DISSERTAÇÃO

Mestrado em Engenharia Eletrotécnica - Eletrónica e Telecomunicações

Structure Tensor-based Depth Estimation from Light Field Images

Rui Miguel Leonel Lourenço

Leiria, Janeiro de 2019

MASTER DISSERTATION

Electrical Engineering - Eletronics and Telecommunications

Structure Tensor-based Depth Estimation from Light Field Images

Rui Miguel Leonel Lourenço

Master dissertation performed under the supervision of Professors Pedro António Amado Assunção and Luís Miguel de Oliveira Pegado de Noronha e Távora both from Escola Superior de Tecnologia e Gestão of Instituto Politécnico de Leiria.

Leiria, January 2019

The important thing is not
to stop questioning;
curiosity has its own
reason for existing.

(Albert Einstein)

Acknowledgments

I would like to thank everyone that helped during the research and development of this work and made it possible to accomplish.

I express my gratitude towards my advisers Prof. Pedro António Amado Assunção and Prof. Luís Miguel de Oliveira Pegado de Noronha e Távora that helped me through this research journey and guided me towards these results. I would also like to thank the members of the Multimedia Signal Processing – Leiria research group, Prof. Sérgio Manuel Maciel de Faria, Prof. Rui Fonseca-Pinto, José Nunes dos Santos Filipe, João Miguel Pereira da Silva Santos e Pedro Miguel Marques Pereira for their help and insight provided through debates and discussions throughout the development of this work. I would additionally like to thank Ricardo Jorge Santos Monteiro and João Filipe Monteiro Carreira, for providing an excellent work environment and the necessary help whenever any doubt or problem arose.

I would like to thank the opportunity of working as a researcher in the project "Dermo-Pleno – Dermo-Plenoptic imaging for skin lesion assessment" (DermoPleno/IT/2016) and Project PlenoISLA – PTDC/EEI-TEL/28325/2017, and the financial support provided by Instituto de Telecomunicações, through Fundação para a Ciência e Tecnologia. I would also like to acknowledge the lab facilities and research environment provided by the Instituto de Telecomunicações and Escola Superior de Tecnologia e Gestão of IPLeiria. I would also like to express my thanks to Daniel Rivero Castillo for the software implementation of the krawtchouk polynomials.

Finally, I would like to extend my deepest thanks to my parents, Gorete Leonel and Rui Carlos da Conceição Lourenço, and my sister Catarina Sofia Leonel Lourenço for their extended support and dedication. This work would not be possible without their presence.

Abstract

This thesis presents a novel framework for depth estimation from light field images based on the use of the structure tensor.

A study of prior knowledge introduces general concepts of depth estimation from light field images. This is followed by a study of the state-of-the art, including a discussion of several distinct depth estimation methods and an explanation of the structure tensor and how it has been used to acquire depth estimation from a light field image.

The framework developed improves on two limitations of traditional structure tensor derived depth maps. In traditional approaches, foreground objects present enlarged boundaries in the estimated disparity map. This is known as silhouette enlargement. The proposed method for silhouette enhancement uses edge detection algorithms on both the epipolar plane images and their corresponding structure tensor-based disparity estimation and analyses the difference in the position of these different edges to establish a map of the erroneous regions. These regions can be inpainted with values from the correct region. Additionally, a method was developed to enhance edge information by linking edge segments.

Structure tensor-based methods produce results with some noise. This increases the difficulty of using the resulting depth maps to estimate the orientation of scenic surfaces, since the difference between the disparity of adjacent pixels often does not correlate with the real orientation of the scenic structure. To address this limitation, a seed growing approach was adopted, detecting and fitting image planes in a least squares sense, and using the estimated planes to calculate the depth for the corresponding planar region.

The full framework provides significant improvements on previous structure tensor-based methods. When compared with other state-of-the-art methods, it proves competitive in both mean square error and mean angle error, with no single method proving superior in every metric.

Keywords: light field, structure tensor, depth map

Resumo

Esta dissertação expõe um método inovador baseado no tensor de estrutura para estimação de profundidade utilizando imagens *light field*.

Um estudo de conhecimento prévio introduz conceitos gerais de estimação de profundidade através de imagens *light field*, isto é seguido por um estudo do estado-da-arte, incluindo uma discussão sobre vários métodos de estimação de profundidade e uma explicação do tensor de estrutura e como este pode ser usado para adquirir uma estimativa de profundidade através de imagens *light field*.

A estrutura desenvolvida melhora duas limitações de mapas de profundidade adquiridos por métodos tradicionais baseados no tensor de estrutura. Usando métodos tradicionais, objetos em primeiro plano exibem fronteiras alargadas no mapa de profundidade estimado. Isto é conhecido como alargamento de silhuetas. O método proposto para melhoria de silhuetas usa deteção de arestas tanto nas imagens de plano epipolar como no seu mapa de disparidade, obtido através do tensor de estrutura, e analisa diferenças entre a posição destas arestas para estabelecer um mapa das zonas erróneas. Estas zonas podem ser *inpainted* com valores da zona correta. Adicionalmente, foi desenvolvido um método para melhorar a informação de arestas ligando segmentos de aresta descontínuos.

Métodos baseados no tensor de estrutura produzem resultados com algum ruído. Isto aumenta a dificuldade de utilizar os mapas de profundidade resultantes para estimar a orientação de superfícies cénicas, uma vez que a diferença entre a disparidade de dois pixéis adjacentes frequentemente não se correlaciona com a orientação verdadeira da estrutura cénica que eles representam. Para melhorar esta limitação, uma abordagem baseada em crescimento de sementes foi usada, detetando e ajustando planos numa abordagem de mínimos quadrados, e utilizando os planos estimados para calcular a profundidade da zona planar correspondente.

A estrutura completa apresenta uma melhoria significativa em relação a métodos prévios baseados no tensor de estrutura. Comparada com outros métodos do estado-da-arte, esta estrutura demonstra ser competitiva tanto em erro quadrático médio como em erro angular médio, sendo que nenhum método é superior em todas as métricas.

Palavras chave: *light field*, tensor de estrutura, mapa de profundidade

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Context and motivation

Depth information is necessary to obtain an accurate 3D representation of a given visual scene. The depth information itself can be quite relevant in the diagnostic of several medical conditions, such as melanoma and other skin lesions [7]. In automatic quality control in industrial processes [8], it may also facilitate computer vision problems such as feature detection and the segmentation of a scene [9] in its various objects.

Traditional cameras only capture the average intensity of the light rays striking the image sensor at a given point, thus losing information about the structure of the rays. Plenoptic or light field photography provides a mechanism to capture both texture and depth information from a single acquisition. Edward H. Adelson and James R. Bergen, in [1], describe the plenoptic function that can fully characterize the information that is contained in every point in space.

Ren Ng [10] further cements Light Field photography as a valuable method for acquiring depth maps by describing an hand-held plenoptic camera. The practicality of Light Field imaging becomes an advantage relative to other methods that require separate acquisitions for color and depth.

The increasing availability of hand-held Light Field cameras has been driving new research efforts in depth estimation from plenoptic images that became an active field of study. These research efforts were aided by previous research. For instance, similarities between plenoptic images and images acquired by a regular camera moving in a single axis lead to the framework of Epipolar Plane Images(EPIs), first described in [11] and further explained in Section 2.1.3. It could be of use in estimating disparity in Light Field images. Methods using EPIs are still used extensively in state of the art methods for depth acquisition.

The application fields where depth information is required are in rapid development and no algorithm has emerged yet as the best among all other alternatives. Thus there is room for further research and improvements on the state of the art to create better performing algorithms.

1.2 Objectives

This thesis aims to investigate and develop a complete depth estimation framework capable of achieving results on par with the state-of-the-art. The work done is mainly focused on the structure tensor [12], described in detail in Section 2.2.3, to estimate disparity, improving previous methods based on a similar approach, such as [13] and [14].

This goal is attained by first studying the limitations of previous methods, and then defining a technical approach to minimize the effect of those problems, creating a new state-of-the-art disparity estimation algorithm. Two main problems are focused upon, the accuracy of silhouette representation and the lack of local smoothness leading to highly inaccurate estimation of the orientation of surfaces in the light field image. Additionally, changes to a Matting-based optimization approach first proposed by [13] are discussed.

1.3 Outline of the Thesis

This introductory chapter is followed by a brief introduction to Light Field Imaging in Chapter 2, starting with its origins with Adelson and Bergen's Plenoptic Function [1], and mostly focusing on the required knowledge to explain disparity acquisition from Light Field images. A description of different state-of-the-art methods follows, providing an overview of the context of this thesis, as well as a detailed explanation of the concepts and methods this work is built upon.

In Chapter 3, two limitations found in the conventional structure tensor algorithm are described: the enlargement of silhouettes and the lack of local smoothness. The proposed solutions to these problems are then presented in detail.

In Chapter 4, test conditions and datasets are presented and the results of our improvements to structure tensor based methods are analyzed in relation to the non-improved results. Additionally, the complete framework is compared with other state-of-the-art methods.

Finally, in Chapter 5, we provide a critical overview of the work done and possible directions for future work.

Chapter 2

State of the Art

In this chapter a brief review of concepts relevant to Light Field disparity estimation is presented in order to provide sufficient background information as well as some context in regard to state-of-the-art disparity estimation methods. A brief introduction to the plenoptic function is followed by an explanation of the parametrization most commonly utilized in practical applications and how one can leverage these representations to extract disparity from a single Light Field image acquisition. Lastly, some relevant disparity estimation methods from the literature are described.

2.1 The Plenoptic Function and Light Field Disparity Estimation

If, using a pinhole camera, the light intensity of a scene is captured, a function $\mathcal{P}(x, y)$ is obtained, depending on the coordinate system. Additionally, if all different wavelengths are discriminated, the function $\mathcal{P}(x, y, \lambda)$ is obtained. By considering this function for all possible points of a scene in three dimensional space, V_x, V_y and V_z, over time, the full plenoptic function is obtained:

$$\mathcal{P}(x, y, \lambda, t, V_x, V_y, V_z) \tag{2.1}$$

Considering additional properties of light, like polarization, one could create an even more complete function. This thesis is focused on a simplified version of the function described in Equation 2.1. Figure 2.1, provides a visual interpretation of the Plenoptic function. Each eye can be seen as a camera, capturing light rays from different directions. Unlike human observers, the plenoptic function also considers the light rays coming from behind the observer.

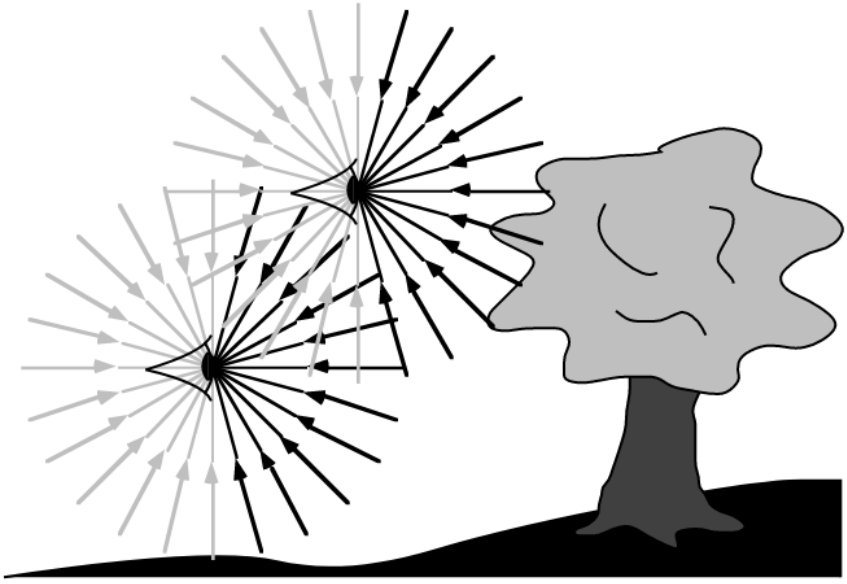


Figure 2.1: A visualization of the plenoptic function. (Adelson and Bergen [1])

While the plenoptic function may seem overly abstract, all imaging devices sample subsets of this function. This thesis will focus on the so-called 4D or Lumigraph parametrization of the plenoptic function [15].

