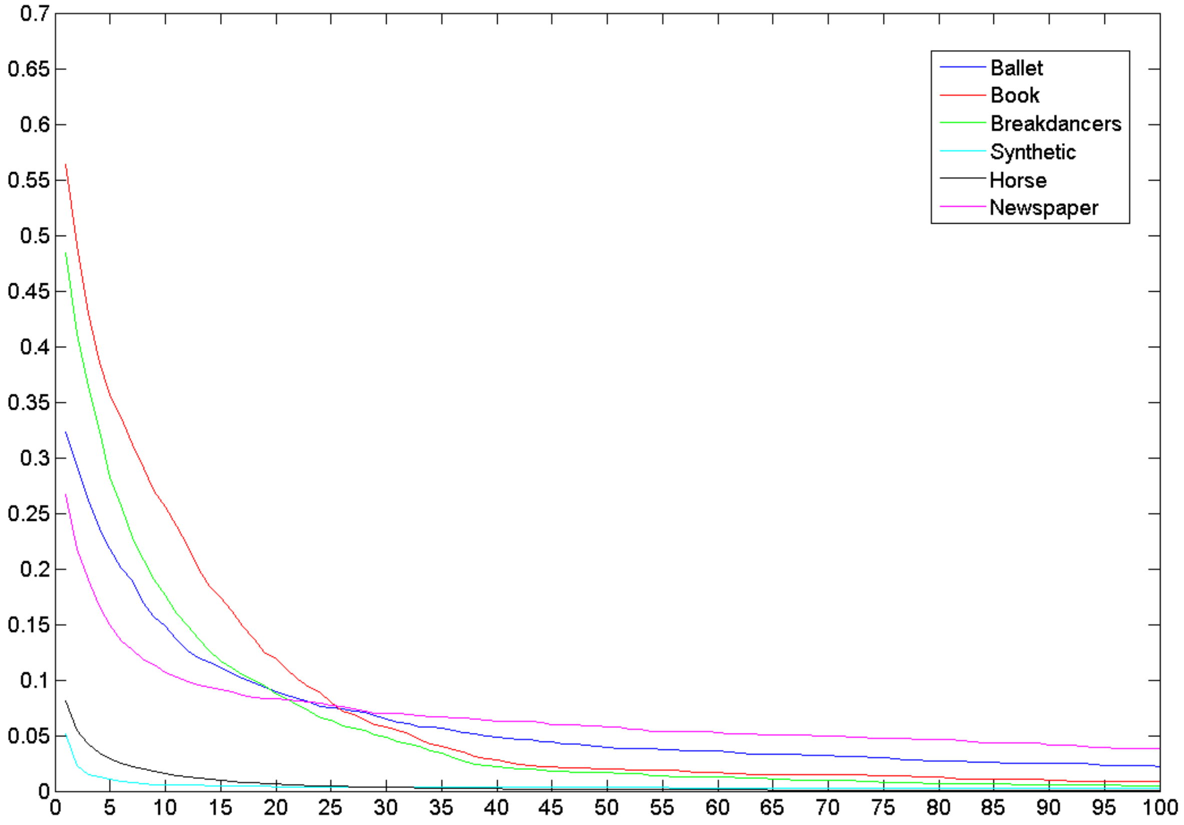


**Figure 4.11:** *Plot of $Acc_R$ array for each sequence. X axis is the absolute variance $k$ used to compare and Y axis is the relative $Acc_R[k]$ (accumulated values of $R_{V,E}$).*

The results presented in figure 4.11 show the predicted behavior with $Acc_R[0] = 1$ and $\lim_{n \to \infty} Acc_R[n] = 0$. $Acc_R$ contains 100 elements for all sequences. As an example, for Breakdancers $Acc_R[30] = 0.05$ means that there are 5% of blocks analyzed that have a variance higher than 30 and no edge pixel. Values in between decrease monotically. For low thresholds, sequences present very disparate values. The Synthetic sequence exemplifies that with $Acc_R[1]$ that has a bit more than a 5% of 3072 blocks with a variance higher than 1 for blocks with no edge pixel in the edge mask. Oppositely, Book has almost a 57% of blocks with a variance higher than 1 and no edge in the edge mask. As justified in Chapter 3, the synthetic sequence has ground truth depth information and, therefore, might present better results that the other ones.

Having a look at the graphic, the Synthetic and the Horse sequences have very low accumulated values for variances higher than 20. The Horse sequence shows this behavior due to the small range of used values (referring to section 3.4.1, the Horse sequence uses only 67 grey values for the depth map). Hence, the smaller the range, the lower the probabilities to have high variance. The other sequences present different characteristics. Analyzing the variance interval $[20, 30]$, Newspaper keeps practically constant at about 10% of blocks, while Book decreases by almost 5% (from 12% to 7% of blocks approximately). The Breakdancers sequence has a similar decreasing profile as Book, while Ballet is more or less in between the lower (Newspaper) and the higher variation (Book). Due to different behavior of the data set in this interval, the first working option is launched for these eleven thresholds, from 20 to 30.

Unlike Option1, where all sequences are launched with the same thresholds, Option2 and Option3 use different thresholds depending on each sequence. In this case, the criterion is regarding the Y axis. The idea behind is to accept a certain number of blocks. Hence, $Acc_R$ is this certain number of blocks, namely tolerance. To achieve that, two different tolerances (degree of acceptance) are chosen in order to evaluate different behavior. Those values are set to 10% and 5% of the blocks. Table 4.3 summarizes the correspondence between tolerance and variance thresholds for each sequence.

| | **Tolerance** | |
|---|---|---|
| **Sequence** | **10 %** | **5 %** |
| Ballet | 18 | 39 |
| Book | 23 | 33 |
| Breakdancers | 18 | 30 |
| Horse | 1 | 3 |
| Newspaper | 12 | 69 |
| Synthetic | 1 | 2 |

**Table 4.3:** *Summary of correspondences between variance and tolerances (number of blocks with variance and no edge).*

As it can be seen in this table, values really differ for each tolerance. In this case, it makes sense to set a fixed tolerance, what leads to a particular variance threshold for each sequence. Hence, Option2 and Option3 are launched twice per sequence, one for a tolerance of a 10% and another one with a tolerance of 5%.

## 4.3 Improvement results

The same MVD data sets as used in Chapter 3 have been tested with this system. Again, the Synthetic sequence is the ground truth depth information, as edges in the scene match perfectly with 3D scene's geometry. Thus, the output when applying the system should be equal to the input. In order to have equalized results, first 25 frames of each sequence have been tested. As sequences present different features (see Appendix for a description), it is ensured that the first 25 frames contain the described motion. However, due to format constraints (the data analyzed is video and the presented work is printed format), in this section the first frame of first camera of each sequence is selected for comparing with original sets (they are in the Appendix).

The system has some fixed parameters, independent of which working option is chosen. Following the module's description presented in section 4.1, the first external parameter is the aperture size of the Gaussian smoothing applied to the color component. As aforesaid, it varies depending on each sequence. For Ballet, Book and Newspaper sequences this value is 3. The aperture size is 5 for the Synthetic and Breakdancers. For the sequence with most texture, the Horse, this value is equal to 11. The second constant parameter for the same subsystem is the aperture size required by the Adaptive Canny Edge Detector. Despite the smoothing module, this value is the same for all sequences and is set to 3. For the next subsystem, the Depth Map Fitting, two more parameters are needed. The first one is the "maximum block layer size" and the other is the "number of allowed recalls". The first one is equal to 64 and the second one, as previously introduced in section 4.1.2, is set to 3. Both are used to set the biggest block size correctly with respect to the input resolution. Sequences with a resolution of 1024 × 768 are

| Sequence | Smoothing | Edge det. | Recalls | Block sizes |
|---|---|---|---|---|
| Ballet, Book and Newspaper | 3 | 3 | 3 | {64,32,16,8} |
| Breakdancers and Synthetic | 5 | 3 | 3 | {64,32,16,8} |
| Horse | 11 | 3 | 3 | {40,20,10,5} |

**Table 4.4:** *Global parameters for all sequences and all working options*

divisible by 64 (in width and height), and therefore the maximum block size is 64. However, for the Horse sequence, which resolution is 1920 × 1080, the height is not divisible by 64. In order to avoid this conflict, the system computes the biggest block size automathically depending on "the number of allowed recalls". In the case of the Horse sequence, the biggest block size that fits to its resolution is 40. Table 4.4 sums up these values for all sequences.

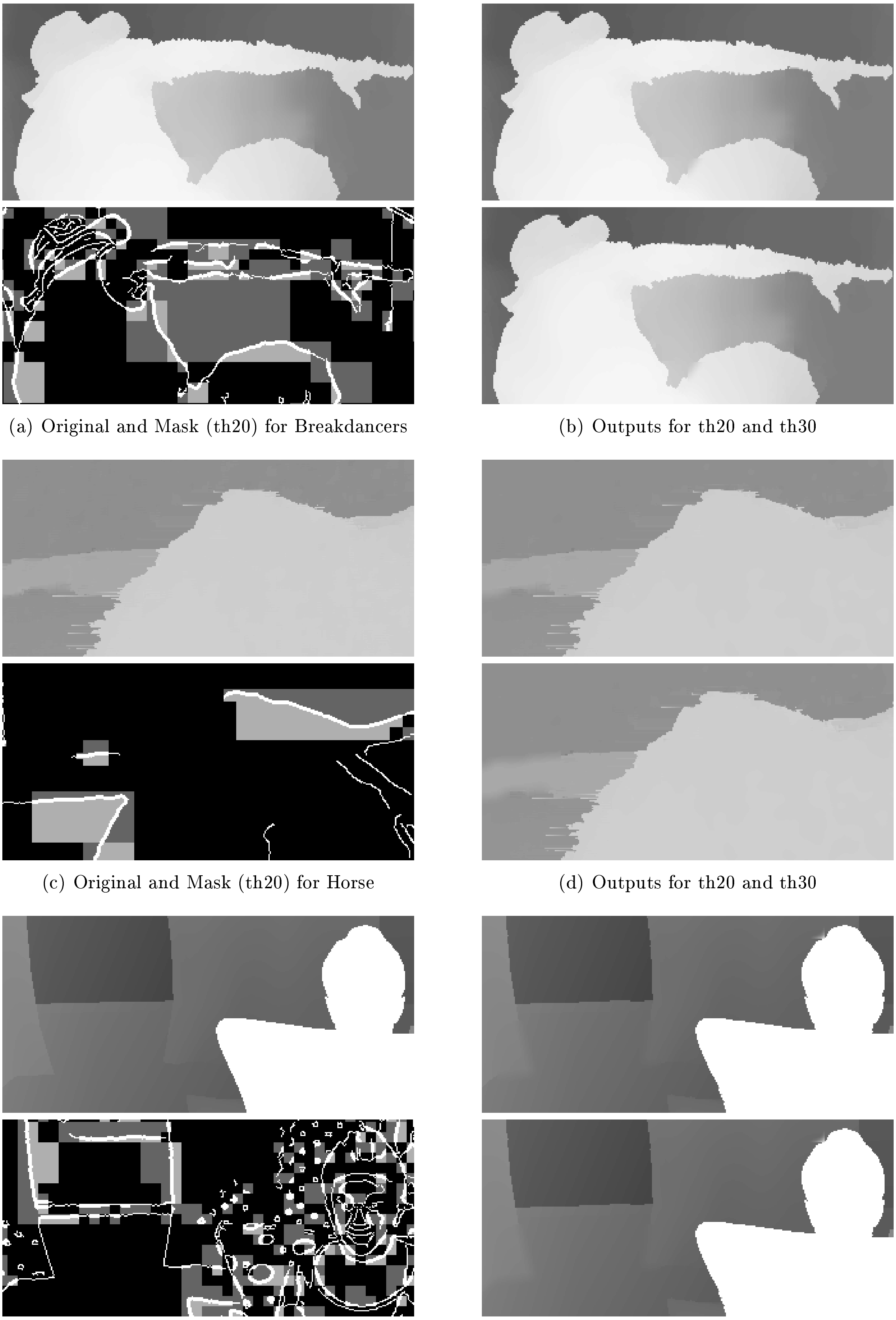
Next, the results are analyzed according to the selected working option. Note, that due to large image resolutions, detail pictures of the results are shown.

Option1 is launched 11 times with different variance thresholds covering the range $[20, 30]$. The Depth blurring module prepares just one smoothed depth map for non-suitable blocks. In this case, the aperture size required for the smoothing process is set to 9. For figure 4.12, two selected thresholds are chosen for highlighting the results. According to the graphic in figure 4.11, the selected thresholds are 20 and 30, as they present the maximum variation.

4 different samples of result images for 3 MVD sequences (Breakdancers, Horse and Synthetic) are shown in figure 4.12. The structure of this figure is the same for each sequence. On top-left the original depth sample of the sequence. On bottom-left the edge mask with dilated lines and the labeled regions with two different grey levels. Note, that for similar blocks with a line crossing vertically, left region could have a different grey level than right region. It is a mere criteria to make regions visible. On top-right the output when chosing a threshold of 20. Finally, on bottom-right, the output for a threshold of 30.

Looking at the presented results, some improvements and some changes can be seen. First, Breakdancers sequence present good results at re-fitting silhouettes. As an example, the arm of the dancer has less jittering than the original for both outputs. It also happens to Horse sequence, although most of the jittering is wider than 1 pixel. In this case, most of the silhouete remains with a noisy contour, as it is explained at the end of section 4.1.2. It is worth mentioning, that in all cases, the re-fitted contours have the same shape, although the thresholds are different. This is due to the low variance (of depth values) of each region in blocks containing an edge and therefore the variance of each region is always below the threshold.

In both sequences a part of the background is smoothed. Particularly, Synthetic reveals some changes, although the desired response would be an invariant behavior. For both outputs of this sequence, the silhouette of the courtains is smoothed with the wall, and the result is a blurred profile in this region. Another interesting effect is a grey angle on top-left of old woman's head. It is easy to understand how a shape like that could appear in this part of the image. This angle corresponds to a bottom-right corner of a block that was labeled as non-suitable (because its variance is below the variance threshold). Thus, it appears in both outputs. Consequently, when composing the new depth map, the composer takes the corresponding block of the smoothed depth map. By using a bigger aperture size, more values are included in the average. Hence, foreground values (old woman) are mixed with values of the background (wall). Finally, note that the contour of the armchair is a bit distorted, caused by the re-fitting process (this process sets the maximum value of the 4-pixel neighborhood to each pixel for the Line region).