2.1.1 The 4D-Light Field Parametrization

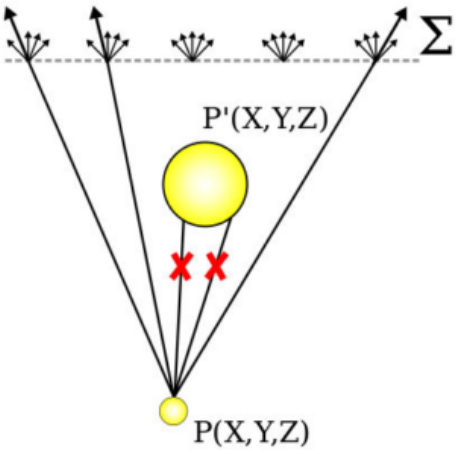


Figure 2.2: Opaque surface only assumption. (*Wanner* [2])

The 4D parametrization of the plenoptic function assumes a gray scale camera, neglecting the wavelength parameter λ. Additionally, if a stationary photography is considered the time parameter t can be excluded. To further reduce dimensionality, one can assume that the intensity of a light ray does not depend on the actual position along the ray. This is

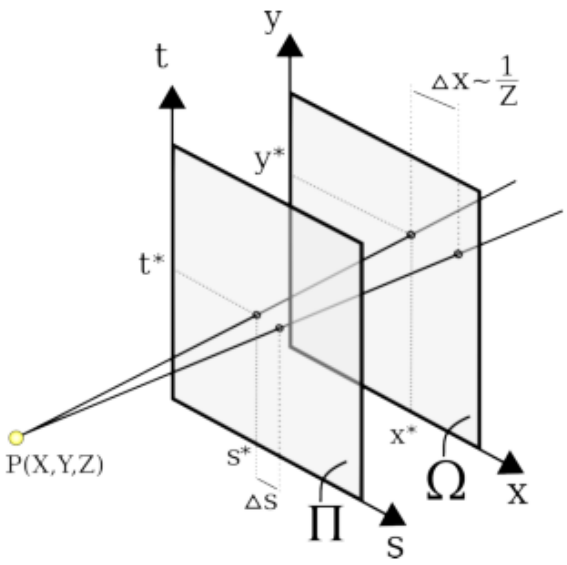


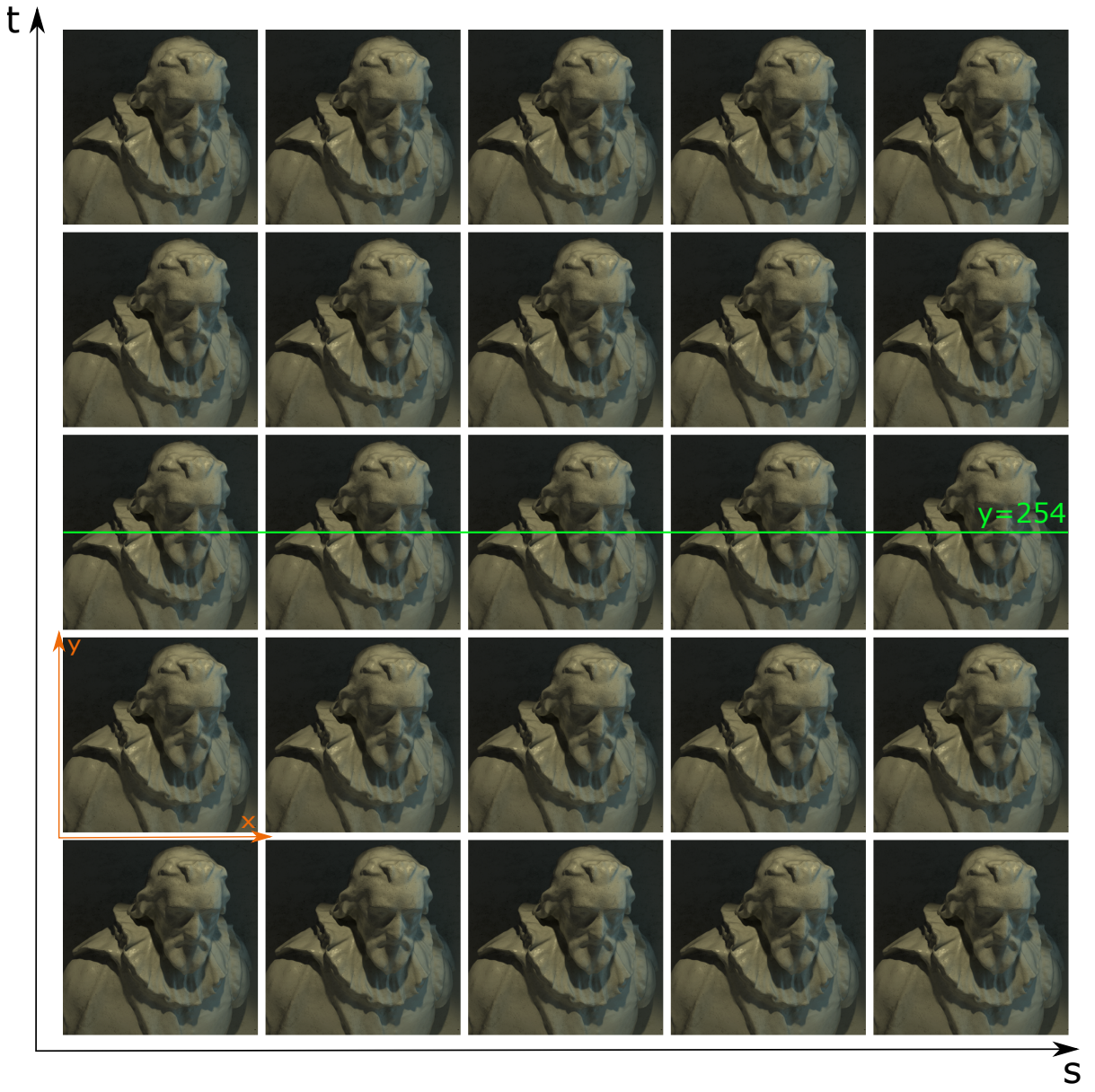
Figure 2.3: Two plane parametrization of Light Field images. (*Wanner* [2])

equivalent to sampling the plenoptic function in a surface Σ. As can be seen in Figure 2.2, this model assumes that light intensity does not change between opaque objects and the imaging device, such that the rays originating from object P that are occluded by object P' are considered irrelevant. This also assumes the non-existence of semi-transparent objects that alter the light ray in-between an opaque object and the surface Σ.

Thus as shown in Figure 2.3, the plenoptic function can be parametrized by the intersection point with two planes. Each pinhole camera located in the plane Π captures a different view of the scene on the image plane Ω. We obtain a four variable function $\mathcal{P}(x, y, s, t)$.

This parametrization of the plenoptic function can be seen as a matrix of all different views of a scene with s and t representing the indices of the view matrix and x and y representing the image coordinates of each view.

In practical cases, it is common to represent a light field as a view matrix of colored images. The consideration of color coincides with a five dimensional plenoptic function, but it is more often conceptualized as three discrete 4D plenoptic functions, each representing light intensity in different parts of the frequency spectrum. A visual representation of this can be seen in Figure 2.4(a). Each image represents a view of the same scene from a different view point, so that the indices of the view matrix represent the angular coordinates (u, v) of the plenoptic function, and the coordinates (x, y) within each image, represent the spatial coordinates of the plenoptic function.



(a)

(b)

Figure 2.4: (a) View Matrix representation of a Light Field image. The green line represents the cut used to obtain the EPI below. (b) EPI for $t = 3$, $y = 254$ of the view matrix representation.

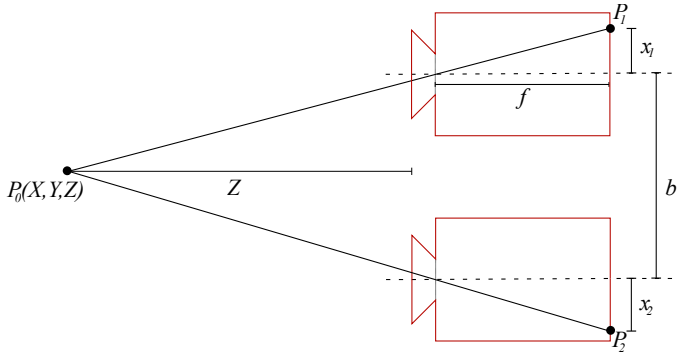


Figure 2.5: A diagram of a stereo setup.

2.1.2 Disparity Estimation From Light Field Images

Estimating depth from sets of photographs captured from different perspectives of the same scene is an age old problem in image processing. A very well developed approach is stereo imaging which provides a very simple and straightforward method for detecting disparity.

As can be seen in Figure 2.5, a stereo setup consists of two cameras separated by a distance b. A point $P_0(X, Y, Z)$ is projected onto different positions in the sensors of the two cameras with the same focal length, f. This difference $x_1 - x_2$ is known as the disparity d. This disparity is inversely proportional to the distance Z of the object to the camera, as per the following equation:

$$d = x_1 - x_2 = f\Big(\frac{X + \frac{b}{2}}{Z} - \frac{X - \frac{b}{2}}{Z}\Big) = b\frac{f}{Z} \tag{2.2}$$

Using the equivalence above, stereo-based algorithms for depth estimation can be summarized into two essential steps. The first step, designated the matching problem, involves finding the corresponding pixels in the different images. The second step involves finding the depth of the pixel, knowing the disparity between the two views.

Light-Field cameras, due to the ordered and regular displacement of the obtained views, provides easy solutions to the matching problem. Different methods of disparity estimation for Light Field images in the literature take advantage of this regularity in different ways.

Although disparity and depth are closely related, the transformation from disparity to depth requires an additional step and the use of camera parameters. Therefore, results of depth estimation algorithms are often presented and compared in terms of disparity. The same is true for the results presented in this work.

2.1.3 The Epipolar Plane Image Approximation to the Matching Problem

Bolles *et al.* [11], in their studies with a camera moving in a straight rail, taking pictures periodically, found that by sequentially stacking corresponding horizontal lines of each view, the resulting image, dubbed an Epipolar Plane Image (EPI), formed quadrilateral surfaces, with its slant being proportional to the depth of the points included in those surfaces. This phenomena was possible due to the small displacement of the perspective between adjacent views. Corresponding pixels in adjacent views shifted very slightly so that they formed slanted lines when stacked on top of each other. This slant is proportional to the disparity of the pixel.

This reduces the problem of estimating disparity from matching interest points and calculating their disparity, to a that of estimating the orientation of linearly symmetric structures in an image, a potentially simpler and well studied image processing problem. The concept of linear symmetry in images is defined in section 2.1.4.

Since light field images have very low displacement between adjacent views, they can make great use of this mechanism. A line or column of the view matrix is equivalent to the temporal sequence of images obtained by Bolles *et al.* [11] with a camera moving in a straight rail. Mathematically, EPIs are equivalent to a cut of the 4D plenoptic function, where corresponding angular and position variables are fixed to a number. Figure 2.4(b) showcases an horizontal EPI obtained by fixing $t = 3$ and $y = 254$.

2.1.4 Linear Symmetry in Images

As mentioned above, the Epipolar Plane approximation reduces depth estimation to a problem of finding the direction of linear symmetries in images. It is thus important to define linearly symmetric images, in order to better describe the way depth estimation algorithms take advantage of this property of EPIs. It is important to reinforce that, mathematically, any sub-region of an image can still be treated on its own as a smaller image, and thus the concepts presented in this chapter can be useful for small, linear-symmetric patches of a bigger image that is not linearly symmetric.

Let us consider a continuous image $f(\boldsymbol{r})$, where $\boldsymbol{r} = [x, y]^T$ is a two dimensional real vector representing the coordinates of a point in the image plane and the vector $\boldsymbol{k} = [k_x, k_y]^T$, a unit vector that represents a constant direction in the image plane.

The function f is called a linearly symmetric image if there exists a scalar function of

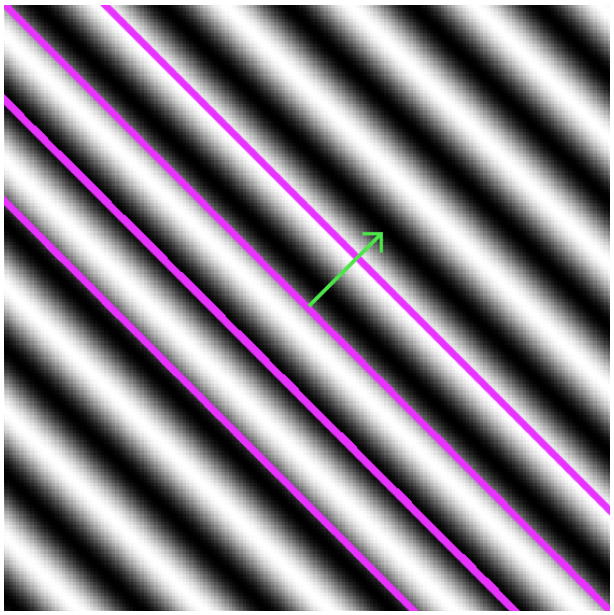


Figure 2.6: Linearly symmetric image created from $g(t) = sin(wt)$.

one variable, g such that:

$$f(x, y) = g(k_x x + k_y y) = g(\boldsymbol{k}^T \boldsymbol{r}) \tag{2.3}$$

where the direction of linear symmetry is $\pm\boldsymbol{k}$.

From equation 2.3, linearly symmetric images have the same value at all points $\boldsymbol{r}$ where $\boldsymbol{k}^T\boldsymbol{r} = C$. Since $\boldsymbol{k}^T\boldsymbol{r}$ describes a line in the image plane, it follows that the values of f along this line do not change and are equal to $g(C)$. These lines are designated the isocurves of f, and an alternative definition of linear symmetry is that all isocurves of f share a common direction. Figure 2.6 shows an example of a linearly symmetric image, generated from the function $g(t) = sin(wt)$, some of the image's isocurves are marked in pink and the vector $\boldsymbol{k}$ can be seen in green.

As demonstrated in [5], the Fourier transform of a linearly symmetric image is concentrated to a line. Additionally, if the Fourier transform of a an image is concentrated to a line, we can deduce that the image is linearly symmetric. This fact is used in Section 2.2.3 to derive the structure tensor.

2.2 State of the Art Disparity Estimation Methods

In this section, several state of the art disparity estimation methods are described. It is deemed important to provide a brief overview of different estimation methods to give an outlook of the field of Light Field based disparity estimation.

The plenoptic function was proposed by Adelson and Bergen in 1991 but little work was done in Light Field disparity estimation in that decade. In 2004, Dansereau and Bruton [16], presented a gradient-based method for disparity estimation, calculating the gradient vector of EPIs to estimate disparity. This method proved unreliable in areas of the light field with zero gradient. Wanner and Goldluecke [15] apply the structure tensor to the same EPIs. This structure tensor based approach is the basis for the main contributions of this thesis and will be described with detail in this chapter.

Since the higher availability of commercial Light Field cameras has driven more research studies, other initial estimation methods were proposed. The work of Tao *et al.* [4] in using focus as well as correspondence cues has been well regarded in the literature, and has been expanded upon in more recent methods, such as the approach of M. Strecke *et al.* [17] and Willem *et al.* [18]. As one of the basis for initial estimation, this section expands on the work of Tao *et al.*, explaining its basic framework of disparity estimation.

Several other initial estimation methods were developed, in this section we give relevance to a method by Tosic *et al.* [3], as it shows a somewhat different, but relevant approach to disparity estimation, borrowing from the theory of scale spaces.

Finally, a Matting based disparity map optimization step, first utilized by [13] is described as it is the optimization method chosen for the proposed disparity estimation framework.

While not presented here in detail, some other recent methods also achieve good results. One such method is the spinning parallelogram operator method [19], that uses the difference in the sum of different halves of a sliding and rotating parallelogram to infer the disparity at each point of an Epipolar plane image. Another method is the one developed by Alessandro Neri *et al.* [20], that utilizes a multi-resolution approach, foregoing the traditional EPI and multi-view stereo approaches.

2.2.1 Light-Field Scale and Depth Spaces

Making use of the EPI structure, the task of depth estimation can be formulated as a problem of estimating the angle of rays. Figure 2.7 shows an example EPI and how it can be subdivided into various rays, each corresponding to a similarly colored chunk in the image line being analyzed. Tosic *et al.*, in [3], use concepts of scale space to estimate depth from the EPI rays.

First presented in [21], scale space is a theory developed by the computer vision community, motivated by physics and biologic vision. The idea is to handle the multi-scale nature of the real-world. If one aims to create automatic algorithms for interpreting scenes of unknown origin, it is impossible to know beforehand the best scale for analysis. Scale

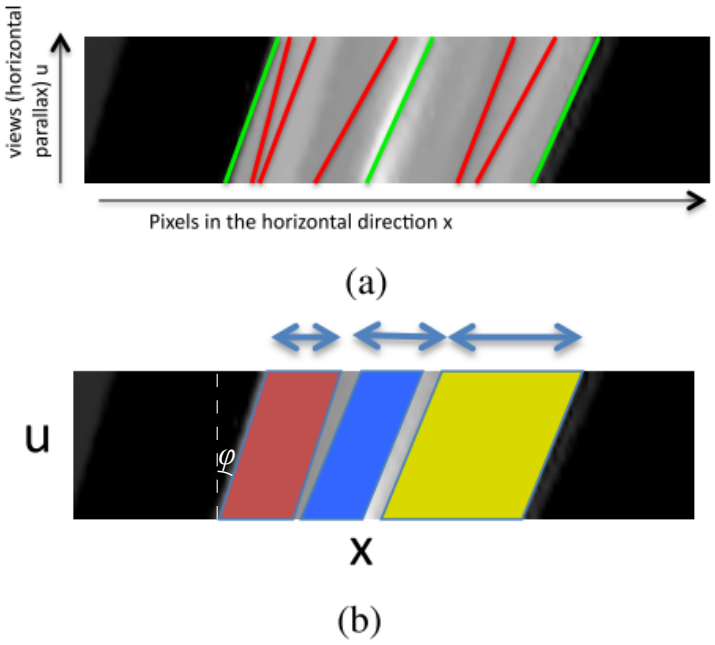


Figure 2.7: Diagram of Ray structures in EPIs. a) Ray edges, in green, can be easily estimated, yet the angle of uniform regions is ambiguous (gray lines). b) Tosic *et al.*'s approach of detecting entire Ray structures. (adapted from Tosic *et al.* [3])

space theory thus states that a logical approach is to analyze scenes at every possible scale.

Given a set of axioms, designed so that coarse scale representations should correspond to true simplifications of fine-scale structures, it has been shown that convolution with Gaussian Kernels and Gaussian derivatives provide an accurate scale-space representation. A detailed review of scale space by Tony Lindberg [22] is available for a more detailed overview of the theory.

In [23], Tosic *et al.* introduce the concept of Ray Gaussian Kernels,

$$\mathcal{R}_{\sigma,\varphi}(x,u) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\frac{(x+u\tan\varphi)^2}{\sigma^2}} \tag{2.4}$$

This kernel is Gaussian in the x direction, with scale σ and a slanted ridge in the vertical direction, with an angle equal to φ, with x and u being the kernel's coordinates in pixels.

Through convolution of an EPI image with this kernel, one can represent the Light Field scale and depth space $\mathcal{L}(x;\sigma,\varphi)$:

$$\mathcal{L}(x;\sigma,\varphi) = (I * \mathcal{R}_{\sigma,\varphi})(x,u)|_{u=0} \tag{2.5}$$

Where I represents an EPI and $(*)$ represents the convolution operator. Of the full two dimensional convolution, only the results for $u = 0$ are kept. This is equivalent to applying

convolution in a single direction. This is ideal since, excluding occlusions, features will be found in all views. Thus, it is ideal that $\mathcal{L}$ depends only on the spatial dimension x.

The Ray Gaussian Kernel, introduces an angle parameter to the regular scale-space representation. This angle can be related to the depth of the objects involved in the EPIs, it is thus correct to refer to the φ parameter as a depth parameter, and to $\mathcal{L}$ as a Light field scale and depth (Lisad) space representation of the Light Field.

Furthermore, it is expected that the inner product of an image ray of angle φ with a Ray Gaussian Kernel with the same angle should be independent of the value of φ, otherwise there would be a bias towards certain angles.

Considering an EPI with no occlusions $f_\varphi(x, u)$, which is similar to assuming the EPI is linearly symmetric, it is possible to embed this function in a one dimensional space so that $f_\varphi(x, u) = h(x + u\ tan\ \varphi)$, as mentioned in section 2.1.4. Tosic *et al.* use this to prove that the inner product of $h(x + u\ tan\ \varphi)$ with the Ray Gaussian, $\mathcal{R}_{\sigma,\varphi}(x, u)$, does not depend on φ, and therefore the angle invariant property of Lisad spaces.

Another important property of scale spaces is scale invariance. This property states that any scenic feature, like an image edge or an object, at scale σ should obtain the same response as that same feature at a larger scale $s\sigma$:

$$I(sx) * K_\sigma(x) = (I(x) * K_{s\sigma})(sx) \tag{2.6}$$

Where K is any kernel dependent on σ that satisfies the equation and s is an arbitrary real number denoting the scaling ratio.

Unlike in standard image scale-spaces, in Lisad spaces down-sampling in both directions would be ill-advised, as it would be equivalent to dropping views. Scale invariance is thus only analyzed in the x direction:

$$\mathcal{R}_{\sigma,\varphi}(x, u) = s\mathcal{R}_{s\sigma,\varphi'}(sx, u), \text{ where } \varphi = arctan(s\ tan\ \varphi), \varphi \in (-\pi, pi/2) \text{ and } s > 0 \tag{2.7}$$

Eq. 2.7 shows that a Ray Gaussian with scale σ and angle φ is equal to its down sampled in x by factor s version at scale $s\sigma$ and angle φ' with values multiplied by s.

Applying this property in Eq. 2.6, Tosic *et al.* proved that scale invariance holds for both simple Lisad spaces, as well as for spaces generated from the first derivative of the Ray Gaussian kernel, denominated Lisad-1 spaces. Furthermore, in [3] it is further proved that this assertion holds true for the normalized second derivative of the Ray Gaussian kernel, forming the Lisad-2 space. Figure 2.8 gives a visual intuition of the creation of the Lisad-2 as well as all the different cuts of the space.