(a) Original and Mask (th20) for Breakdancers

(b) Outputs for th20 and th30

(c) Original and Mask (th20) for Horse

(d) Outputs for th20 and th30

(e) Original and Mask (th20) for Synthetic

(f) Outputs for th20 and th30

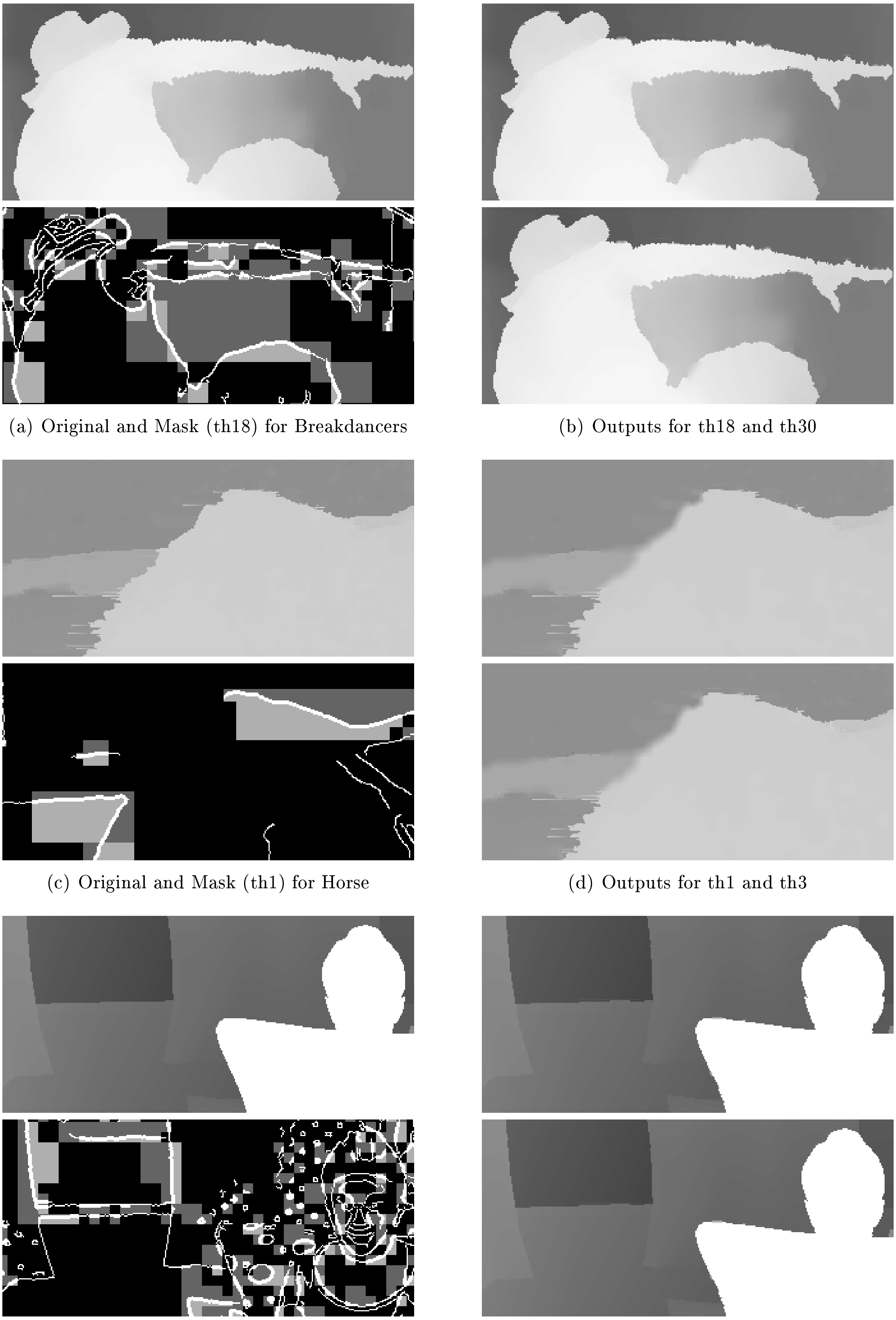
**Figure 4.12:** *Results for Breakdancers, Horse and Synthetic chosing Option1. (a), (c), (e) are original depth maps and their mask with regions marked before refitting. (b), (d), (f) are results for variance threshold 20 (up) and 30 (down). Results are post-processed by changing brightness and contrast for a better visualization.*

Option2 is launched twice with different thresholds for each sequence and according to a fixed tolerance (see section 4.2). Unlike Option1, the Depth Blurring module prepares four different smoothed depth maps, namely hierarchical depth smoothing, whose kernel sizes are 9, 7, 5 and 3. Then, the Composer module takes the corresponding block from the smoothed depth map according to the relation between block size and kernel size (as presented in table 4.1) for those blocks that have not been improved. Figure 4.13 again presents result details for Breakdancers, Horse and Synthetic MVD sequences.

First, note that the Horse and the Synthetic sequences use low thresholds (1 and 3 for the Horse and 1 and 2 for the Synthetic). This has a certain impact on background areas, because just those blocks that have a very low variance are selected as non-suitable blocks. Consequently, less background and/or foreground areas are smoothed. In this case, it improves some strange effects (the corner over the old woman's head), that occured for the Synthetic sequence. Looking at figure 4.13, silhouettes are equally re-shaped as done for Option1. In this sense, the re-fitting process is independent of the selected option , but not the smoothing process. Due to printed format constraints, the effects of using the hierarchical depth blur are not visible enough.
As occured for Option1, the results of the Synthetic sequence has some changes respect to the orginal input. First, some contours are a bit shifted due to the re-fitting process. Hence, contours of the output do not exactly match with the input. On the other hand, less background and foreground areas are smoothed. However, due to the use of a hierarchical depth smoothing, some "skipped" blocks are visible in the output. Specially, in the case of the Horse sequence both outputs present a blurred contour at the horseback. To a minor degree, this also happens to the Synthetic sequence, such as the armchair contour, specifically at the top-left of it. Here an $8 \times 8$ block contains no edge information and consequently the composer takes the corresponding block of the blurred depth map (with a kernel size equal to 3). However, the aim of a hierachical depth blur is to smooth background areas and not depth transitions. In this case, due to an undetected color edge, the system fails at composing the final depth map. It is therefore proven that Option2 modifies depth maps more than Option1.

Finally, Option3 uses the same parameters as Option2 but it lists blocks as suitable by edge criteria. Figure 4.14 shows the results.

Unlike Option2, this configuration analyzes all blocks that have edge information, ignoring the quantity of variance of depth values. Thus, the masks shown have more marked regions than in Option1 and Option2. By using the same thresholds as for Option2, outputs are highly similar. Just that blocks that are listed as suitable for Option3 but not for Option2 are re-fitted. As an example of this, the pelvis of Breakdancers results (figure 4.14(b)) is re-fitted (in Option2 this part is smoothed). However, blocks that are listed as suitable for Option3 present a low variance and probably they belong to very flat regions. Hence, changes are not that significant in those areas.



(a) Original and Mask (th18) for Breakdancers

(b) Outputs for th18 and th30

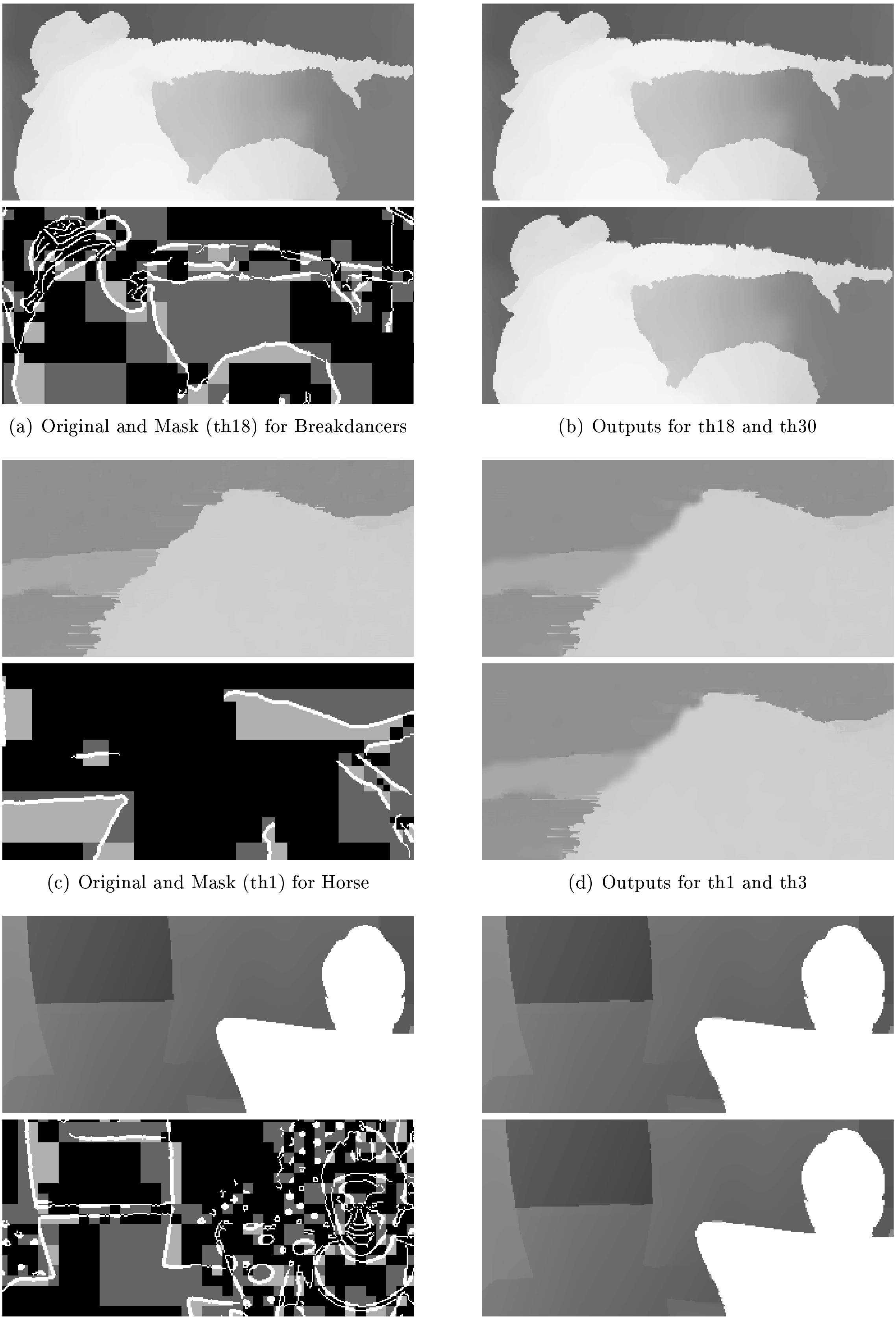
(c) Original and Mask (th1) for Horse

(d) Outputs for th1 and th3

(e) Original and Mask (th1) for Synthetic

(f) Outputs for th1 and th2

**Figure 4.13:** *Results for Breakdancers, Horse and Synthetic chosing Option2. (a), (c), (e) are original depth maps and their mask with regions marked before re-fitting. (b), (d), (f) are results for tolerance 10 (up) and 5 (down). Results are post-processed by changing brightness and contrast for a better visualization.*



(a) Original and Mask (th18) for Breakdancers

(b) Outputs for th18 and th30

(c) Original and Mask (th1) for Horse

(d) Outputs for th1 and th3

(e) Original and Mask (th1) for Synthetic

(f) Outputs for th1 and th2

**Figure 4.14:** *Results for Breakdancers, Horse and Synthetic chosing Option3. (a), (c), (e) are original depth maps and their mask with regions marked before re-fitting. (b), (d), (f) are results for tolerance 10 (up) and 5 (down). Results are post-processed by changing brightness and contrast for a better visualization.*

# Chapter 5

# Results

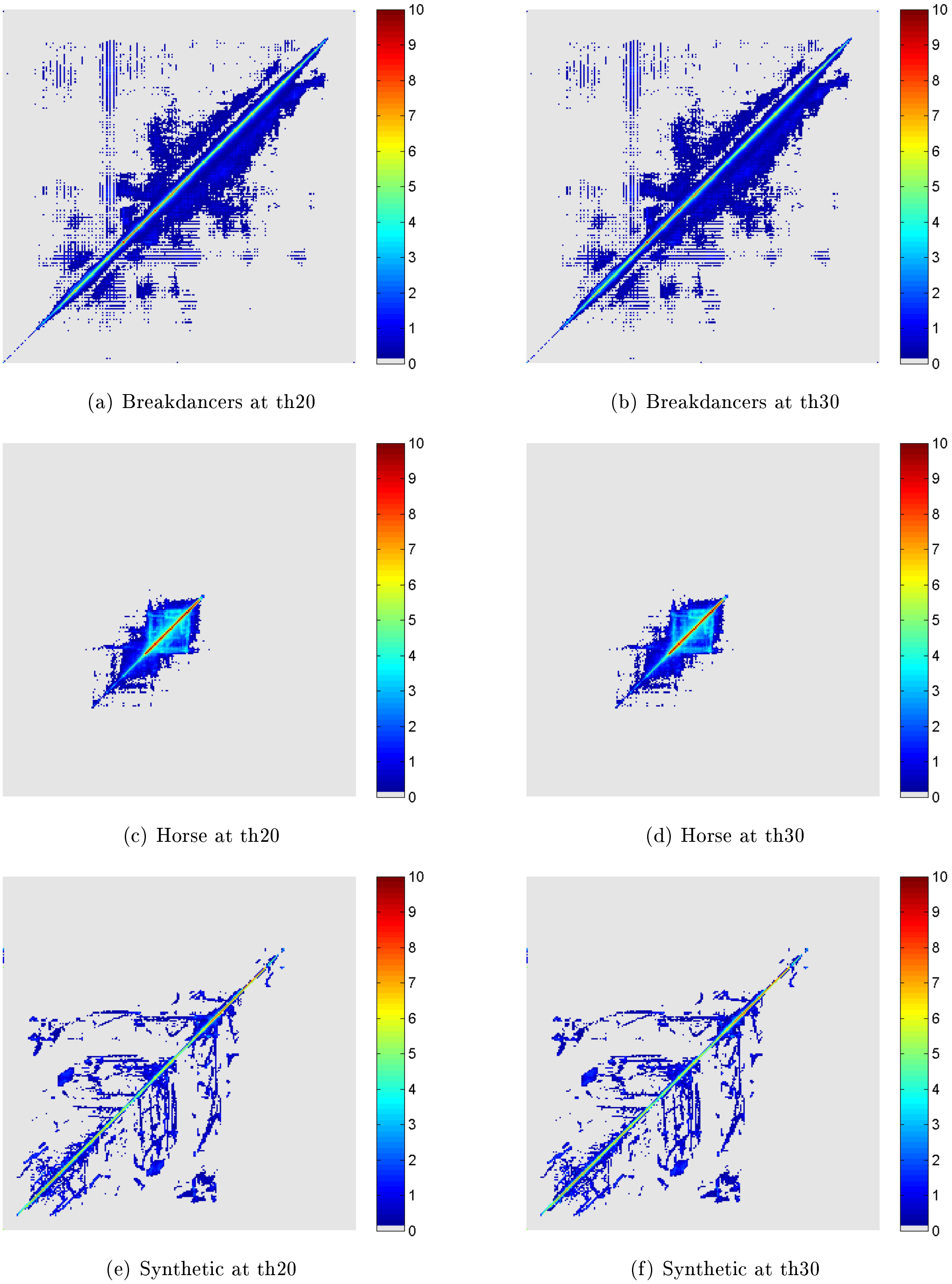
For this chapter the six MVD data sets have been analyzed after applying the improvement algorithm presented in chapter 4. The synthetic sequence is again the ground truth depth information, as it presents consistent results when using correlation histograms. Moreover, the output should be invariant when the input is the synthetic sequence. On one hand, results in chapter 3 analyze the six MVD sequences with correlation histograms (and also with histogram and spectrum techniques), highlighting differences between color and depth components, and between temporal and spatial references. On the other hand, chapter 4 presents qualitative results of improved depth component. Consequently, it is reasonable to use the new methods introduced in chapter 3 to analyze the output of the system presented in chapter 4. The results are organized according to the three working options described in chapter 4. For each working option, temporal and spatial correlation histograms are presented in order to evaluate changes in the output of the system. The selected pair of pixels is the same as for th results in chapter 3. Therefore, definitions given in section 3.3 will be used to describe the outputs.

As done in chapter 3, results are equalized regarding the different resolutions, number of cameras and just for first 25 frames of the given data sets. Three selected sequences (Breakdancers, Horse and Synthetic) are chosen to be analyzed. The results of these sequences can be found in this chapter. The results of the other data set (Ballet, Book and Newspaper) can be found in the Appendix.

1. **Option1**

   As mentioned in chapter 4, this option is the one of the three options that modifies the content of the depth maps less than the other two options. In this case, the overall content of improved depth maps is pretty similar to the original input. Therefore, the spatial and temporal correlation histograms of the output of Option1 are expected to be pretty similar to original depth maps. Next, outputs of spatial correlation histogram are presented and later the ones for temporal correlation histogram.