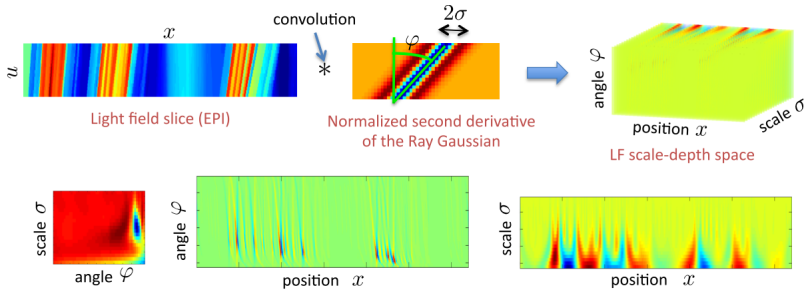


Figure 2.8: Creation and different cuts of the Lisad-2 space. (Tosic *et al.* [3])

Gaussian scale-spaces have been used extensively in the past to detect points of interest in images. Algorithms like Scale-invariant feature transform (SIFT) [24] and Speeded-UP Robust Features (SURF) [25], make use of the Difference of Gaussians(DoG) kernel to find blob features as extrema in an approximation of second derivative Gaussian scale-spaces.

Using a similar approach in the Lisad-2 space finds extrema points that correspond to rays in the EPI image. Each key point will be defined by its parameters in the Lisad-2 space, (x, σ, φ). These parameters hold information about the respective image ray: the position of the center of the ray x, the width of the ray 2σ and the angle of the ray φ. It is thus possible to create dense depth map from the extrema points of the Lisad-2 space.

Having detected and characterized EPI rays, there is a need of resolving conflicts between overlapping rays. Since angle invariance does not hold in case of occlusions, the value of the Lisad-2 space is not a good metric to choose the order of rays. The variance of the EPI image points that belong to a ray along it's direction ϕ is used, as a ray in the foreground will give a smaller variance, as it does not cross any occlusion boundaries. Since larger angle indicates a smaller depth (object closer to the camera), rays with larger angles should always be in the foreground. However, due to image noise, detected rays sometimes conform to an impossible ordering. When such cases are found, the "occluded" ray is removed.

Besides eliminating rays due to occlusion conflicts it is also necessary to eliminate rays with one weak edge. Those rays are often found next to object boundaries, so that one edge of the ray is an accurate object edge while the other is within an uniform background. To solve this problem, the Lisad-1 space is used. Much like extrema in the Lisad-2 space correspond to rays in the EPI, extrema in the Lisad-1 space correspond to edges [23]. A condition is imposed that the angle of each ray must be within a small threshold of the angle of both his corresponding edges.

Lisad space construction and occlusion detection are performed separately in the horizontal and vertical directions. To generate the dense disparity map, an angle value is associated with each pixel. Some pixels are assigned two disparity angles. The angle attributed by the ray with the largest associated Lisad-2 space value is chosen. After

this step there might still be pixels with no depth-value assigned. These images are inpainted using median filtering with masking of the missing regions. Finally, total variation denoising [26] is used to remove outliers.

2.2.2 Depth Estimation from Focus and Correspondence Cues

The algorithms so far described use only correspondence between the same feature in different view points to estimate depth from Light Field images. However, another important feature of Light fields is their ability to refocus images. It is thus possible to make use of defocus cues to aide in depth estimation.

Tao *et al.* in [4], describe a sound algorithm that combines defocus and correspondence cues to obtain a dense depth map. Defocus cues take advantage of the fact that when integrating over all views of an EPI, only objects with zero disparity (described by vertical EPI lines) will maintain their sharp borders. Correspondence cues are based on the idea of finding for all views, the position of the pixel that corresponds to the same feature, and analyzing its disparity. By slanting the EPI in a geometrical transformation designated shearing, Tao *et al.* manage to iteratively change which depth will create a zero disparity, or vertical lines. This is used to compute both defocus and correspondence responses, which are later merged. Shearing is performed, as proposed by Ng *et al.* [10]:

$$L_\alpha(x,u) = L_0(x + u(1 - \frac{1}{\alpha}), u) \tag{2.8}$$

where L_0 denotes the input EPI, and L_α denotes the EPI sheared by a value of *alpha*.

For each sheer value, the algorithm computes a defocus cue response. The first step is to integrate the sheared EPI across the angular u dimension,

$$\bar{L}_\alpha(x) = \frac{1}{N_u} \sum_{u'} L_\alpha(x,u'), \tag{2.9}$$

where N_u is the number of angular pixels. $\bar{L}_\alpha(x)$ is the refocused image for the sheer value α. The defocus response is measured on $\bar{L}_\alpha(x)$.

$$D_\alpha(x) = \frac{1}{|W_D|} \sum_{x' \in W_D} \Delta_x L_\alpha(x,u'), \tag{2.10}$$

where W_D is the size of the window, and Δ_x is the horizontal Laplacian operator. Windows with less contrast can be said to be less focused and will thus produce a smaller value of $D_\alpha(x)$, whereas high contrast areas will produce a larger metric. Thus, D_α is a measure of the defocus response for each α.

Certain high-frequency regions and bright lights may yield a high contrast even in out-

of-focus regions. This is a known problem of using solely defocus cues. Correspondence cues are thus necessary.

As an alternative to calculating the angle of EPI rays, as many other methods described in this chapter, Tao *et al.* take a different approach to calculating correspondence cues. When the shearing angle matches the angle of a ray, the vertical variance should be low, as all pixels in a vertical line should correspond to the same real-world feature, the variance is calculated as:

$$\sigma_\alpha(x)^2 = \frac{1}{|N_u|}\sum_{u'}(Ł_\alpha(x,u') - \bar{L}_\alpha(x))^2, \tag{2.11}$$

To increase robustness, the variance is averaged within a small window, to generate the correspondence metric:

$$C_\alpha(x) = \frac{1}{|W_C|}\sum_{x'\in W_C}\sigma_\alpha(x'), \tag{2.12}$$

where W_C is the window size around the current pixel.

Nevertheless, this metric presents some problems. Occlusions can cause impossible correspondence and large displacements are responsible for correspondence errors due to a limited search space. Matching can also be ambiguous in areas with repeated patterns and in noisy regions.

Since defocus cues perform better in repeating textures and noise, while correspondence is robust around bright areas and features, combining both cues is ideal.

The objective is to maximize spatial contrast for defocus while minimizing angular variance for correspondence across all shears,

$$\begin{aligned}\alpha_D^*(x) &= \arg\max_\alpha D_\alpha(x)\\ \alpha_C^*(x) &= \arg\min_\alpha C_\alpha(x)\end{aligned} \tag{2.13}$$

Defocus and correspondence cues might not agree. This is addressed by using Peak Ratio [27] as a measure of the confidence of $\alpha_D^*(x)$ and $\alpha_C^*(x)$,

$$\begin{aligned}D_{conf}(x) &= D_{\alpha_D^*}(x)/D_{\alpha_D^{*2}}(x)\\ C_{conf}(x) &= C_{\alpha_C^*}(x)/C_{\alpha_C^{*2}}(x)\end{aligned} \tag{2.14}$$

where α^{*2} is the next largest peak or dip. Thus, this measure produces higher values the more the maximum is higher than any other value.

Markov Random Fields(MRF) propagation is used to propagate the results and choose the optimal result given both defocus and correspondence measures and their respective confidence values. The two estimations are concatenated as follows to facilitate the sum-

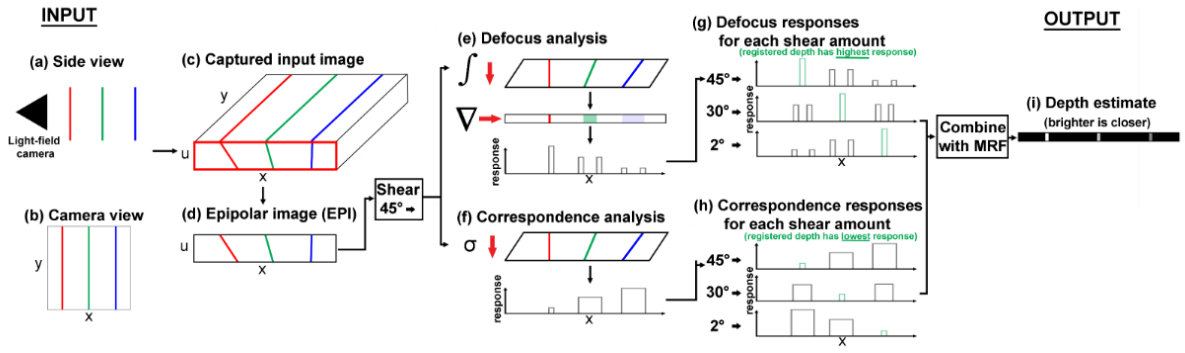


Figure 2.9: Copied from the original paper. Here we cite the original caption: *Framework. This setup shows three different poles at different depths with a side view of (a) and camera view of (b). The light-field camera captures an image (c) with its epipolar image (EPI). By processing each row's EPI (d), we shear the EPI to perform refocusing. Our contribution lies in computing both defocus analysis (e), which integrates along angle u (vertically) and computes the spatial x (horizontal) gradient, and correspondence (f), which computes the angular u (vertical) variance. The response to each shear value is shown in (g) and (h). By combining the two cues using Markov random fields, the algorithm produces high quality depth estimation (i).* (Tao *et al.* [4])

ming notation in Equation 2.16:

$$\begin{aligned} \{Z_1^{source}, Z_2^{source}\} &= \{\alpha_C^*, \alpha_D^*\} \\ \{W_1^{source}, W_2^{source}\} &= \{C_{conf}, D_{conf}\} \end{aligned} \tag{2.15}$$

where source is used to denote the initial data term. The following equation describes the minimization process:

$$\begin{aligned} \underset{Z}{\text{minimize}} \sum_{source} \lambda_{source} \sum_i W_i^{source} |Z_i - Z_i^{source}| \\ + \lambda_{flat} \sum_{(x,y)} \left(\left| \frac{\partial Z_i}{\partial x} \right|_{(x,y)} + \left| \frac{\partial Z_i}{\partial y} \right|_{(x,y)} \right) \\ + \lambda_{smooth} \sum_{(x,y)} |(\Delta Z_i)|_{(x,y)} \end{aligned} \tag{2.16}$$

λ_{source} controls the weight between defocus and correspondence. λ_{flat} controls the Laplacian constraint for flatness of the output depth estimation map, while λ_{smooth} controls the second derivative kernel, which enforces smoothness.

Minimizing Equation 2.16 provides the result Z^*, which may deviate from source, flatness and smoothness constraints. The result is improved by finding the error between Z^* and the optimization constraints, and iterating the minimization function, using Z^* as the input, and the error as the weight matrix. This process continues until the Root

Mean Square Error(RMSE) of the result to the previous iteration is below a predefined threshold, indicating that the result has converged.

Figure 2.2.2 shows an overview of the entire framework, starting with the estimation of both metrics and finalizing with MRF propagation of the results.

2.2.3 Structure Tensor Based Estimation

In Section 2.1.4, the concept of linearly symmetric function was introduced. In this section, a method for detecting linear symmetry in images and estimating its direction is described. Additionally, the general problem is reduced to the EPI case where the expressions used to translate the structural information into a depth value are described.

Let the scalar function $f(\boldsymbol{r})$, represent an image with coordinates $\boldsymbol{r} = (x, y)^T$. Let F denote the Fourier transform of f. $|F(\boldsymbol{\omega})|$ denotes the magnitude of the spectrum of f, where $\boldsymbol{\omega} = (\omega_x, \omega_y)$ represents the Fourier transform coordinates in angular frequency. $|F(\boldsymbol{\omega})|^2$ denotes the power spectrum of f.

The direction of a linearly symmetric function $f(\boldsymbol{r}) = g(\boldsymbol{k}^T\boldsymbol{r}))$ is well defined from the vector $\boldsymbol{k}$. It is also known that if and only if f is linearly symmetric, its power spectrum $|F(\boldsymbol{\omega})|^2$ is concentrated to a line in the $\omega_x\omega_y$ plane with direction $\boldsymbol{k}$.

Thus it is possible to detect if an image is linearly symmetric with direction $\boldsymbol{k}$ by analyzing the error of the fit of its power spectrum $|F(\boldsymbol{\omega})|^2$ to a line with that same direction. This can be done in the total least squares sense.

This problem has similarities with the common problem of fitting an axis to a finite set of points. The axis fitting problem is classically solved by minimizing the error function:

$$e(\boldsymbol{k}) = \sum_{\omega} d^2(\boldsymbol{\omega}, \boldsymbol{k}) \tag{2.17}$$

where $d(\boldsymbol{\omega}, \boldsymbol{k})$ is the shortest distance from any point $\boldsymbol{\omega}$ to a candidate axis $\boldsymbol{k}$. The projection of ω on the unitary vector $\boldsymbol{k}$ is equal to $\langle\boldsymbol{\omega}, \boldsymbol{k}\rangle\boldsymbol{k} = (\boldsymbol{\omega}^T\boldsymbol{k})\boldsymbol{k}$. The vector $\boldsymbol{d}$, represents the difference between $\boldsymbol{\omega}$ and its projection, so that:

$$\boldsymbol{d} = \boldsymbol{\omega} - (\boldsymbol{\omega}^T\boldsymbol{k})\boldsymbol{k} \tag{2.18}$$

with the norm $\|\boldsymbol{d}\| = d(\boldsymbol{\omega}, \boldsymbol{k})$. The square of the norm $d.^2(\boldsymbol{\omega}, \boldsymbol{k})$ is thus equal to:

$$\begin{aligned} d^2(\boldsymbol{\omega}, \boldsymbol{k}) &= \|\boldsymbol{\omega} - (\boldsymbol{\omega}^T\boldsymbol{k})\boldsymbol{k}\|^2 \\ &= (\boldsymbol{\omega} - (\boldsymbol{\omega}^T\boldsymbol{k})\boldsymbol{k})^T(\boldsymbol{\omega} - (\boldsymbol{\omega}^T\boldsymbol{k})\boldsymbol{k}) \end{aligned} \tag{2.19}$$

Equation 2.17 is defined for a sparse point set, while the power spectrum of f is a

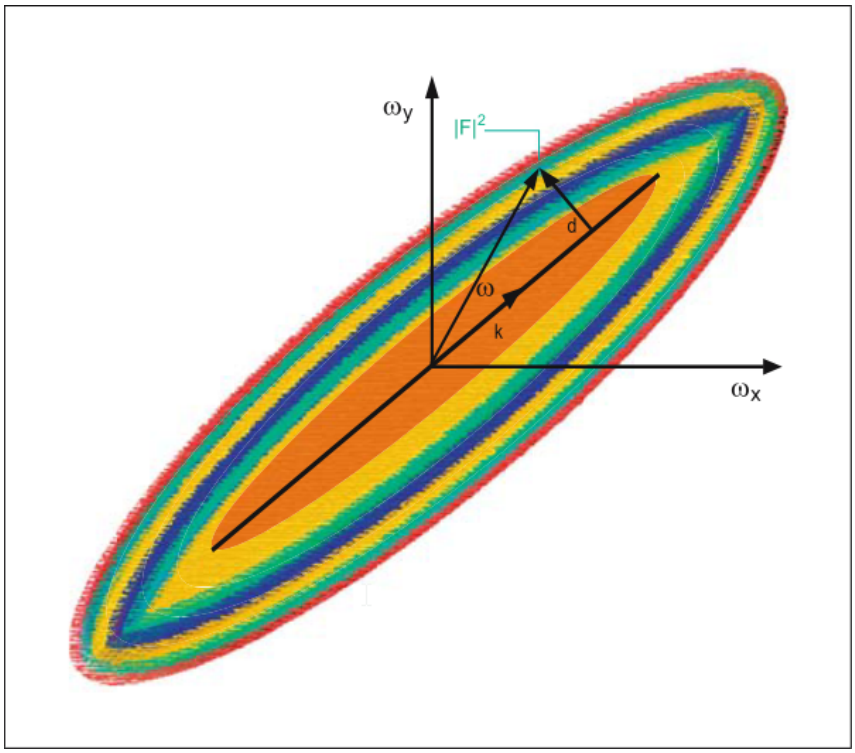


Figure 2.10: Line Fitting Process, (J. Bigun [5])

continuous, real-valued, function. A generalization to dense point sets is needed. A simple approach is to weight the contribution of the square distance at any $\boldsymbol{\omega}$ with the value of the power spectrum at that same point $|F(\boldsymbol{\omega})|^2$, and integrate over all contributions

$$e(\boldsymbol{k}) = \int d^2(\boldsymbol{\omega}, \boldsymbol{k}) |F(\boldsymbol{\omega})|^2 d\boldsymbol{\omega} \tag{2.20}$$

where the integral is a double integral over E_2 and $d\boldsymbol{\omega}$ is equal to $d\omega_x d\omega_y$.

Figure 2.2.3 provides an intuition for Equation 2.20. As can be seen, the vector $d(\boldsymbol{\omega}, \boldsymbol{k})$ represents the distance of any point $\boldsymbol{\omega}$ to a line with direction $\boldsymbol{k}$ passing through the origin. In that sense, the total least squares error $e(\boldsymbol{k})$ integrates over the entire domain of F, the power spectrum $|F(\boldsymbol{\omega})|^2$, weighed by the distance of each point to the line with direction $\boldsymbol{k}$.

The contribution of the power spectrum $|F(\boldsymbol{\omega})|^2$ is null in points along the axis $\boldsymbol{k}$, as the distance $d^2(\boldsymbol{\omega}, \boldsymbol{k})$ will be zero. The contribution of the power spectrum will also increase as the square distance of a point $\boldsymbol{\omega}$ from the axis increases. Therefore, if we assume a non-trivial F, $e(\boldsymbol{k})$ will only be zero, if $|F(\boldsymbol{\omega})|^2$ is zero everywhere except along the axis $\boldsymbol{k}$. It can thus be stated that F is only concentrated to a line if $e(\boldsymbol{k})$ is zero for some $\boldsymbol{k}$.

From the fact that $\boldsymbol{\omega}^T\boldsymbol{k}$ is a scalar and equal to $\boldsymbol{k}^T\boldsymbol{\omega}$, and that $\|\boldsymbol{k}\|^2 = \boldsymbol{k}^T\boldsymbol{k} = 1$, it is

possible to write $d^2(\boldsymbol{\omega}, \boldsymbol{k})$ in quadratic form

$$d^2(\boldsymbol{\omega}, \boldsymbol{k}) = \boldsymbol{k}^T(\boldsymbol{I}\boldsymbol{\omega}^T\boldsymbol{\omega} - \boldsymbol{\omega}\boldsymbol{\omega}.^T)\boldsymbol{k} \tag{2.21}$$

where $\boldsymbol{I}$ is a 2×2 identity matrix.

It is thus possible to express Equation 2.20 as:

$$e(\boldsymbol{k}) = \boldsymbol{k}^T\boldsymbol{J}\boldsymbol{k} = \boldsymbol{k}^T(\boldsymbol{I} \cdot Trace(\boldsymbol{S}) - \boldsymbol{S})\boldsymbol{k} \tag{2.22}$$

where

$$S = \int \boldsymbol{\omega}\boldsymbol{\omega}^T|F|^2 d\boldsymbol{\omega} = \begin{pmatrix} \boldsymbol{S}(1,1) & \boldsymbol{S}(1,2) \\ \boldsymbol{S}(2,1) & \boldsymbol{S}(2,2) \end{pmatrix} \; with \; \boldsymbol{S}(i,j) = \int \omega_i\omega_j|F(\boldsymbol{\omega})|^2 d\boldsymbol{\omega} \tag{2.23}$$

where the components of $\boldsymbol{\omega} = (\omega_x, \omega_y)^T$ are defined as $\omega_x = \omega_1$ and $\omega_y = \omega_2$.

The matrix $\boldsymbol{S}$ is called the structure tensor of the image f. The structure tensor can be readily obtained from $\boldsymbol{J}$ and vice versa.

The problem is thus reduced to minimizing the quadratic form $\boldsymbol{k}^T\boldsymbol{J}\boldsymbol{k}$. This is done by choosing $\boldsymbol{k}$ as the least eigenvector of $\boldsymbol{J}$.

From Equation 2.22 it is known that $\boldsymbol{J} = \boldsymbol{I} \cdot Trace(\boldsymbol{S}) - \boldsymbol{S}$. It follows that, for eigenvector eigenvalue pairs of $\boldsymbol{J}$ $\{\lambda_1, \boldsymbol{u}_1\}$ and $\{\lambda_2, \boldsymbol{u}_2\}$, and of $\boldsymbol{S}$ $\{\lambda_1', \boldsymbol{u}_1'\}$ and $\{\lambda_2', \boldsymbol{u}_2'\}$

$$\{\lambda_1', \boldsymbol{u}_1'\} = \{\lambda_1, \boldsymbol{u}_2\}, \quad and \quad \{\lambda_2', \boldsymbol{u}_2'\} = \{\lambda_2, \boldsymbol{u}_1\} \tag{2.24}$$

Thus, the eigenvectors of $\boldsymbol{J}$ and $\boldsymbol{S}$ are equal, with their corresponding eigenvalues switched. Therefore, as we've proved the $\boldsymbol{J}$ contains enough information to estimate the linear symmetry of f, the same is also true for the structure tensor $\boldsymbol{S}$.

The structure tensor $\boldsymbol{S}$ is defined in the frequency domain, which is computationally expensive if we want to compute it multiple times for various images, as is the case for disparity estimation of Light Field Images. However, the Parseval-Plancherel theorem,

$$\int_{-\infty}^{\infty} f(t)^* g(t) dt = 2\pi \int_{-\infty}^{\infty} F(\omega)^* G(\omega) d\omega \tag{2.25}$$

where $(\cdot)^*$ refers to the complex conjugate of a function, states that the inner product is maintained under the Fourier Transform. Applying the theorem to $\boldsymbol{S}(i,j)$ gives us

$$\boldsymbol{S}(i,j) = \int \omega_1\omega_2|F|^2 d\boldsymbol{\omega} = \frac{1}{4\pi^2} \int \frac{\partial f}{x_i}\frac{\partial f}{x_j} d\boldsymbol{x} \qquad i,j : 1,2 \tag{2.26}$$

where $x_1 = x$, $x_2 = y$, $d\boldsymbol{x} = dxdy$ and the integral is a double integral over the entire 2D

plane. In matrix form, $\boldsymbol{S}$ can be restated as:

$$\boldsymbol{S} = \frac{1}{4\pi^2} \int (\nabla f)(\nabla^T f) d\boldsymbol{x} \tag{2.27}$$

While the least eigenvalue of $\boldsymbol{J}$ will minimize Equation 2.20, if the image, f does not contain any linear symmetry, this direction is not necessarily unique. In [12], it is demonstrated that in these cases, the least eigenvector of $\boldsymbol{J}$ will have a multiplicity higher than one. In the two dimensional case, this means $\lambda_0 \approx \lambda_1$, while a true optimal orientation occurs when $\lambda_0 = 0$. Thus a coherence or reliability function can be defined, that gives a higher reliability to the orientation estimation when the least eigenvalue is zero, and a lower reliability when the difference between the two eigenvalues is small,

$$r = \left(\frac{\lambda_1 - \lambda_0}{\lambda_1 + \lambda_0}\right)^2 s = \frac{(J_{yy} - J_{xx})^2 + 4J_{xy}^2}{(J_{xx} + J_{yy})^2}, \tag{2.28}$$

This reliability measure provides a maximum value of one when $\lambda_0 = 0$ and a value approaching zero as the difference $\lambda_1 - lambda_2$ increases.