   Figure 5.1 shows the spatial correlation histogram for outputs of Option1. Left column of images corresponds to threshold 20 and the right one to threshold 30. First, it would be interesting to mention some particular characteristics of this option. The main difference



(a) Breakdancers at th20

(b) Breakdancers at th30

(c) Horse at th20

(d) Horse at th30

(e) Synthetic at th20

(f) Synthetic at th30

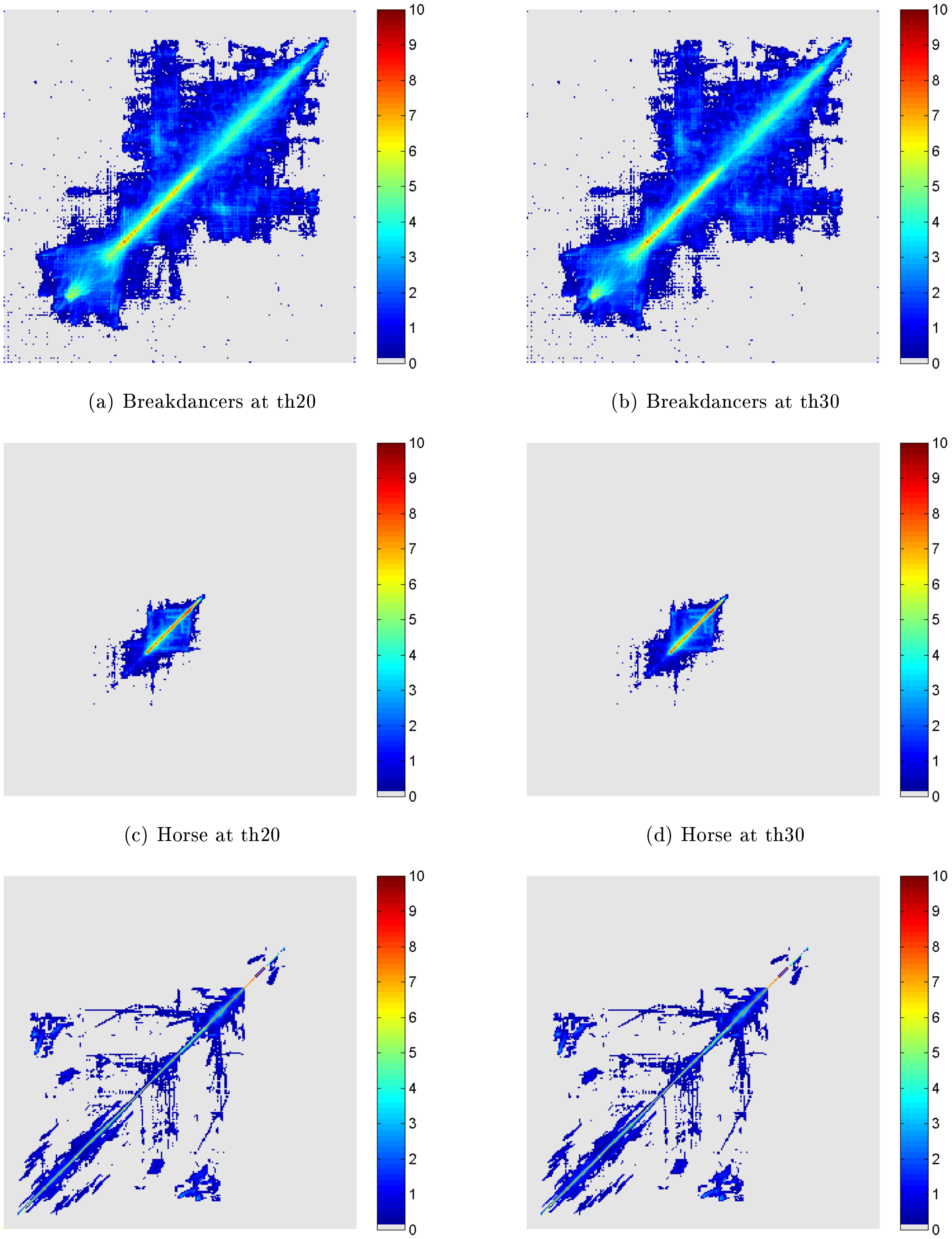
**Figure 5.1:** *Spatial Correlation Histograms for Breakdancers, Horse and Synthetic sequences after improvement with Option1 and thresholds at 20 and 30.*

between depth map outputs with th20 and th30 is the number of non-suitable blocks (flat blocks, meaning background and foreground) that consequently are smoothed with a Gaussian filter. Loosely speaking, the higher the threshold, the higher the number of smoothed background/foreground blocks. Hence, smoothed blocks will get new values slightly different to the original. Another characteristic is that silhouettes that have been improved have a 1 pixel-width jittering. Therefore, due to algorithm's behavior, just values around the contour have been modified, representing a low number of adjusted pixel. According to this explanation, the diagonal and quadrants I and IV are regions that show more changes in the correlation histograms in figure 5.1. In contrast, lines describing depth transitions (quadrants II and III) keep almost invariant regarding to original outputs. A proof of that is the Breakdancers sequence. Comparing th30 and th20, there is less presence of single spots in quadrant I due to the filtering. To a minor degree it also happens in quadrant IV, but due to printed format it may not be visible enough. The Horse sequence also illustrates this effect (figures 5.1(c) and 5.1(d)). In this case, values around the diagonal region in th30 are more concentrated than in th20. Furthermore, some single spots in quadrants I and IV also desappear, getting absorbed by their closest patch region. Synthetic keeps almost invariant, although it has been seen in chapter 4 that the Synthetic sequence also suffers some change. However, the impact of the improvement process with the synthetic reveals low changes for this sequence. Therefore, the improvement of depth maps with Option1 has a very good invariant behavior when applying it to ground truth depth information.

For the remaining sequences (see figure A.7), Ballet sequence presents similar characteristics as described for breakdancers. However, Book and Newspaper show some interesting effects. Both sequences present a original spatial correlation histogram with single spots due to the non-use of available depth values. When applying the filter to their depth component, their values change. This effect can be easily identified with the continuity around the diagonal region as it has been previously foreseen. Some continuous linear patterns describing depth transitions (instead of discreet linear patterns) also appear in both outputs. As previously described, th30 outputs have more smoothed depth blocks and consequently their region around the one-pixel diagonal is bigger than for th20.

Temporal correlation histograms for Breakdancers, Horse and Synthetic sequences are shown in figure 5.2. As commented in chapter 3, temporal correlation histograms should present a straight diagonal line with no distortion in quadrant I. However, natural sequences did not present this characteristic. Quadrant I in Breakdancers and Horse sequences present a patch region instead of the desired line. With the same argument as for spatial correlation histograms, outputs with th30 have more smoothed areas than outputs with th20. Hence, values tend to be more accumulated in the diagonal for outputs with th30 than th20. This is the case for all sequences. However, the synthetic presents results that are practically identical. Just quadrant I is modified from th20 to th30, having a less concentrated area around the 1-pixel width diagonal for th20. Note, that first 25 frames of this sequence present more motion in the background than in the foreground. Therefore, a patch region appears in quadrant I around the diagonal line.

Figures in the Appendix (A.8) show akin characteristics as the afore described for Book and Newspaper. Here, the appearance of linear patterns representing interaction between the background and the foreground is more visible. Th30 also shows more concentration around the diagonal than th20, which is consistent with the other results.



(a) Breakdancers at th20

(b) Breakdancers at th30

(c) Horse at th20

(d) Horse at th30

(e) Synthetic at th20

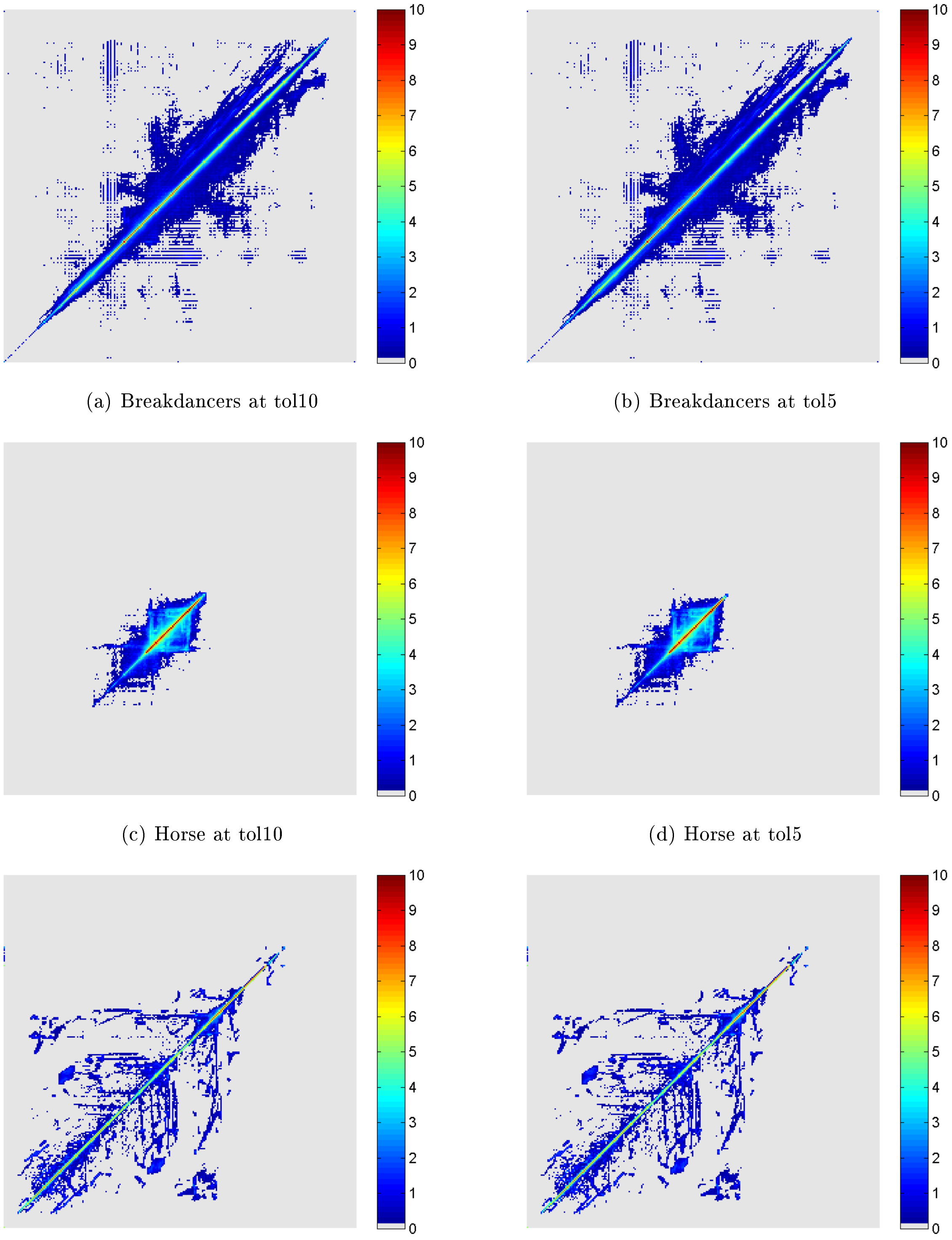
(f) Synthetic at th30

**Figure 5.2:** *Temporal Correlation Histograms for Breakdancers, Horse and Synthetic sequences after improvement with Option1 and thresholds at 20 and 30.*

2. **Option2**

The general description of spatial correlation histograms done in Option1 is also useful for Option2. Nevertheless, the output should be interpreted a bit different. As Option1 just smoothes the non-suitable blocks, Option2 uses the hierarchical depth blur for "skipped" blocks. Hence, more blocks are smoothed, leading to a more different output than the one obtained with Option1. This implies that spatial and temporal correlation histograms are more different, by concentrating more values in the diagonal region. Figure 5.3 shows the results for spatial correlation histograms. The Horse sequence is the one that illustrates the commented effects best. For a tolerance of 10% (for the Horse equivalent to a threshold equal to 1), the top of the diagonal presents a wider region. However, for a tolerance of 5% (equivalent to a threshold equal to 3), this area presents a sharp line, meaning that values of the foreground have been smoothed. Changes are also visible in Breakdancers' results, specially in quadrant I. In this sequence, the output for a tolerance of 5% (threshold 30) presents less single spots than choosing a tolerance of 10% (threshold 18). However, what is worth mentioning, is the region above the diagonal. In Option1 as well as the original, Breakdancers' output lacks some values above the diagonal (see 5.1(a) or 5.1(b)). In Option2, this region is filled with blue values (according to the color map of the representation), resulting from the hierarchical blur. As tolerance 5% is equivalent to threshold 30, changes from 5.1(b) to 5.3(b) are uniquely caused by choosing the hierarchical smoothing. Due to the small difference in threshold values for synthetic sequence (a tolerance of 5% corresponds to th2 and the 10% to th1), its outputs are almost identical. Nevertheless, it should be mentioned that values around the diagonal are a bit more dispersed than the original.
The remaining sequences, that can be found in the Appendix (A.9), present similar results. Ballet sequence, as it occured with Option1, has a similar behavior as Breakdancers. A region with no correlated values above the diagonal (in Option1) is filled with a blue area for this working option. As said, this is due to the use of hierarchical depth smoothing. This effect can be also seen in Book and Newspaper, where a wider area around the diagonal is present, especially in quadrant IV.

In the case of temporal correlation histograms (figure 5.4), differences between choosing a tolerance of 10% or 5% are less than for the spatial. Having a look at Breakdancers, only some single spots are different. The whole outputs are highly alike, which also applies to the Synthetic. However, the Horse sequence present the same effect as described for the spatial correlation histograms output. Again, the area at the top of the diagonal is bigger for a tolerance of 10% than for a tolerance of 5%. The reason is the same: the use of the hierarchical depth smoothing.
Compared to Option1, values from quadrant I tend to be more concentrated in the diagonal. However, this region is still too big to ensure temporal consistency in the background, e.g. for Breakdancers. The algorithm working with this option has improved depth maps, but the desired shape is still different to the synthetic one. The Horse sequence also presents better results for this option, where values are more concentrated around the diagonal and lines are a bit sharper. This sequence presents very low motion and hence frames should be highly correlated. Consequently, depth maps should be correlated as well and, in terms of temporal correlation histograms, a sharp diagonal (with a huge concentration of values in it) and some sharp lines would appear in the output of this sequence, as it is the case for the synthetic sequence. However, the output of temporal correlation histogram for the original set does not present this characteristic. After applying the algorithm, the output of the 2D Histogram is more similar to the desired