So far, the structure tensor has been defined as a tool to work on continuous signals. For discrete signals such as EPIs, an approximation is required. For that purpose, discrete convolution with Gaussian kernels is used as a form of sampling the derivatives of a discrete image I in an accurate manner. The structure tensor of a discrete image is thus calculated as

$$\mathcal{S}(x, y) = \begin{pmatrix} G_{\sigma_o} * D_x^2 & G_{\sigma_o} * (D_x D_y) \\ G_{\sigma_o} * (D_x D_y) & G_{\sigma_o} * D_y^2 \end{pmatrix} = \begin{pmatrix} \mathcal{S}_{xx} & \mathcal{S}_{xy} \\ \mathcal{S}_{xy} & \mathcal{S}_{yy} \end{pmatrix} \tag{2.29}$$

where D_{x_i} is the derivative of the smoothed image D in the x_i direction, with $x_1 = x$ and $x_2 = y$ and G_{σ_o} represents a discrete Gaussian kernel with a scale σ_o. D represents a filtered version of the original image I obtained by means of a Gaussian averaging at an inner scale σ_i (i.e. $K = G_{\sigma_i} * I$).

According to the study made by Wanner in [2], the values for σ_i that generate better results are in a range between 0.6 and 0.8 while optimal values for σ_o are tendentially larger, with acceptable values ranging between 1.3 and 1.7 for the images tested.

$\mathcal{S}$ estimates the structure tensor for a window of size equal to the size of the outer Gaussian Kernel G_{σ_o} around all points of the image I.

There are several different methods proposed to compute the values of D_x and D_y. The most straightforward involve first convolving the image I with the inner Gaussian kernel to obtain a smoothed image. Then convolution filters are used to estimate the derivatives. A variety of filters have been proposed of which two have achieved the most

relevance, the Sobel filter and the Sharr filter [28]. This filters for the horizontal direction are

$$\text{Sobel: } \mathcal{H}_x = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix}$$

$$\text{Sharr: } \mathcal{H}_x = \begin{pmatrix} 3 & 0 & -3 \\ 10 & 0 & -10 \\ 3 & 0 & -3 \end{pmatrix}$$

and in the vertical direction $\mathcal{H}_y = \mathcal{H}_x^T$. The Sharr operator optimizes the rotational symmetry of D and therefore should be better suited for estimating the orientation of lines. D_{x_i} is thus equal to

$$D_{x_i} = I * G_\sigma * \mathcal{H}_{x_i} \tag{2.30}$$

Alternatively, using the properties of convolution, we have that

$$\frac{\partial (I * G_\sigma)}{\partial x_i} = I * \frac{\partial G_\sigma}{\partial x_i} \tag{2.31}$$

where the first derivative of the Gaussian function $\frac{\partial G_\sigma}{\partial x_i}$ can be obtained from sampling the continuous normalized first derivative of the Gaussian function.

Applying Equation 2.29 to an EPI $I(x, u)$, it is possible to obtain the disparity, $d(x, u)$, for all image points, by calculating the direction of the eigenvector that minimizes Equation 2.22. In [14], a direct formula for calculating disparity is demonstrated, obtaining

$$d = \frac{\mathcal{S}_{yy} - \mathcal{S}_{xx} + \sqrt{(\mathcal{S}_{yy} - \mathcal{S}_{xx})^2 + 4\mathcal{S}_{xy}^2}}{2\mathcal{S}_{xy}}. \tag{2.32}$$

This proves computationally advantageous to computing the eigenvectors generally.

2.2.4 Image Matting and Disparity Map Optimization

Image matting is an image processing field with the purpose of achieving foreground and background separation. This problem is simplified to the estimation of an alpha-matte, a gray image that quantifies how much each pixel belongs to the foreground of an image. This problem is summarized in what's called the matting equation:

$$I_i = \alpha_i F_i + (1 - \alpha_i) B_i \tag{2.33}$$

where I identifies the image, F represents an image that contains only the foreground contents of I, B represents an image that contains only the background elements of I, α is the alpha-matte image and the index i indicates a pixel of the image.

J. Li *et al.* in [13] use an analogous function that equates the matting problem with the disparity estimation problem. Similarly to how an image can be decomposed into it's background and foreground elements, it can be decomposed into the elements closer and farther from the camera, at which point depth or disparity can be thought off as an alpha matte of the image.

The matting equation is severely unconstrained, therefore assumptions and prior information are required to obtain an alpha matte. In [29] a closed form solution to the matting equation is derived. Admitting only gray scale images, for clarity of explanations, one can postulate that small windows in a gray scale image always have constant background B and foreground F.

This allows the rewriting of Equation 2.33 as:

$$\alpha_i \approx aI_i + b, \forall i \in w. \tag{2.34}$$

where $a = \frac{1}{F-B}, b = -\frac{B}{F-B}$ and w is a small window around each pixel of image I.

This equation is still unconstrained, but a closed form solution can be found by minimizing the corresponding cost function:

$$J(\alpha, a, b) = \sum_{j \in I} \Big(\sum_{(i \in w_j)} (\alpha_i - a_j I_i - b_j)^2 + \epsilon a_j^2 \Big) \tag{2.35}$$

where w_j is a small window around the pixel j. The cost function includes a regularization term on a. This term is added mostly for numerical stability, however, minimizing the norm of a biases the solution towards smoother α mattes ($a_j = 0$ implies that α is constant over the j^{th} window). The parameter ϵ controls the smoothness of the result, with lower values implying a smoother disparity map.

Equation 2.35 can be re-written in matrix form as:

$$J(\alpha, a, b) = \sum_k \left\| G_k \begin{pmatrix} a_k \\ b_k \end{pmatrix} - \bar{\alpha}_k \right\|^2 \tag{2.36}$$

where for every window w_k, G_k is defined as a $(card(w_k)+1) \times 2$ matrix. For each $i \in w_k$, G_k contains a row of the form $(I_i, 1)$ and the last row of G_k has the form $(\sqrt{\epsilon}, 0)$. $\bar{\alpha_k}$ is a $(card(w_k)+1) \times 1$ vector, whose entries are α_i for each i in w_k and whose last entry is 0.

In the [29], Levin *et al.* derive the optimal (minimal) pair (a_k^*, b_k^*) for each matte α by

minimizing the above matrix in a least squares sense and obtaining:

$$J(\alpha) = \sum_k \bar{\alpha}_k^T \bar{G}_k^T \bar{G}_k \bar{\alpha}_k \tag{2.37}$$

where $\bar{G}_k = I - G_k(G_k^T G_k)^{-1} G_k^T$.

Finally, rewriting $L = \bar{G}_k^T \bar{G}_k$, simplifies the cost function:

$$J(\alpha) = \alpha^T L \alpha, \tag{2.38}$$

where L is designated the Laplacian Matrix, an $N \times N$ matrix whose (i, j) entry equals:

$$\sum_{k|(i,j)\in w_k} \Big(\delta_{i,j} - \frac{1}{|w_k|}\Big(1 + \frac{1}{\frac{\epsilon}{|w_k|} + \sigma_k^2}(I_i - \mu_k)(I_j - \mu_k)\Big)\Big) \tag{2.39}$$

where μ_k and $sigma_k$ represents respectively the expected value and the standard deviation of the window k.

Further extending this derivation to colored images relaxes the uniformity constraint. Instead of the background B and foreground F being required to be even in every window w_k, they are only required to respect the color-line model.

According to the color-line model, it is accurate to predict that all colors present in a small window of an image are a linear combination of two different colors. Or in other words, if the color samples present in a small window were plotted in the RGB cube, the samples would form a line:

$$\begin{aligned} F_i &= \beta_i^F F_1 + (1 - \beta_i^F) F_2 \\ B_i &= \beta_i^B B_1 + (1 - \beta_i^B) B_2 \end{aligned} \tag{2.40}$$

where F_1 and F_2, and B_1 and B_2 are vectors representing colors and β represents the weight of each of the two colors that compose the foreground and background images and i represents a color in RGB space.

As can be seen in figure 2.11 this approximation is very accurate for small windows, even across image edges.

Levin *et al.* further demonstrate that if the color line model is verified, the conclusions taken from the gray-scale case are still valid, with slight adjustments to the elements of the Laplacian Matrix L, so that they equal:

$$\sum_{k|(i,j)\in w_k} \Big(\delta_{i,j} - \frac{1}{|w_k|}(1 + (I_i - \mu_k)(\Sigma_k + \frac{\epsilon}{|w_k|} I_3)^{-1}(I_j - \mu_k))\Big) \tag{2.41}$$

where Σ_k is a 3×3 covariance matrix, μ_k is a 3×1 mean vector of the colors in the

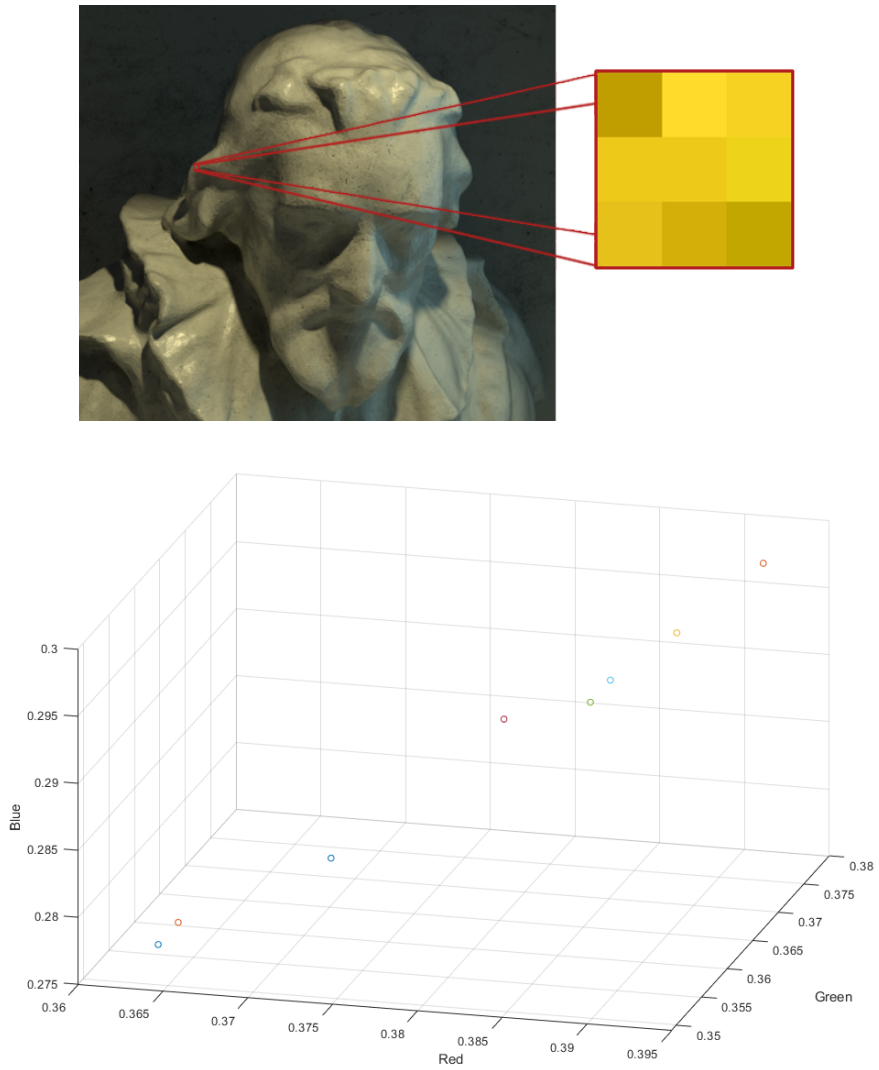


Figure 2.11: Test of the color-line assumption. A window from the *Cotton* image is evidenced. Below the color values of the window are plotted in the RGB cube.

window k and I_3 is the identity matrix.