(a) Breakdancers at tol10

(b) Breakdancers at tol5

(c) Horse at tol10

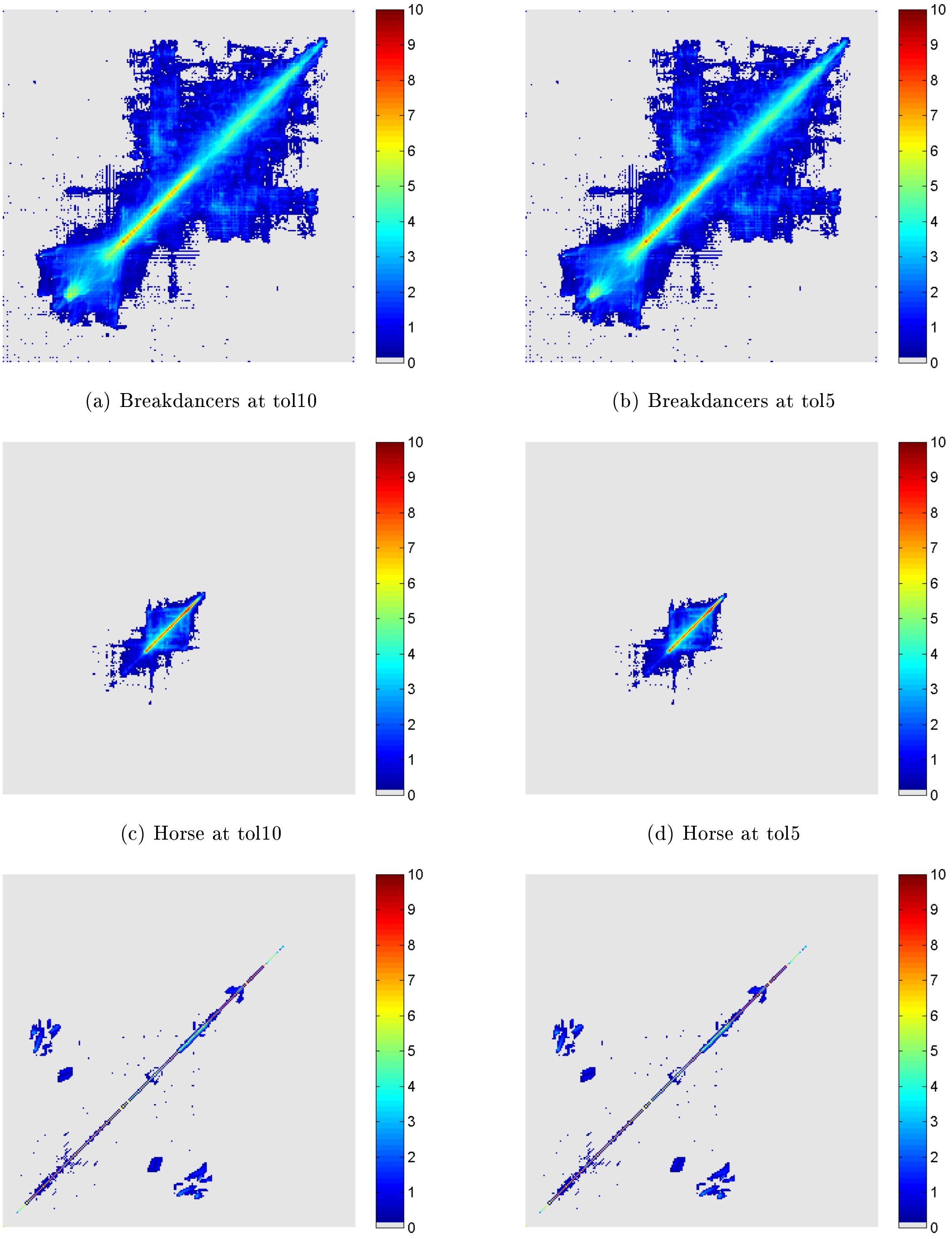
(d) Horse at tol5

(e) Synthetic at tol10

(f) Synthetic at tol5

**Figure 5.3:** *Spatial Correlation Histograms for Breakdancers, Horse and Synthetic sequences after improvement with Option2 and tolerances 10 and 5.*

than the original one (see 3.12(d)). Finally, the synthetic presents better results than in Option1. The diagonal at quadrant I is tighter than in Option1, and therefore it is more consistent in a temporal sense.
In Book an Newspaper (see A.10), the region around the diagonal is wider in both cases for a tolerance of 5% than for 10%. As it happened with Option1, a higher variance, leads to a higher number of smoothed blocks. Hence, regions are wider but values are more concentrated in the diagonal. Finally, the output for ballet sequence almost keeps invariant. As it was previously said, it has a similar behavior as the breakdancers sequence.



(a) Breakdancers at tol10

(b) Breakdancers at tol5

(c) Horse at tol10

(d) Horse at tol5

(e) Synthetic at tol10

(f) Synthetic at tol5

**Figure 5.4:** *Temporal Correlation Histograms for Breakdancers, Horse and Synthetic sequences after improvement with Option2 and tolerances 10 and 5.*

3. **Option3**

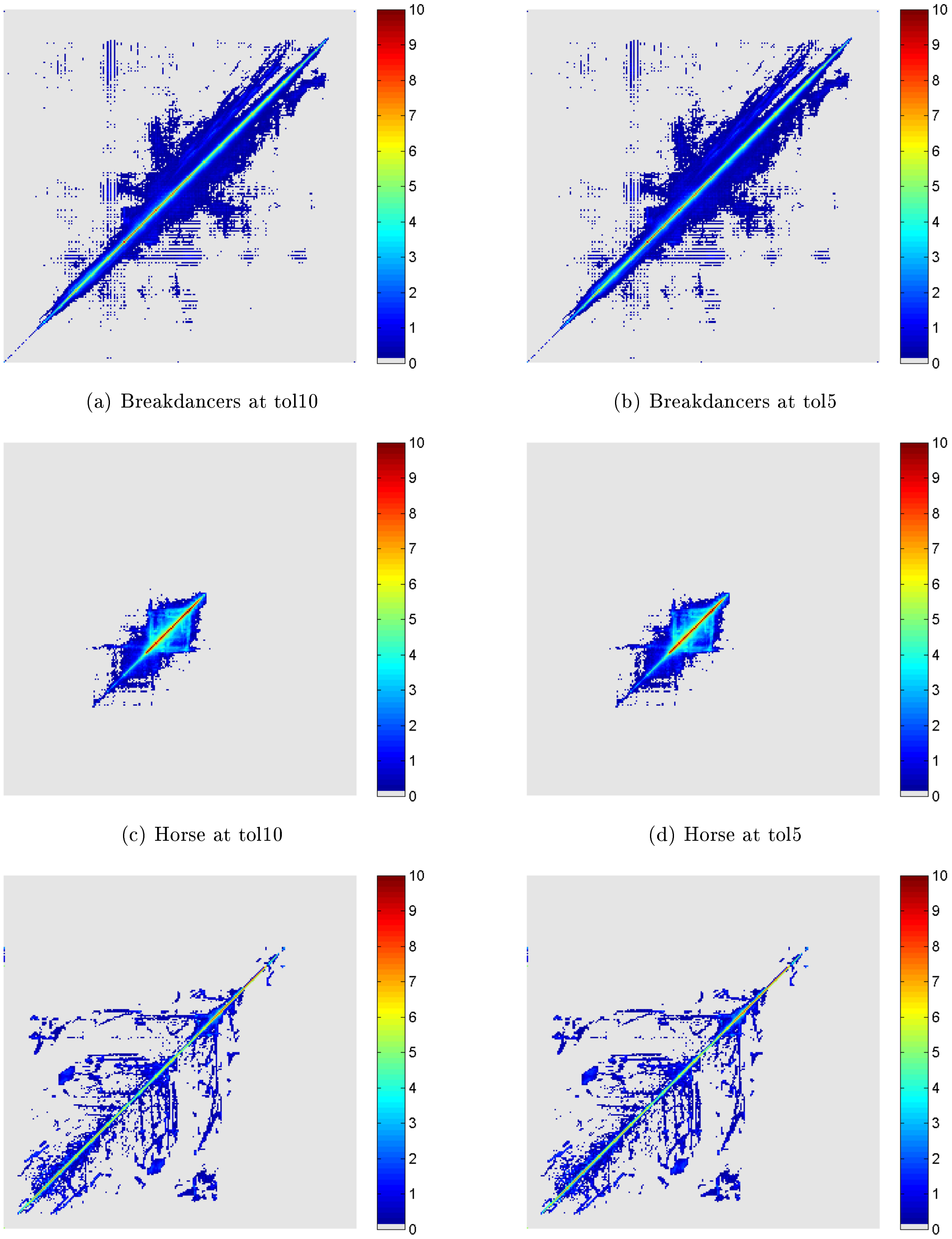
   As explained in chapter 4, the difference between Option2 and Option3 is the List of suitable blocks. For the first case, the calculus of the variance is the method used to threshold between suitable and non-suitable blocks to be processed. For the second, this list is created according to edge information, by setting the suitable label to all those block whose corresponding block in the edge mask contains edge information. Otherwise, if the block of the edge mask does not contain any edge pixel, it is set as a non-suitable block. Hence, the main difference between Option2 and Option3 is the number of suitable blocks. Thus, Option3 has more or at least an equal number of suitable blocks than Option2. This behavior is already justified in section 4.2. In general terms, less blocks will be smoothed, but more of them will be re-fitted. A proof of this is the Horse sequence. With this working option the top of the diagonal region keeps invariant for both tolerances, while in Option2 the top got smoothed. As less blocks are smoothed, values are less accumulated to the diagonal. What can be seen again, is the influence of the hierarchical depth smoothing. For Breakdancers sequence, the blank patch above the diagonal is filled with blue values. Thus, it is demonstrated that during the process subblocks are smoothed according to the external variance threshold. Synthetic sequence presents no change from tolerance 10% to 5%, and, on top of that, the output is also invariant from Option2 to Option3.

   The explanation done in Option2 for the rest of sequences (see figure A.11) is also valid for this working option. Here, Ballet has a similar behavior as the Breakdancers sequence (referring to this option, obviously). On the other hand, Book and Newspaper sequences present a wider area around the diagonal due to the smoothing of some blocks.

   Figure 5.6 shows the results for temporal correlation histograms. Looking at the presented outputs, no change between choosing a tolerance of 10% or 5% can be seen. In fact, it means that much less foreground and/or background regions have been smoothed. Hence, results of Option3 should be more similar to the original temporal correlation histograms than the outputs of Option2.

   An interesting sequence to analyze is Book. As it happens to all sequences, the difference from choosing a tolerance or another in this option is little. However, it is interesting to see the difference between choosing Option2 and Option3. In Option2, this sequence presented values that are concentrated around the diagonal in quadrant I (see figure A.12(d) and A.12(c)). Nevertheless, due to less smoothing, values are more dispersed in this region, being more similar to the original than to Option2.

Three different working options have been presented in chapter 4 and have been analyzed here according to the introduced tools in chapter 3. Basically, changes at the outputs are more visible when a certain number of blocks has been smoothed. As the re-fitting process tries to preserve depth information by modifying depth transitions, re-shaped blocks do not affect to the correlation histograms output as much as smoothed blocks do. Hence, the algorithm acts more or less aggressive depending on the chosen criterias.



(a) Breakdancers at tol10

(b) Breakdancers at tol5

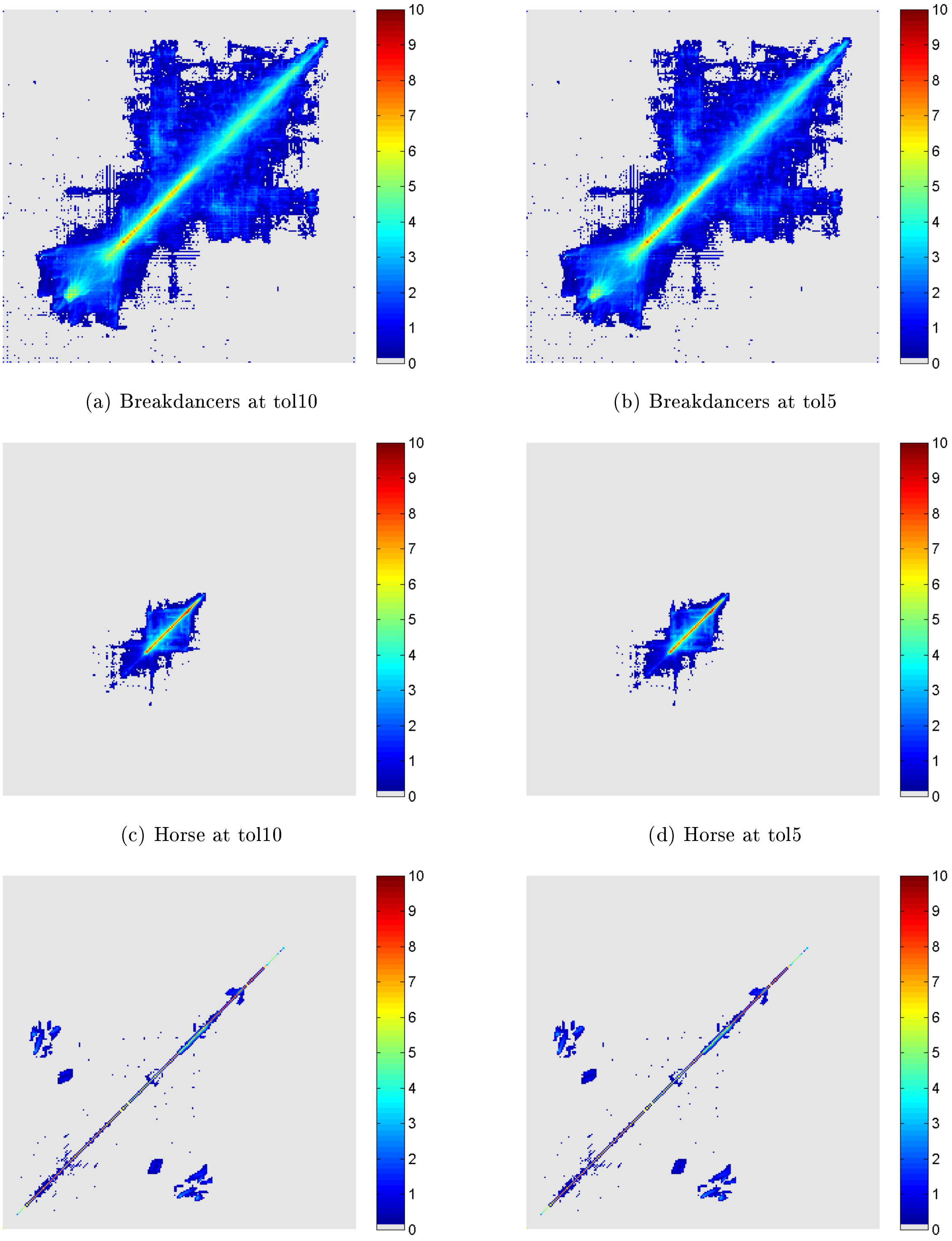
(c) Horse at tol10

(d) Horse at tol5

(e) Synthetic at tol10

(f) Synthetic at tol5

**Figure 5.5:** *Spatial Correlation Histograms for Breakdancers, Horse and Synthetic sequences after improvement with Option3 and tolerances 10 and 5.*



(a) Breakdancers at tol10

(b) Breakdancers at tol5

(c) Horse at tol10

(d) Horse at tol5

(e) Synthetic at tol10

(f) Synthetic at tol5

**Figure 5.6:** *Temporal Correlation Histograms for Breakdancers, Horse and Synthetic sequences after improvement with Option2 and tolerances 10 and 5.*

# Chapter 6

# Summary and Conclusions

This work presented a framework for improving MVD data, obtained by a multi-camera system and finally used for a multiview display. MVD data is a compound of color video plus per-pixel depth information. Depth maps are obtained by estimating them from the color information with different methods. However, they present some errors and imprecissions. The existence of these errors is problematic when using the depth information to render a stereo video pair, and results in a misplacing of the content of the scene when rendered. Hence, this project aims at improving these errors.

This work starts by describing MVD data. Once MVD data is described, it is analyzed in order to obtain insight into similarities and differences of the color and depth characteristics. For this, known methods, as histogram analyis and spectrum of an image, are used to analyze six MVD sequences (five natural sequences and a synthetic one). Moreover, a new method called correlation histograms is introduced. It has been shown that for spatial as well as temporal correlation histograms, color and depth components have considerably different characteristics, as video represents the color texture and depth the geometry of the scene. Hence, spatial correlation histogram outputs for color present a continuous region around the diagonal. Applying this method to the depth, outputs show high concentration on and around the diagonal, resulting from flat regions, and additional frayed and discrete areas, resulting from depth transitions. By using the temporal correlation histogram, outputs of the color component present once again continuous regions around the diagonal. As already pointed out, they could be a qualitative measure of the degree of motion in a scene. By applying it to depth maps, it was observed that they lack temporal consistency in background and foreground areas when compared to the characteristics of the synthetic data (used as ground truth depth information).

The aforementioned errors that cause abnormal visual effects are of two types. The first is the non-correspondence of depth transitions with silhouettes present in the color component. The second is the temporal inconsistency, as previously introduced. In order to address both errors, a system based on color edge detection for depth map re-fitting is designed. The main properties of this system are modularity and flexibility. Firstly, it is modular in that, it is a compound of sets and subsets which have specific targets. Secondly, it is flexible in that it accepts different configuring options, and allows modules to be implemented in another way if their target is respected. As a result, the same system can be designed differently, but following the same processing chain.

The system is divided in two subsystems. The first is the Color Edge Detection that is in charge of yielding an edge mask to the subsequent subsystem. The Color Edge Detection is composed of two modules. The first is a smoothing filter which avoids a high number of wrong miss-detections. The second is an adaptive Canny edge detector. As explained in section 4.1.1, the first module is very important, as false edge detections are preferred over missing structual information. The system presented is implemented with a Gaussian filter which external parameter is set depending on the input. However, it can be improved with a Bilateral filter. The disadvantage of this kind of filter is that it needs more external parameters. In order to give dynamic behavior to the system, a texture pre-analysis is proposed. This step can be achieved by analyzing the spectra of the image or by computing a statistical parameter from the spatial correlation histogram. Future investigations can put effort on finding a parameter (instead of a pattern) representative of the amount of texture in an image.
The second subsystem, the Depth Map Fitting, reshapes the silhouettes by using the edge information extracted by the first subsystem. Here, silhouettes with 1 pixel-width jittering are improved. This width (1 pixel) covers the majority of the jittering present in the data set. Additionally, a solution for any kind of jittering is proposed. This solution is based on finding the region between edges from color video and edges obtained from depth transitions. Hence, all depth transitions are adapted to color silhouettes. Moreover, this subsystem also smoothes plain blocks that are intended to be background and/or foreground. To determine which blocks will be smoothed, three different working options have been defined. Two of them are based on a variance criterion, while the third is based on the presence of edge information. The evaluation presented in this work reveals high correlation between edges in color video and depth maps. Blocks labeled as background or foreground also have a different treatment depending on the working option. The first option copies values of the original depth map directly. This criteria avoids distorting the input to a minor degree. The second and the third option use a hierarchical depth smoothing for these background or foreground blocks, meaning low variance for the second option and edge information for the third option.

Each working option showed different characteristics in the output. For the first, more blocks were labeled as background or foreground by setting a high threshold. Therefore, some depth edges were blurred by applying a rough smoothing and consequently the representation of the 3D geometry of the scene worsered, because edges in depth maps must be sharp instead of smooth transitions. Consequently, the variance threshold can not be set as a constant paremeter for any sequence. On the other hand, the non use of a hierarchical depth smoothing distorts the input to a minor degree. It can be a good solution for sequences showing low temporal variations in background and foreground regions (i.e. the Horse sequence). In contrast, sequences that require improvements on these regions (i.e. Ballet and Breakdancers) would need more than a rough smoothing. Therefore, it is crucial to find a combination that satisfies the trade-off between re-shaping contours and improving background and foreground areas.
In order to improve sequences with temporal inconsistency in the background and foreground regions, Option2 is based on a hierarchical depth smoothing. By applying this criterion, more blocks labeled as foreground or background during the process are smoothed as well, which roughness depends on the size of the block. Hence, it can be concluded that this option modifies the input more than the first one. Moreover, this option is also based on the variance criterion. However, variance thresholds are set depending on each input sequence (taken from the study presented in section 4.2). The values that have been used resulted in similar thresholds for 3

sequences (Ballet, Book and Breakdancers) as the thresholds taken in Option1. Nevertheless, the values used for testing Option2 for the other 3 sequences (Horse, Newspaper and Synthetic) were different as from those Option1. This implies that more blocks were taken into account for processing and therefore less background and foreground blocks have been smoothed. In this case, it is a good progress, as Synthetic and Horse are preferred to have less smoothed background and foreground areas (because they do not show significant temporal inconsistencies) than Ballet or Breakdancers (background and foreground areas must be improved). Consequently, the aforementioned trade-off is achieved by setting the variance threshold depending on the input of the image.
Finally, Option3 is based on edge information for deciding if a block belongs to a background or foreground area or not. This criteria is based on the color component, instead of the variance method, or in other words, the depth component. As Already justified, this option processes more blocks than the other two, meaning that this option focuses more on re-shaping the depth contours than improving background and foreground regions. For a good behavior of this working option, it is very important to have an edge mask with structural information about object contours and (ideally) no false edge detections (mainly due to rough textures). To ensure this, the sequences were filtered with a Gaussian filter, for which the aperture size was set depending on each sequence. Hence, each sequence presented a good edge mask for this working option. Therefore, this option is a very good solution for labeling background and foreground blocks if the edge mask is robust against false edge detections. Otherwise, textured regions are treated as edges and consequently the corresponding depth blocks are processed instead of being smoothed. On the other hand, the hierarchical depth smoothing criteria is preserved and therefore it acts on blocks labeled as background or foreground as in the second option. Consequently, the trade-off is once again achieved.

These options lead to the conclusion that the choice of the working option will always depend on the data to be improved according to the aforementioned trade-off. From these three options, the third one presented best results from the six sequences with different characteristics. This criteria relies more on color component (100% true information) than on depth component (which are errorneous and imprecise). The challenge that this option proposes is to find a good mechanism for obtaining a robust edge mask. If this mechanism is not found, the second option attains to the system's target, as edges from color video are highly correlated with depth maps.

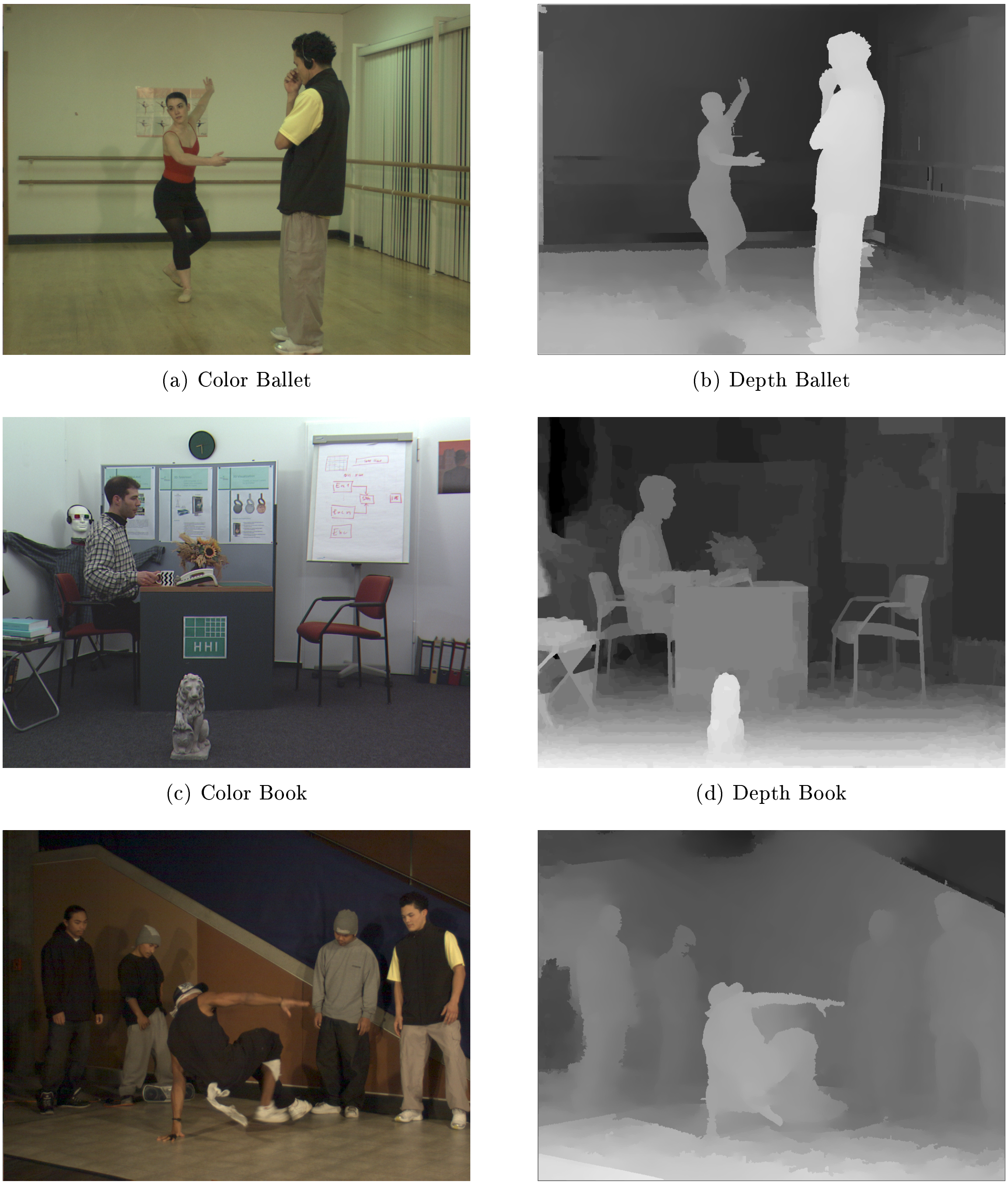
Results from evaluating the six MVD sequences using the system, reveal that this new tool is good for evaluating the quality of the new depth maps and the impact of changes. On the other hand, qualitative results are shown in order to highlight the success of the presented system. In summary, the presented analysis techniques describe the MVD data and the system improves the depth component of MVD.

# Appendix A

# Appendix

### Explanation of the MVD sequences

The used data set is a total of 6 MVD sequences, five natural and one synthetic. In order to describe the motion in those sequences, a brief explanation of them is presented, instead of attaching all frames. Ballet sequence (figures A.1(a) and A.1(b)) is a indoor scene with a person at the foreground who keeps static and the ballet dancer that creates more motion. In Book Arrival set (figures A.1(c) and A.1(d)), the person who is sitting first stands up to shake his hand with a man who comes into the scene. The other dancers sequence, the Breakdancers (figures A.1(e) and A.1(f)), presents four young men standing up, with little motion, watching a fifth one who is breakdancing and, therefore, presenting fast motion. The only sequence that is outdoor is Horse (figures A.2(a) and A.2(b)) where the animal does not move significantly. The last natural sequence Newspaper (figures A.2(c) and A.2(d)), shows two young persons reading a newspaper who welcome a third person (who creates more motion), but without standing up. Finally, the synthetic sequence (figures A.2(e) and A.2(f)) is a computer graphics rendered sequence where an old woman is sitting totally static and the motion created in the scene is done by the floating balls, moving from the background to the foreground and some objects moving on the floor.



(a) Color Ballet

(b) Depth Ballet

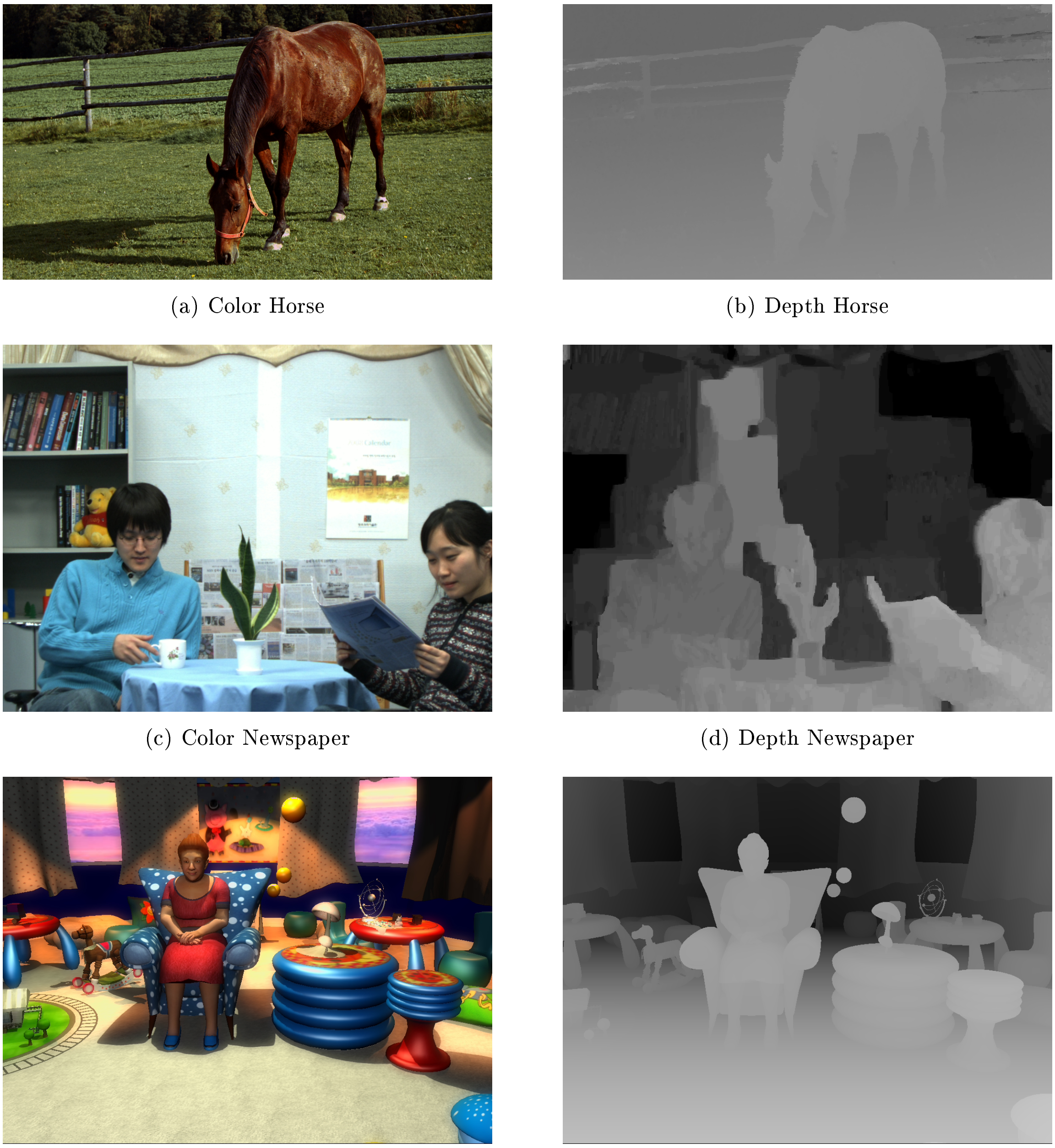
(c) Color Book

(d) Depth Book

(e) Color Breakdancers

(f) Depth Breakdancers

**Figure A.1:** *First frames of color and associated per-pixel depth maps corresponding to first camera of MVD sequences.*



(a) Color Horse

(b) Depth Horse

(c) Color Newspaper

(d) Depth Newspaper

(e) Color Synthetic

(f) Depth Synthtetic

**Figure A.2:** *First frames of color and associated per-pixel depth maps corresponding to first camera of MVD sequences.*

## Percentiles of color and depth components (section 3.4.1):

| | **Percentiles** | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Sequence** | **5** | **10** | **15** | **20** | **25** | **30** | **35** | **40** | **45** | **50** | **55** | **60** | **65** | **70** | **75** | **80** | **85** | **90** | **95** | **100** |
| Ballet | 42 | 76 | 95 | 101 | 105 | 107 | 110 | 112 | 115 | 118 | 121 | 124 | 127 | 130 | 134 | 137 | 141 | 145 | 152 | 253 |
| Book | 22 | 26 | 30 | 34 | 38 | 42 | 46 | 50 | 55 | 61 | 76 | 93 | 112 | 136 | 164 | 185 | 206 | 221 | 239 | 256 |
| Breakdancers | 20 | 26 | 29 | 32 | 35 | 37 | 38 | 40 | 42 | 45 | 51 | 57 | 63 | 68 | 74 | 82 | 92 | 101 | 110 | 256 |
| Horse | 8 | 16 | 24 | 31 | 39 | 48 | 58 | 67 | 74 | 80 | 85 | 89 | 94 | 98 | 102 | 107 | 112 | 119 | 129 | 215 |
| Newspaper | 8 | 17 | 28 | 39 | 53 | 66 | 80 | 92 | 104 | 116 | 129 | 147 | 164 | 177 | 187 | 199 | 211 | 221 | 229 | 243 |
| Synthetic | 11 | 24 | 32 | 40 | 45 | 49 | 57 | 67 | 79 | 94 | 111 | 130 | 151 | 178 | 193 | 201 | 208 | 215 | 223 | 236 |