While Levin *et al.* utilize scribbles as a constraint for the matting problem, in the case of disparity map optimization the initial estimation of the disparity map serves as the initial matrix, weighed with the confidence value for each estimate if such value is available.

In [13], J. Li *et al.* define the energy function as:

$$J(d) = d^T L d + \lambda (d - \bar{d})^T C (d - \bar{d}), \tag{2.42}$$

where d and $\bar{d}$ represent $1 \times N$ vectors, with $N = Height \times Width$, that represent the optimal and initial disparity estimation, respectively. C is a diagonal $N \times N$ matrix with the reliability values of each disparity estimate, L is the Laplacian matrix. λ weighs the importance of the data term.

The first term of J is the smoothing term, which is smaller for a smoother optimal disparity d, while the second term is a data term that constrains the optimal disparity to be similar to the initial estimation $\bar{d}$. This energy function describes a convex function and therefore can be minimized by finding the zero of its derivative.

To minimize the run time and memory usage of the algorithm, as well as to avoid over smoothing of edges, segmentation of the initial image utilizing algorithms such as mean shift segmentation is performed, and each segment is optimized separately. This process obtains mixed results, as will be discussed in Chapter 3.

Chapter 3

Improvements to the Structure Tensor Based Approach

This chapter describes contributions that aim to improve existing methods of light field disparity estimation. First, the decision for improving the structure tensor approach is justified by highlighting the advantages of the method and the systematic nature of the estimation errors that are found in state-of-the-art estimation methods based on the structure tensor. Then, the methods developed in this work to improve silhouette precision and preserving angular continuity in planar regions are described. The results of these improvements are presented and discussed in Chapter 4.

3.1 Advantages of The Structure Tensor Approach

Described in section 2.2.3, the structure tensor approach is a fast method that is able to accurately estimate the disparity of light field images from its EPIs. The low computational complexity of this initial processing step allows the implementation of further complex improvement algorithms while keeping the total running time within acceptable bounds.

When compared to other state of the art methods, the initial disparity estimation provided by the structure tensor proved to be at least similar to all other methods when excluding occlusion areas. Therefore, improving the prediction accuracy in problematic areas of the scene is expected to contribute towards the development of a competitive framework.

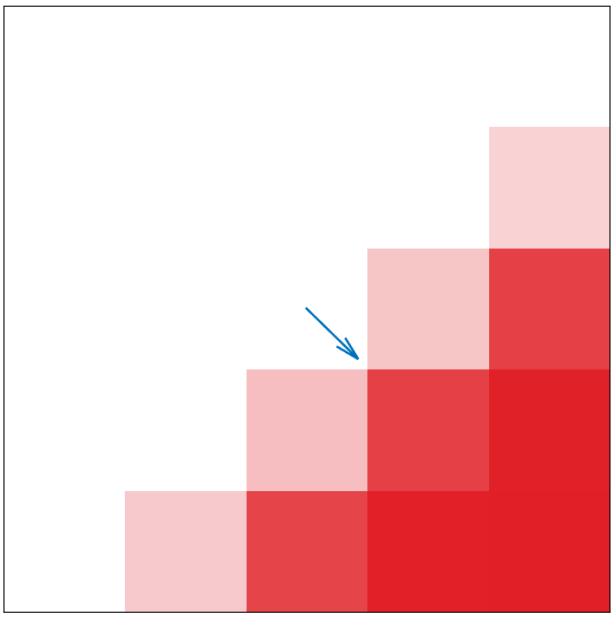


Figure 3.1: A 5×5 detail of an image edge. The arrow shows the direction of linear symmetry in the window, estimated using the structure tensor.

3.2 Systematic Errors of the Structure Tensor Approach

In this chapter, two limitations of the structure tensor initial estimation are addressed. Firstly silhouette enlargement is described, explaining the origin of this artifact and justifying the necessity of a secondary processing step to minimize its effect. Secondly, the problem of low amplitude local noise in structure tensor-based disparity estimations is addressed, explaining its effect in the computation of the surface normals of any given scene.

3.2.1 Silhouette Enlargement

When capturing a scene with various objects along a given direction, only those located closer to the camera plane will be definitely free from occlusions. On the EPI representation of a light field, the color difference between two points inherent to the transitions between such objects appears as complete and well defined lines spanning over all views contained in the EPI. The opposite stands for objects partially occluded, or even for texture patterns on the background plane, where the corresponding EPI lines might be missing for some views.

As explained in Section 2.2.3, the structure tensor is computed in sliding windows of the Light Field's EPIs, and the corresponding direction of linear symmetry is assigned to

the center pixel of each window. Then, disparity is estimated from the linear symmetry.

An example of a 5×5 window from an EPI, containing an edge, is shown in Figure 3.1. This can be used to explain why the structure tensor on its own does not always provide an accurate disparity estimation near such edges. Assuming that the center pixel is located on a background region (white area), and given that its linear symmetry is strongly influenced by the shadowed (red) pixels of the foreground, as they form an edge, then the disparity is wrongly assigned. This happens because a background pixel is assigned the disparity of the foreground region.

Since the distinction between foreground and background is not known while computing the structure tensor, it is not possible to correct the disparity estimate at this processing stage. The disparity calculated in the remaining windows of the EPI must be known to resolve this ambiguity.

The problem described above can be observed in Figure 3.2, which shows the disparity estimation obtained from the structure tensor (top left), the ground truth (top right), and their difference (bottom). The difference between the ST estimation and the ground truth shows that the foreground object in the former is larger than in the latter, which is an artifact known as silhouette enlargement. Overall this corresponds to obtaining disparity maps where occluding objects are enlarged due to spreading their disparity into the surrounding occluded regions. This problem is exacerbated by the multiple smoothing steps necessary to obtain accurate derivative information from a discrete image.

In [13], a penalty measure for the reliability (r) of the structure tensor was proposed as an attempt to overcome this silhouette enlargement artifact. However, this proved to be highly sensitive to the texture in the background regions neighboring object edges, providing only a minimal penalty when the algorithm fails in regions of uniform texture. This chapter presents a more reliable method of Silhouette Enhancement in Section 3.3.

3.2.2 Low amplitude local noise

The structure tensor provides accurate local estimations in most areas of the image. However, while the error in each estimation is low, this error exists and is uncorrelated with the error of neighboring pixels. The result is a large error when attempting to estimate the orientation of object surfaces in any light field image. Figure 3.3 compares the disparity along a single horizontal line in the *Cotton* light field image. The figure shows the ground truth disparity and the disparity estimated from the structure tensor. In the structure tensor estimation, one can see some high amplitude errors, but mostly low amplitude local noise, that is not present in the ground truth disparity.

A normal map $N(x, y)$ is an image that associates each image point (x, y) with a three

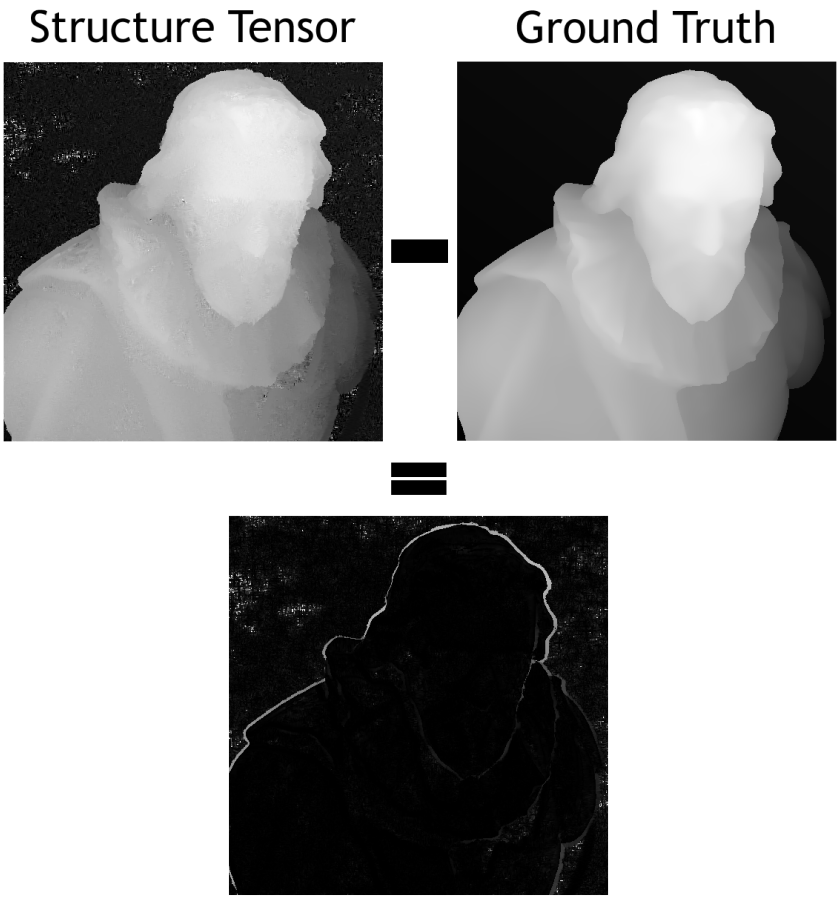


Figure 3.2: The difference image between the base disparity estimation and the ground truth showcases the enlarged silhouettes.