**Table A.1:** *Summary of percentiles taken in intervals of 5 for color component of all MVD sequences*

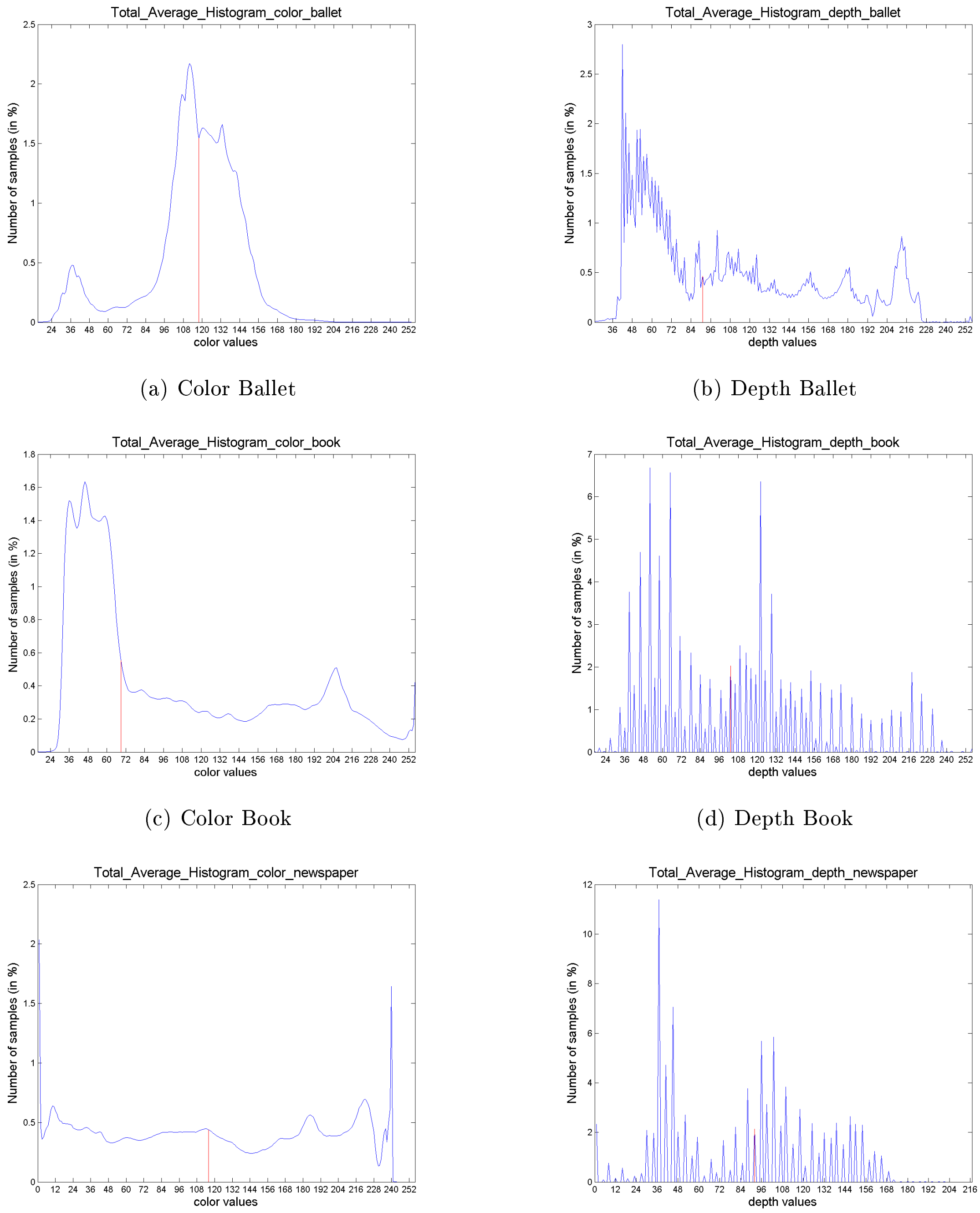
| | **Percentiles** | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Sequence** | **5** | **10** | **15** | **20** | **25** | **30** | **35** | **40** | **45** | **50** | **55** | **60** | **65** | **70** | **75** | **80** | **85** | **90** | **95** | **100** |
| Ballet | 44 | 47 | 51 | 54 | 57 | 61 | 66 | 71 | 79 | 91 | 101 | 110 | 120 | 132 | 149 | 163 | 179 | 198 | 213 | 256 |
| Book | 39 | 46 | 52 | 52 | 58 | 65 | 68 | 78 | 90 | 103 | 113 | 119 | 122 | 129 | 135 | 148 | 160 | 180 | 211 | 256 |
| Breakdancers | 54 | 78 | 86 | 89 | 91 | 95 | 98 | 103 | 108 | 111 | 114 | 119 | 123 | 128 | 135 | 150 | 168 | 183 | 200 | 256 |
| Horse | 105 | 106 | 108 | 112 | 115 | 117 | 120 | 122 | 124 | 126 | 129 | 131 | 132 | 133 | 134 | 135 | 135 | 136 | 137 | 157 |
| Newspaper | 30 | 37 | 37 | 37 | 45 | 45 | 52 | 67 | 81 | 92 | 96 | 99 | 103 | 110 | 118 | 125 | 136 | 147 | 154 | 201 |
| Synthetic | 26 | 34 | 39 | 46 | 57 | 70 | 82 | 113 | 126 | 144 | 153 | 156 | 159 | 162 | 166 | 171 | 176 | 182 | 187 | 237 |

**Table A.2:** *Summary of percentiles taken in intervals of 5 for depth component of all MVD sequences.*

| **Sequence** | $\Pi(5)$ | $\Pi(95)$ | **Range** | $\eta$ |
|---|---|---|---|---|
| Ballet | 44 | 213 | 232 | 0.728 |
| Book | 39 | 211 | 244 | 0.705 |
| Newspaper | 30 | 154 | 201 | 0.617 |

**Table A.3:** *Concentration of used depth values*

## Histogram outputs (section 3.4.1)



(a) Color Ballet

(b) Depth Ballet

(c) Color Book

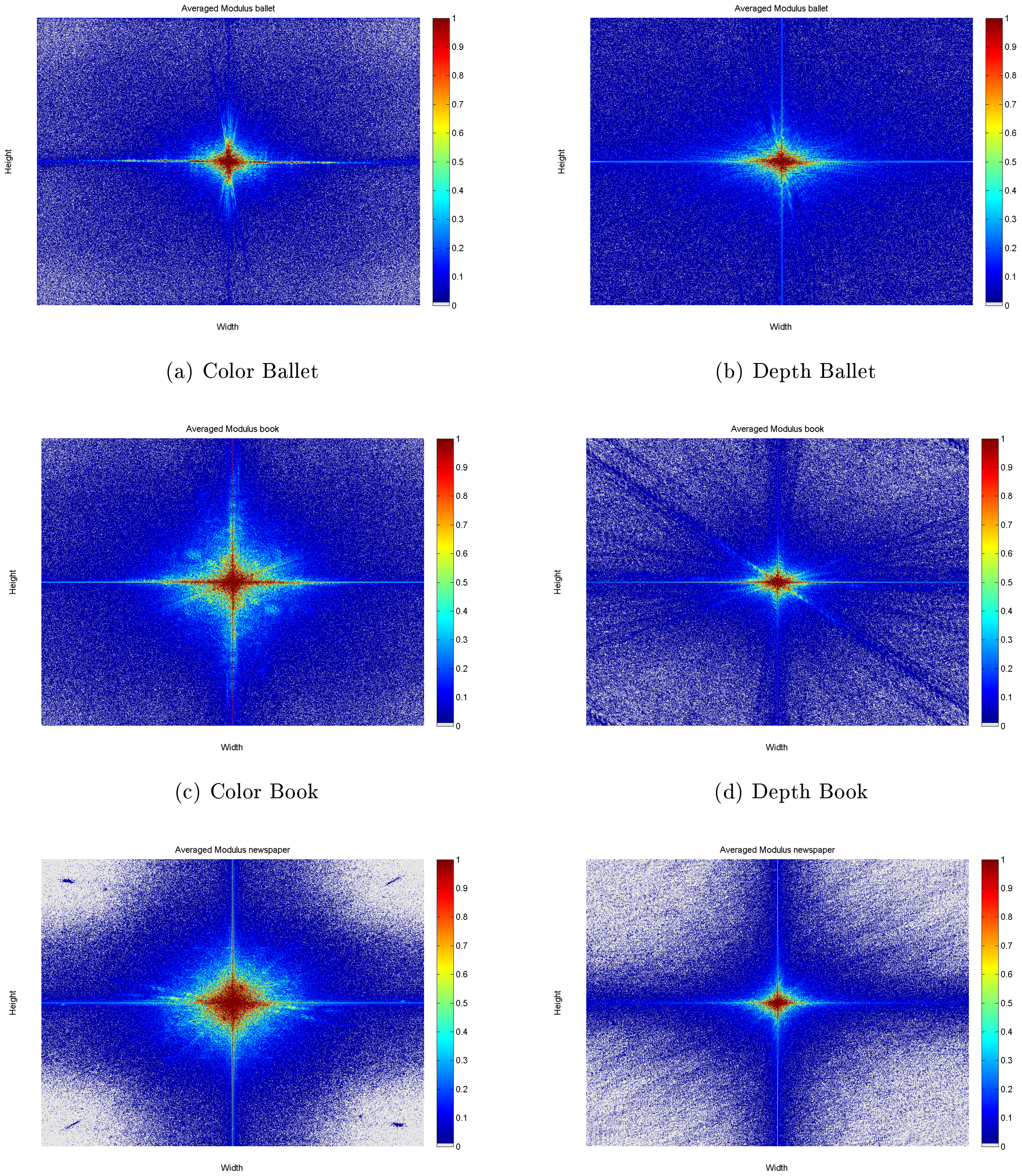
(d) Depth Book

(e) Color Newspaper

(f) Depth Newspaper

**Figure A.3:** *Histograms of the other 3 MVD sequences, corresponding to color and depth.*

## Spectrum outputs (section 3.4.2)



(a) Color Ballet

(b) Depth Ballet

(c) Color Book

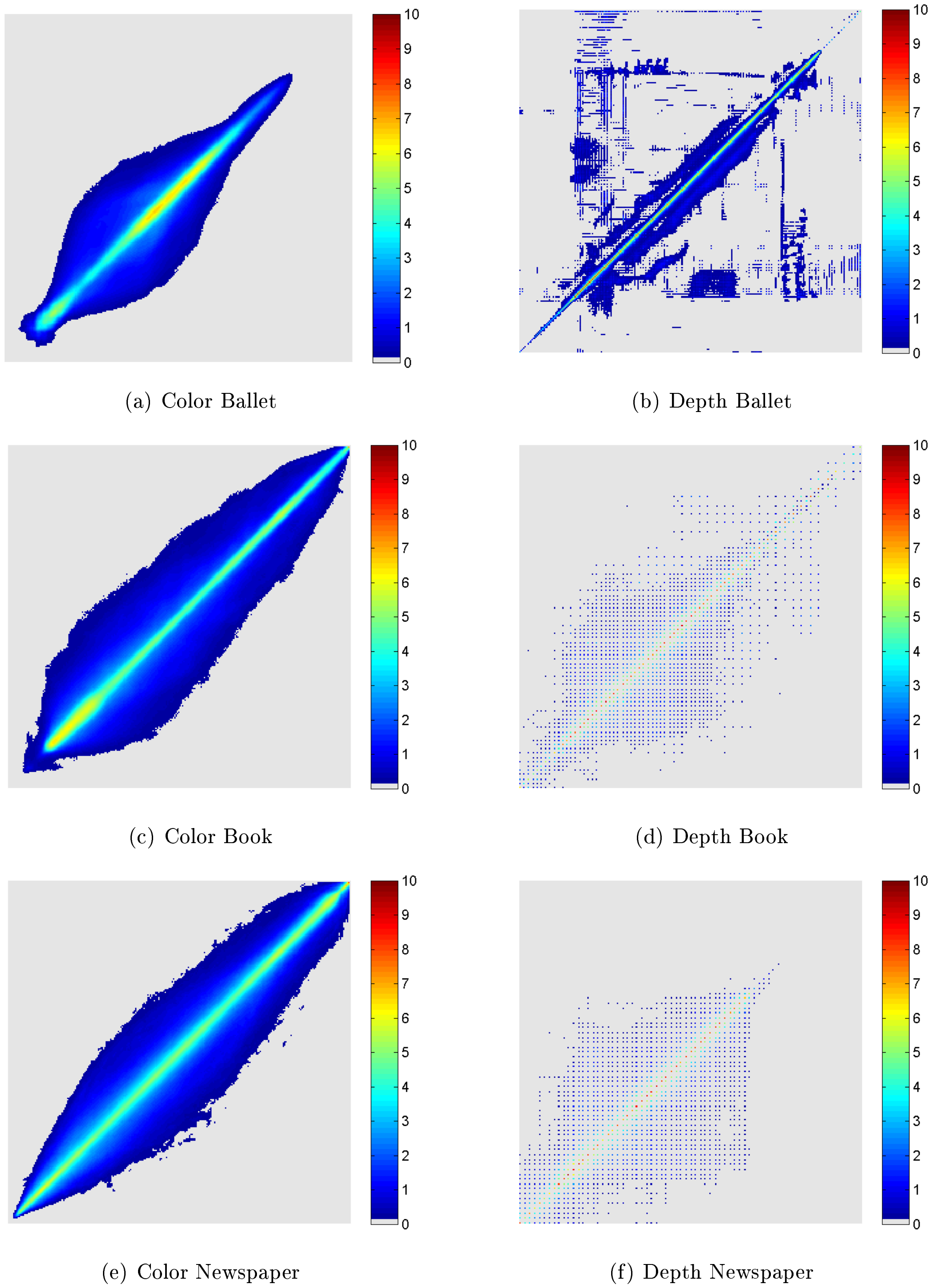
(d) Depth Book

(e) Color Newspaper

(f) Depth Newspaper

**Figure A.4:** *Modulus representation of the other 3 MVD sequences, corresponding to color and depth.*

## Spatial Correlation Histogram Outputs (3.4.3):



(a) Color Ballet

(b) Depth Ballet

(c) Color Book

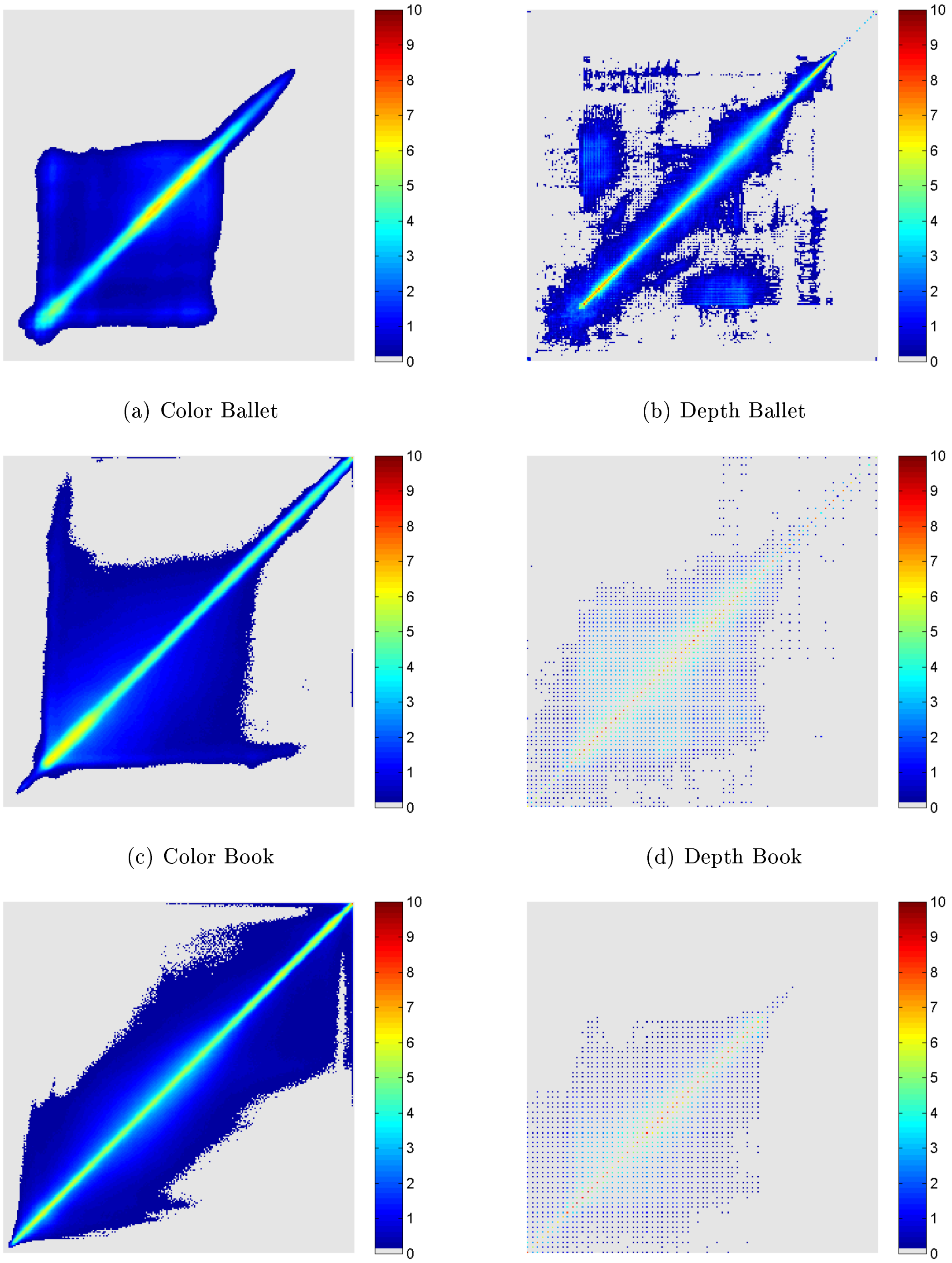
(d) Depth Book

(e) Color Newspaper

(f) Depth Newspaper

**Figure A.5:** *Spatial Correlation Histograms of the other 3 MVD sequences, corresponding to color and depth.*

### Temporal Correlation Histogram Outputs (3.4.3):



(a) Color Ballet

(b) Depth Ballet

(c) Color Book

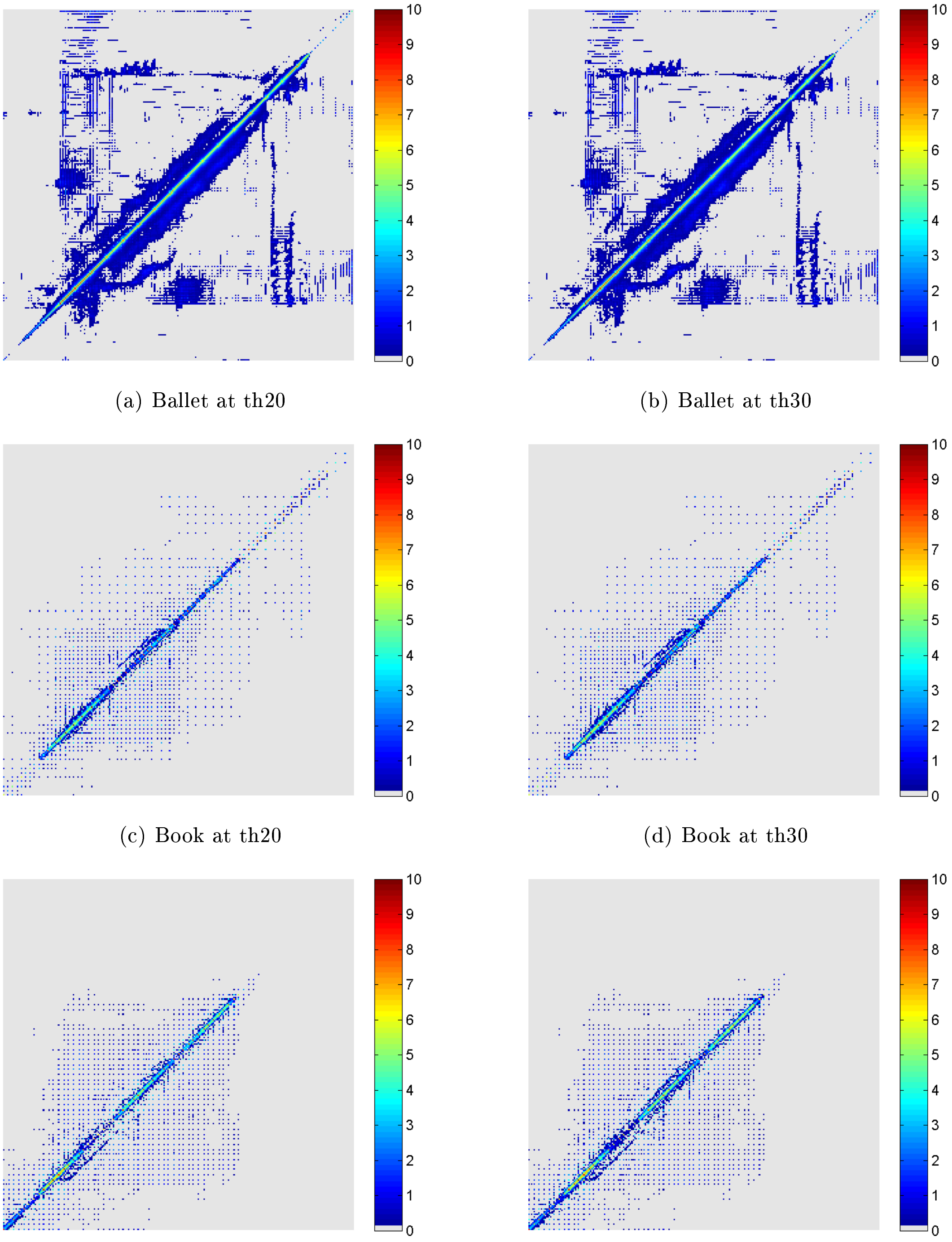
(d) Depth Book

(e) Color Newspaper

(f) Depth Newspaper

**Figure A.6:** *Temporal Correlation Histograms of the other 3 MVD sequences, corresponding to color and depth.*

## Spatial Correlation Histogram Outputs for Option1



(a) Ballet at th20

(b) Ballet at th30

(c) Book at th20

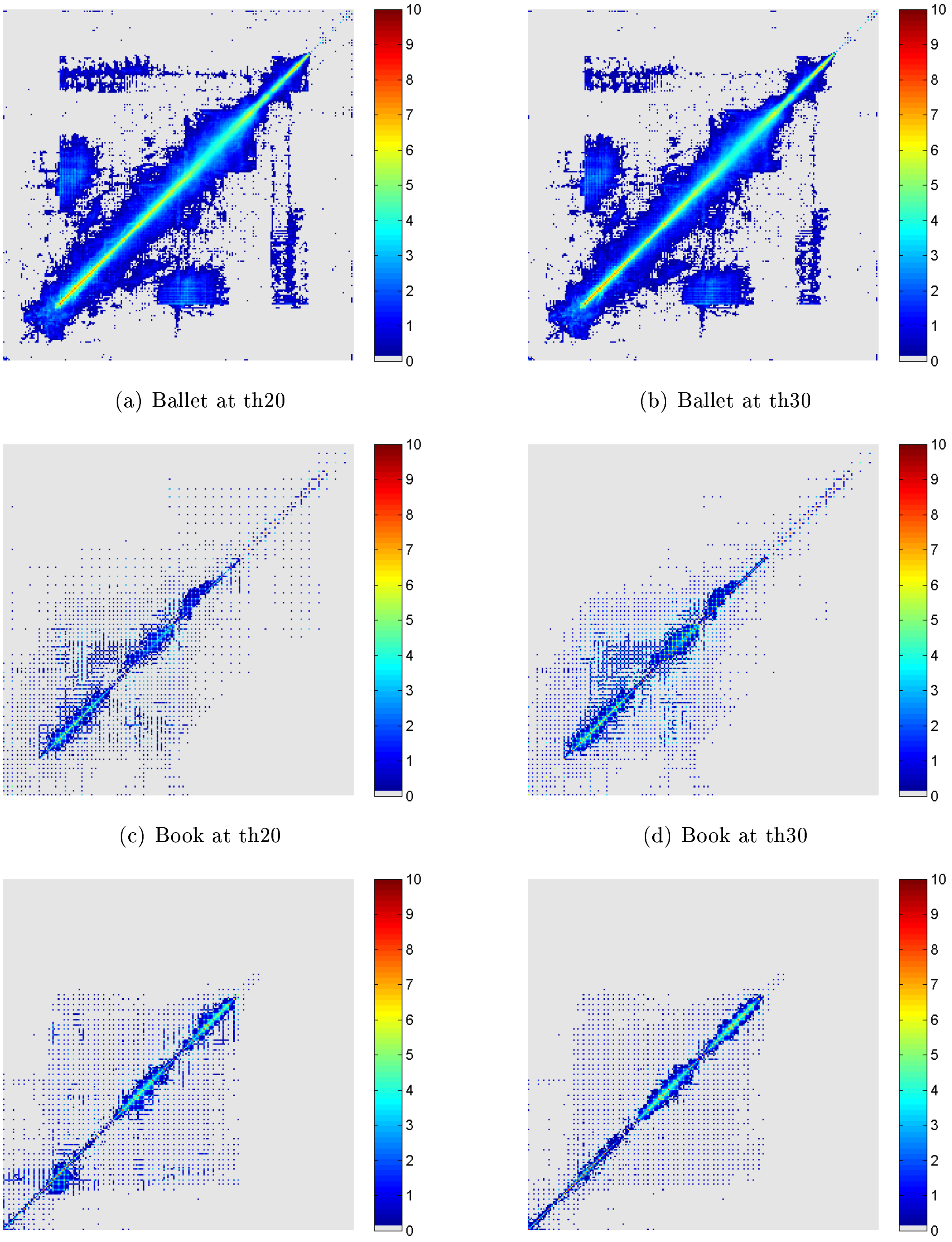
(d) Book at th30

(e) Newspaper at th20

(f) Newspaper at th30

**Figure A.7:** *Spatial Correlation Histograms for Ballet, Book and Newspaper sequences after improvement with Option1 and thresholds at 20 and 30.*

## Temporal Correlation Histogram Outputs for Option1



(a) Ballet at th20

(b) Ballet at th30

(c) Book at th20

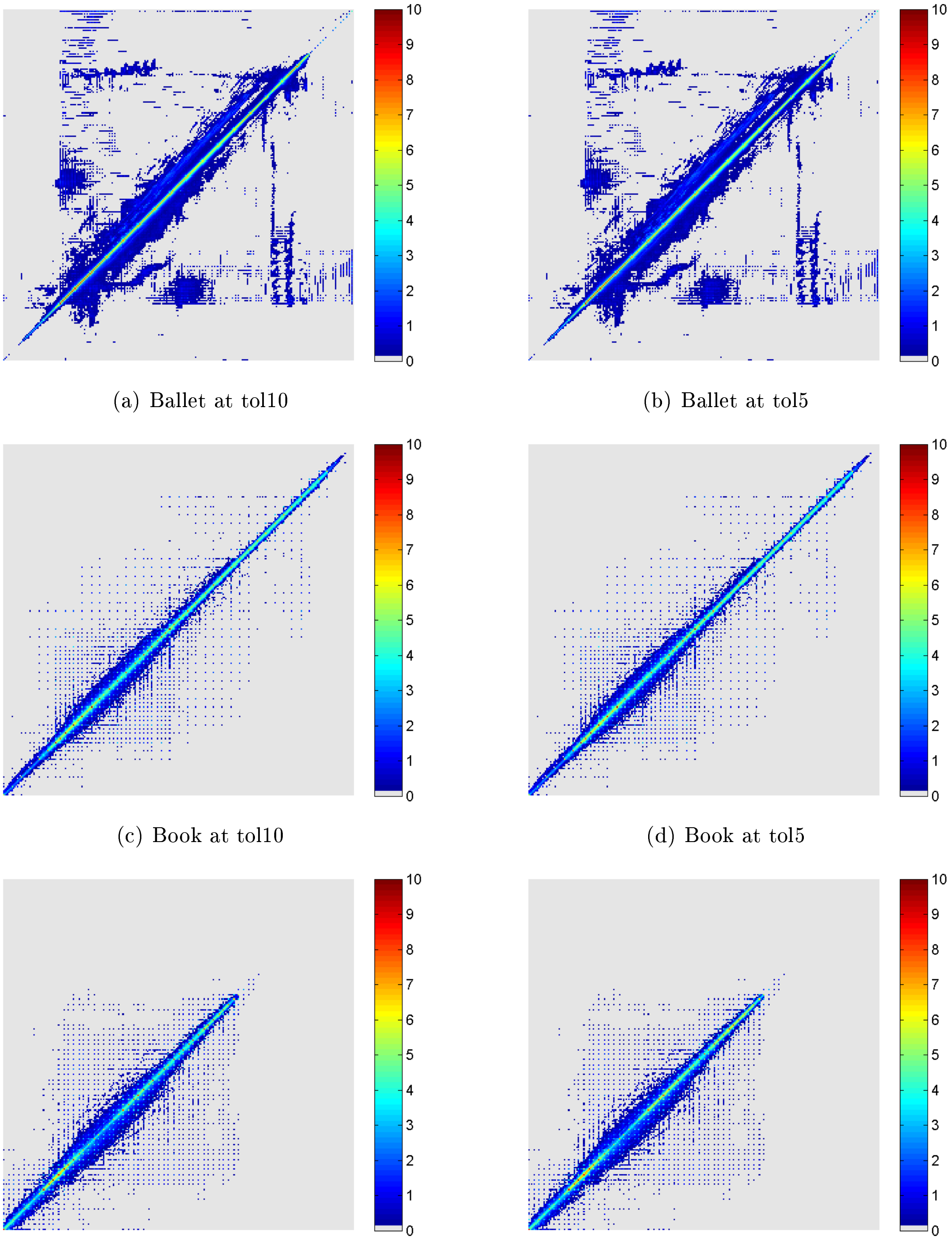
(d) Book at th30

(e) Newspaper at th20

(f) Newspaper at th30

**Figure A.8:** *Temporal Correlation Histograms for Ballet, Book and Newspaper sequences after improvement with Option1 and thresholds at 20 and 30.*

## Spatial Correlation Histogram Outputs for Option2



(a) Ballet at tol10

(b) Ballet at tol5

(c) Book at tol10

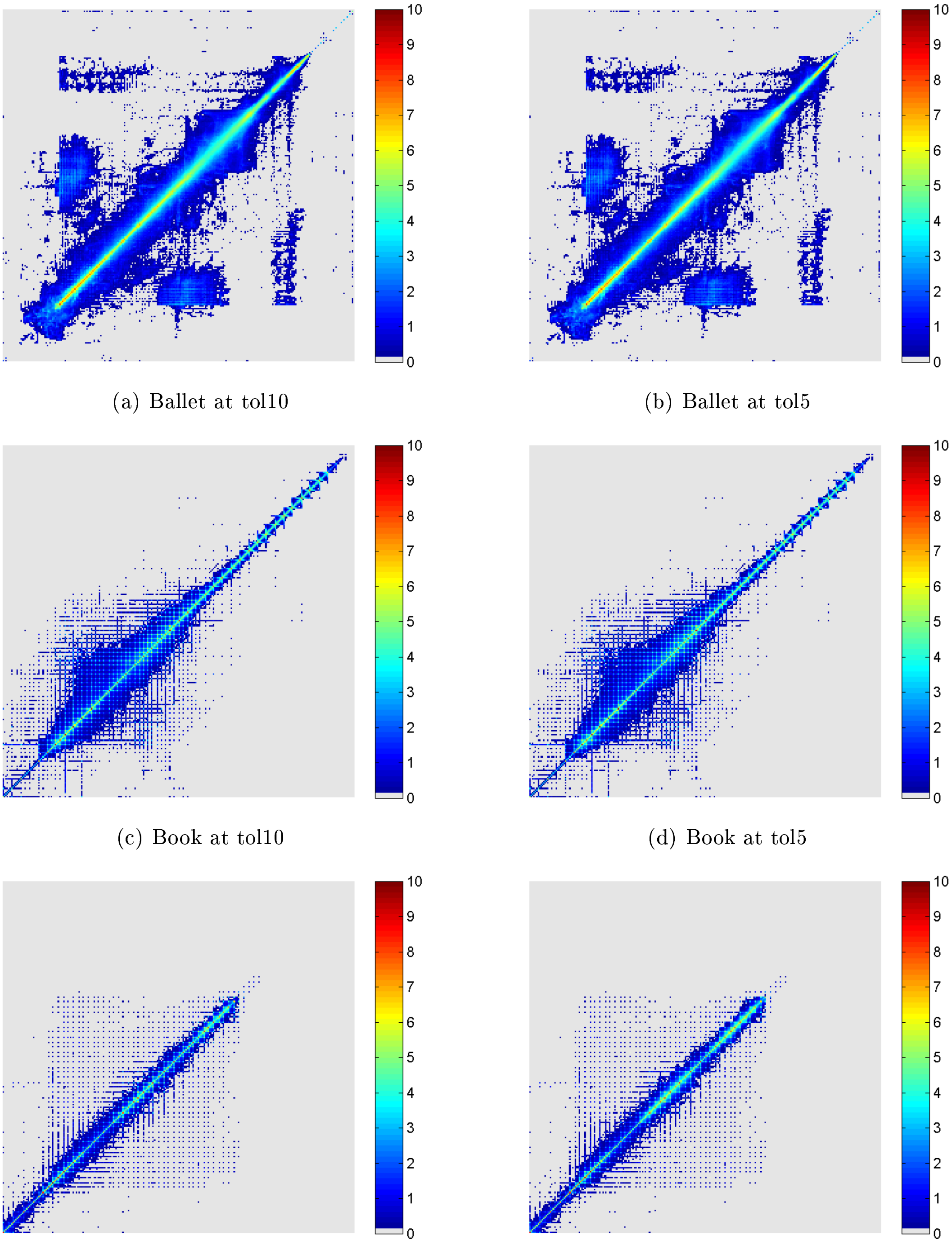
(d) Book at tol5

(e) Newspaper at tol10

(f) Newspaper at tol5

**Figure A.9:** *Spatial Correlation Histograms for Ballet, Book and Newspaper sequences after improvement with Option2 and tolerances 10 and 5.*

## Temporal Correlation Histogram Outputs for Option2



(a) Ballet at tol10

(b) Ballet at tol5

(c) Book at tol10

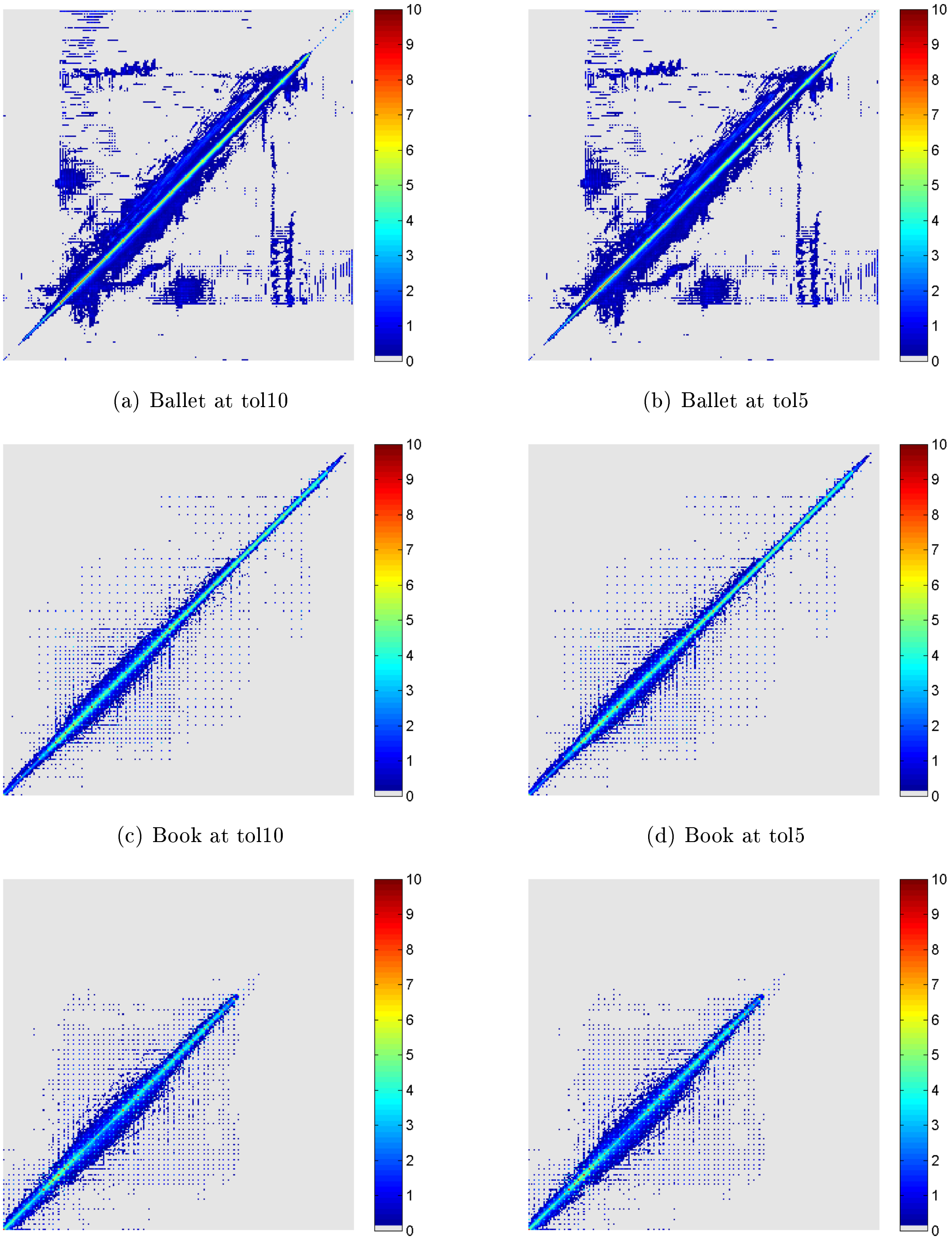
(d) Book at tol5

(e) Newspaper at tol10

(f) Newspaper at tol5

**Figure A.10:** *Temporal Correlation Histograms for Ballet, Book and Newspaper sequences after improvement with Option2 and tolerances 10 and 5.*

## Spatial Correlation Histogram Outputs for Option3



(a) Ballet at tol10

(b) Ballet at tol5

(c) Book at tol10

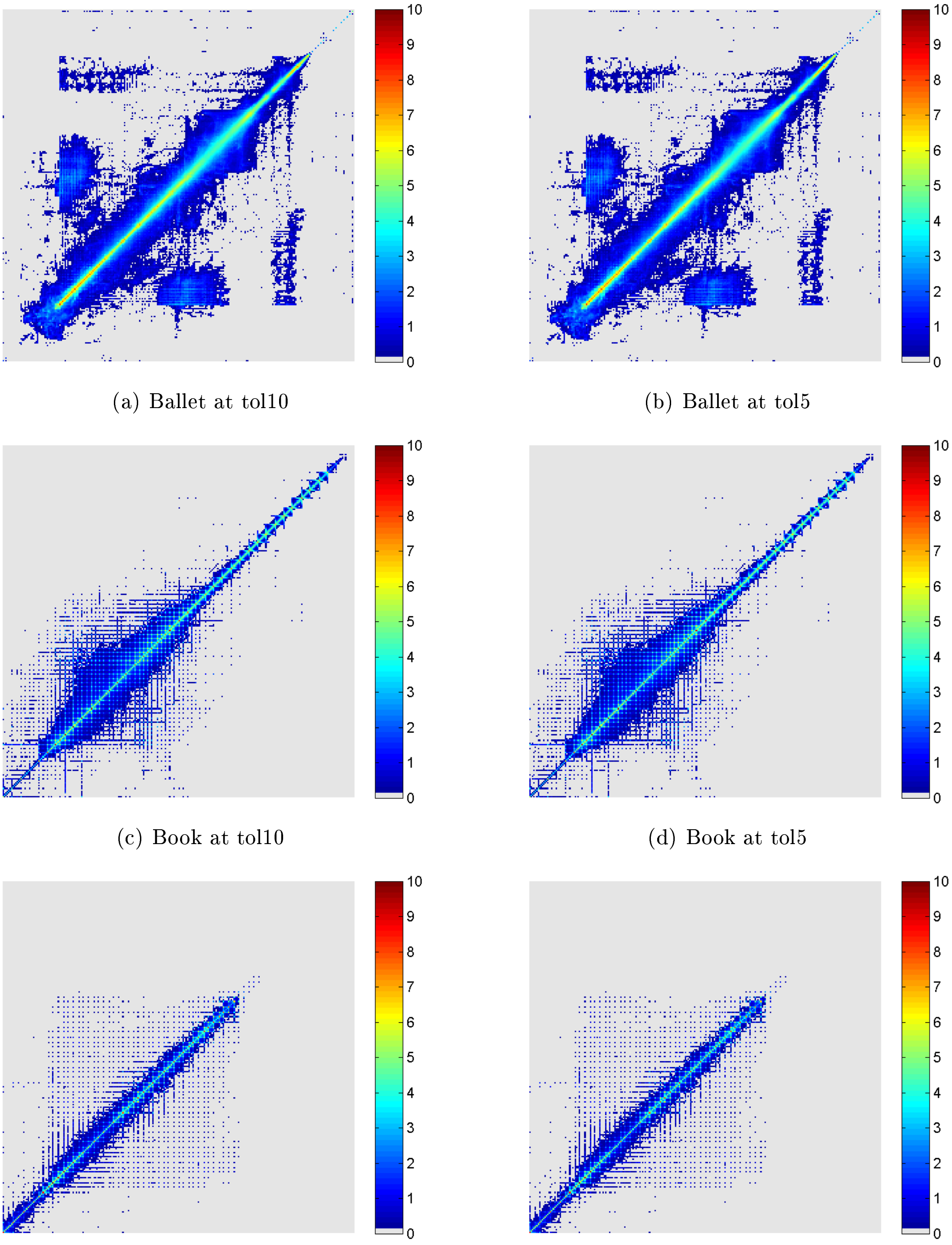
(d) Book at tol5

(e) Newspaper at tol10

(f) Newspaper at tol5

**Figure A.11:** *Spatial Correlation Histograms for Ballet, Book and Newspaper sequences after improvement with Option3 and tolerances 10 and 5.*

## Temporal Correlation Histogram Outputs for Option3



(a) Ballet at tol10

(b) Ballet at tol5

(c) Book at tol10

(d) Book at tol5

(e) Newspaper at tol10

(f) Newspaper at tol5

**Figure A.12:** *Temporal Correlation Histograms for Ballet, Book and Newspaper sequences after improvement with Option3 and tolerances 10 and 5.*

# Bibliography

[1] Aljoscha Smolic, Karsten Müller, Philipp Merkle, Christoph Fehn, Peter Kauff, Peter Eisert, and Thomas Wiegand. 3d video and free viewpoint video - technologies, applications and mpeg standards. *IEEE International Conference on Multimedia*, pages 2161–2164, July 2006.

[2] Janusz Konrad and Michael Halle. 3-d displays and signal processing: An answer to 3-d ills? *IEEE Signal Processing Magazine*, Vol. 24, November 2007.

[3] Oliver Schreer, Peter Kauff, and Thomas Sikora, editors. *3D Videocommunication: Algorithms, concepts and real-time systems in human centred communication*, chapter Human Factors of 3D Displays, pages 219–234. Wiley, 2005. Authors of this chapter: W.A.Ijsselsteijn, P.J.H. Seuntiëns and L.M.J. Meesters.

[4] P. Kauff, N. Atzpadin, C. Fehn, M. Müller, O. Schreer, A. Smolic, and R. Tanger. Depth map creation and image-based rendering for advanced 3dtv services providing interoperability and scalability. *Image Communication*, Vol. 22, Issue 2:217–234, February 2007.

[5] A. Smolic, K. Müller, P. Merkle, T. Rein, M. Kautzner, P. Eisert, and T. Wiegand. Free viewpoint video extraction, representation, coding, and rendering. *IEEE International Conference on Image Processing, ICIP 2004*, pages 3287–3290, October 2004.

[6] Christoph Fehn. Depth-image-based rendering (dibr), compression and transmission for a new approach on 3d-tv. `http://iphome.hhi.de/fehn/Publications/fehn_EI2004.pdf`, 2006.

[7] N. Atzpadin, P. Kauff, and O. Schreer. Stereo analysis by hybrid recursive matching for real-time immersive video conferencing. *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 14, Issue 3:321–334, March 2004.

[8] Philipp Merkle, Aljoscha Smolic, Karsten Müller, and Thomas Wiegand. Multi-view video plus depth representation and coding. *IEEE International Conference on Image Processing, ICIP 2007*, Vol. 1:201–204, September 2007.

[9] T. Bothe, A. Gesierich, W. Li, C. v. Kopylow, N. Köpp, and W. Jüptner. 3d-camera for scene capturing and augmented reality applications. *3DTV Conference, 2007*, May 2007.

[10] Roger Y. Tsai. A versatile camera calibration techniaue for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, Vol. 3, Issue 4:323–344, August 1987.

[11] Shinichi Yamaguchi, Masanobu Kimura, Jyunichi Hosowaka, and Yasuo Takemura. Stereoscopic video movie camera "3d-cam". *IEEE 1988 International Conference on Consumer Electronics,*, June 1988.

[12] ITU-T, recommendation H.262. *Information technology - Generic coding of moving pictures and associated audio information: Video*, 2000.

[13] ITU-T, recommendation H.264. *Advanced video coding for generic audiovisual services*, 2009.

[14] Thomas Wiegand, Gary J. Sullivan, Gisle Bjøntegaard, and Ajay Luthra. Overview of the h.264/avc video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 13, Issue 7:560–576, July 2003.

[15] Aljoscha Smolic and Peter Kauff. Interactive 3-d video representation and coding technologies. In *Proceedings of the IEEE*, volume Vol. 93, Issue 1, pages 98–110, January 2005.

[16] William Sanders and David F. McAllister. Producing anaglyphs from synthetic images. In *Proc. SPIE Stereoscopic Displays and Virtual Reality Systems*, pages 348–358, January 2003.

[17] Janusz Konrad, Bertrand Lacotte, and Eric Dubois. Cancellation of image crosstalk in time-sequential displays of stereoscopic video. *IEEE Transactions on Signal Processing*, Vol. 9 No. 5:897–908, May 2000.

[18] Željko Hocenski, Suzana Vasilić, and Verica Hocenski. Improved canny edge detection in ceramic tiles defect detection. *IEEE 32nd Annual Conference on Industrial Electronics, IECON 2006*, pages 3328–3331, 2006.

[19] Alan V. Oppenheim and Jae S. Lim. The importance of phase in signals. *Proceedings of the IEEE*, Vol. 69, No. 5:529–541, May 1981.

[20] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*, chapter Introduction, pages 1–33. Prentice Hall, 2002. ISBN:0201180758.

[21] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*, chapter Edge Detection, pages 572–585. Prentice Hall, 2002. ISBN:0201180758.

[22] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*, chapter Image Enhancement in the Spatial Domain, pages 128–133. Prentice Hall, 2002. ISBN:0201180758.

[23] John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8 No. 6, November 1986.

[24] Vishvjit S. Nalwa and Thomas O. Binford. On detecting edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8 No. 6, March 1986.

[25] Fredrik Bergholm. Edge focusing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-9 No. 6, November 1987.

[26] C.A. Rothwell, J.L. Mundy, W. Hoffman, and V.-D. Nguyen. Driving vision by topology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-9 No. 6, November 1987.

[27] Lee A. Iverson and Steven W. Zucker. Logical/linear operators for image curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 17 No. 10, October 1995.

[28] Mike Heath, Sudeep Sarkar, Thomas Sanocki, and Kevin Bowyer. Comparison of edge detectors: A methodology and initial study. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'96)*, December 1996.

[29] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Proceedings of the 1998 IEEE International Conference on Computer Vision*, pages 839–846, January 1998.

[30] Daniel Malacara. *Color Vision and Colorimetry: Theory and Applications*. SPIE–The International Society for Optical Engineering, 2002. ISBN:0819442283.

[31] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library*, chapter Image Transforms, Canny, pages 151–153. O'Reilly Media, 2008. ISBN:0596516134.