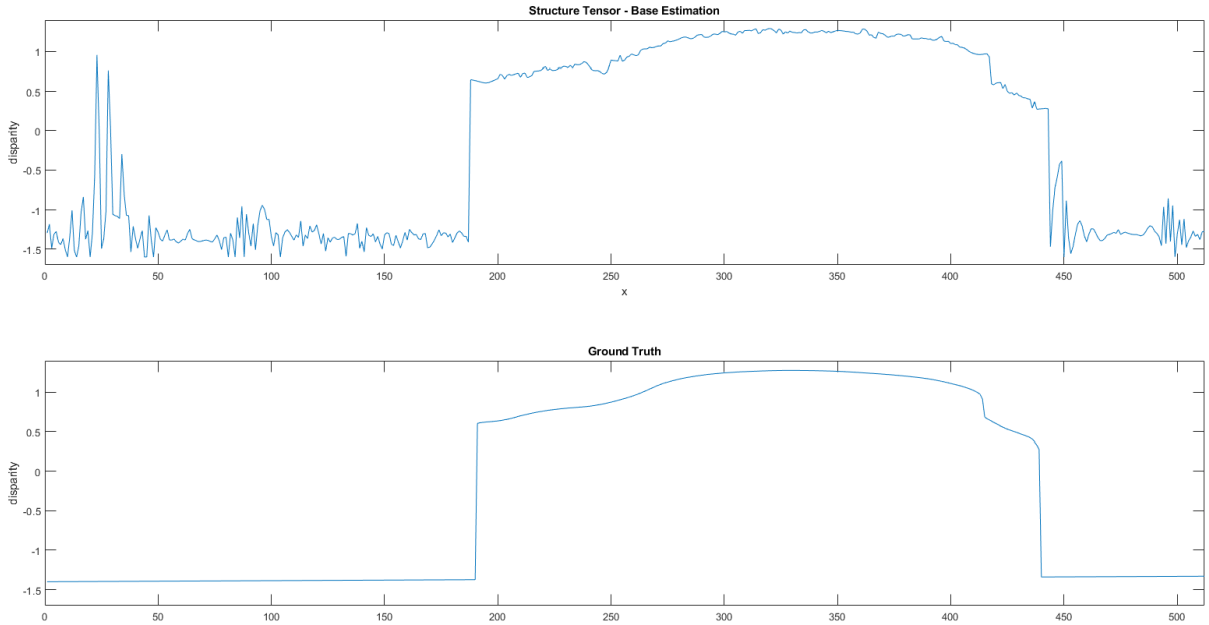
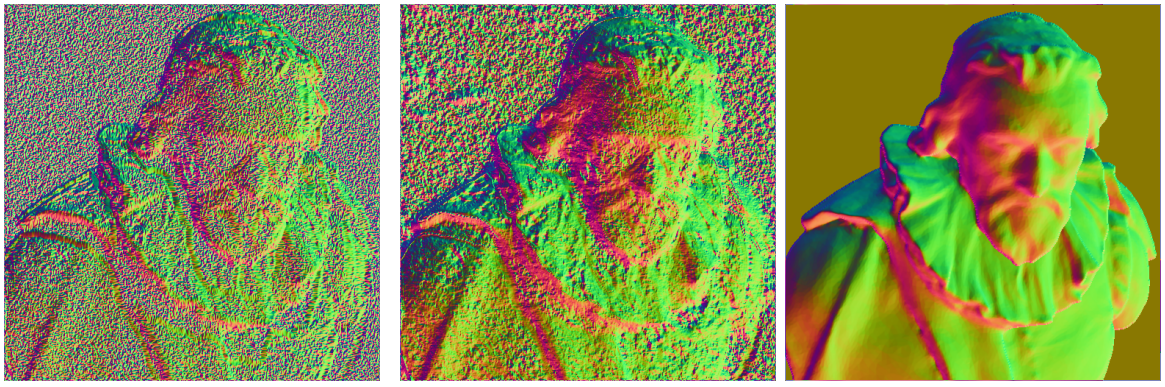


Figure 3.3: Disparity along one horizontal line (*Cotton* dataset). Above; structure tensor estimation. Below: the ground truth disparity.



(a) Structure Tensor (b) Matting Optimized (c) Ground Truth

Figure 3.4: Comparison between different normal maps.

dimensional vector $\boldsymbol{n}$ that is orthogonal to the corresponding surface in the image. The normal vector of a pixel is calculated as the cross product of the difference vectors from said pixel to a vertical and an horizontal adjacent pixel.

In the structure tensor disparity estimation, the difference between two neighboring pixels depends heavily on the estimation error, and not on the real orientation of the surface they represent. Therefore, the normal map is noisy and inaccurate, which can be harmful for several applications of disparity maps. Applications like image segmentation, scene reconstruction or construction of realistic normal maps for 3D modeling rely on locally smooth disparity maps for optimal results.

Global optimization steps, such as graph cuts [30] and belief propagation, [31] are the most common method used to reduce local noise. The framework presented in this focuses on the Laplacian Matting based optimization step, described in Section 2.2.4. While the energy function is minimized by enforcing similarity between neighboring pixels, reducing noise, it does not enforce a similarity between the difference of the pixels in the same surface. Thus, while the amplitude of local noise is greatly reduced, and the accuracy of the algorithm is improved, some low amplitude local noise remains, such that normal maps obtained from the optimized disparity map are still highly inaccurate, this is shown in Figure 3.4 [1]. The figure shows a comparison between a normal map obtained from the structure tensor depth estimation, the matting optimized disparity estimation and the ground truth disparity. For viewing purposes, the normal vector coordinates are normalized between zero and one, and displayed as an RGB image. In this way, the normal vector $\boldsymbol{n} = (0, 0, -1)^T$, referring to a surface parallel to the image plane, is represented by the RGB color #7F7F00, or a brown green color. The influence of low amplitude local noise is clear in the background of the image, where the ground truth normal map shows constant normal vectors, the normal vectors in the structure tensor estimation are both

[1] This figure must be observed in color.

noisy and different from the ground truth normal direction. In section 3.4, a new method for achieving local smoothness in planar regions of the image is presented.

3.3 Silhouette Enhancement

The method herein proposed for silhouette enhancement involves detecting the edges of the disparity map, obtained from Equation 2.32, and finding the corresponding object boundary in the edges detected in the original EPI representation. Whenever an edge on the disparity map is in a different position than the corresponding edge in the EPI, a correction is applied using the neighboring values with high reliability estimations. Additionally, a Laplacian matting-based optimization is applied to smooth object contours so as to reduce the impact of artifacts created by erroneous discontinuities in the detected edges. This includes an improvement to the method presented in [6], adding an enhancing step that reduces the number of artifacts created by the silhouette enhancement process. The proposed method is schematically presented in Figure 3.5, and its main steps are further described as follows.

3.3.1 ST-based Disparity Estimation

The light field, $\mathcal{P}(s,t,x,y)$ is split onto its EPIs, $I(x,s)$ and $I(y,t)$, for the horizontal and vertical directions respectively. For simplicity, the method is solely described for horizontal EPIs.

At this stage an ST-based disparity estimation, $d(x,s)$, is determined from Equation 2.32 for each point of every EPI. The reliability coefficient, $r(x,s)$ is also estimated from Equation 2.28. A first derivative Gaussian kernel is utilized to obtain the coefficients of the Structure Tensor, as it provides more accurate results than either the Sobel or Sharr convolution filters. In accordance with the study carried out by S. Wanner [2], an inner scale $\sigma_i = 0.7$ and an outer scale $\sigma_o = 1.5$ were utilized.

To simplify the algorithm, the disparity of each single-view of the light field is improved at a time. For that effect we keep only the lines $d(x) = d(x,u_0)$ and $r(x) = r(x,u_0)$, with u_0 equal to the u coordinate of the chosen view. In calculating our results, $u_0 = 5$ is utilized, as it refers to the center view of the light field images used in this work.

3.3.2 Edge Detection

This step involves the determination of edges on both the EPI and disparity map. To find edge structures on the EPI $I(x,s)$, two methods were utilized. The Canny Edge

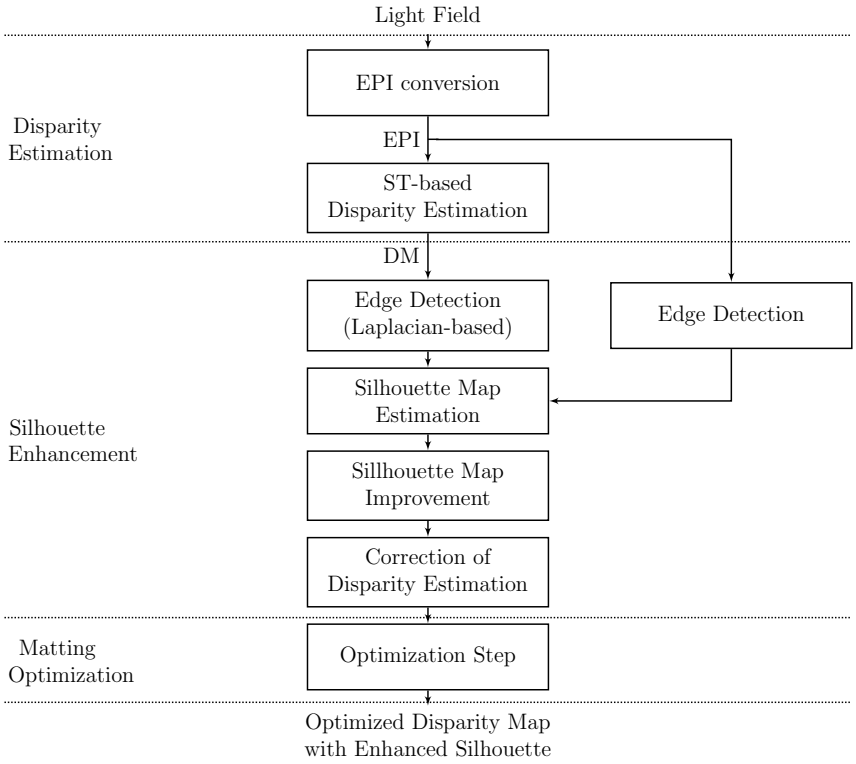


Figure 3.5: Algorithmic structure of of the Silhouette Enhancement Process.

Figure 3.6: Top: EPI. Middle: result of the Canny Edge Detector. Bottom: result of the Krawtchouk Polynomial-based edge detector.

Detector [32] and a method based in 2D Krawtchouk polynomials, first presented in [33]. Their effectiveness is compared in Chapter 4.

The Canny edge detector follows a simple algorithm. First, the image is smoothed with a Gaussian filter before calculating the gradient for each pixel. Much like in the implementation of the structure tensor previously described, many implementations of the algorithm combine the smoothing and gradient estimation steps into a single one by convolving the non-smoothed image with a first derivative of Gaussian kernel. Lastly, the algorithm applies a localization step. This can be sub-divided into non-maximal suppression, removing pixels that are not local maxima, keeping edges that are only one pixel wide, and hysteresis thresholding. Hysteresis thresholding is necessary because using a single threshold can omit weak edges and also allows false positives around strong edges if a low threshold is used. The canny edge detector uses two thresholds, a weak one and a strong one. Pixels above the weak threshold are only considered edges if connected to a pixel above the strong threshold.

The Krawtchouk polynomial method can calculate a matrix of gradient intensity of a discrete image while foregoing the smoothing step. Further reading of [34] and [35] are essential to understand the fundamentals of this method. Once gradient values are calculated. the edge detection methods are similar to the ones described for the Canny edge detector, utilizing threshold hysteresis and non-maximal suppression to obtain an edge map. As these edges are computed from the texture information of the light field, they will be henceforth designated image edges. Figure 3.6 shows a comparison of edge detection on an EPI using both methods. The edge map resulting from the Krawtchouk polynomial edge map (top) appears sparser than the one obtained from the Canny edge estimation (bottom) using similar thresholds, however when compared with the visible edges in the EPI (top), it appears to be accurately describing the position of the strongest edges.

A different approach is necessary to find edge structures in the disparity map $d(x, u)$ as it is an artificially generated image representing the disparity of each pixel of an EPI. A high gradient in a color image implies a rapid change of color, and therefore, an edge in the image. However, a high gradient in a disparity map merely identifies that the depth of the corresponding surface is changing at a rapid rate. This can occur not only at object boundaries, but also along a single object, if it is positioned at steep angle in relation to the image plane. In this manner, the Laplacian of a disparity map provides a more accurate edge map, as an abrupt change in the gradient of the disparity is much more indicative of an object transition.

Since the noise problem described in Section 3.2.2 results in false positives in the Laplacian calculation, a median filter is applied to the image, reducing this noise at a negligible cost to edge precision. The Laplacian of the median filtered disparity map is thresholded with the weighted mean of the disparity values of each EPI.

3.3.3 Silhouette Map Estimation

Not all image edges found through the various methods are relevant to the silhouette enhancement problem. A significant number of edges do not correspond to a transition between two objects at different disparities and therefore such edges do not contribute to silhouette enlargement. These edges are defined as texture edges, while those describing object boundaries are named silhouette edges.

In order to enhance our edge map, it is important to remove texture edges from the initial estimation, as they would further complicate the edge linking process, described in the following section, for no added value. Additionally, false edges caused by image noise are heavily reduced, generating a cleaner map.

The goal is thus to convert the edges found in every EPI $I(x, s)$, into a binary silhouette

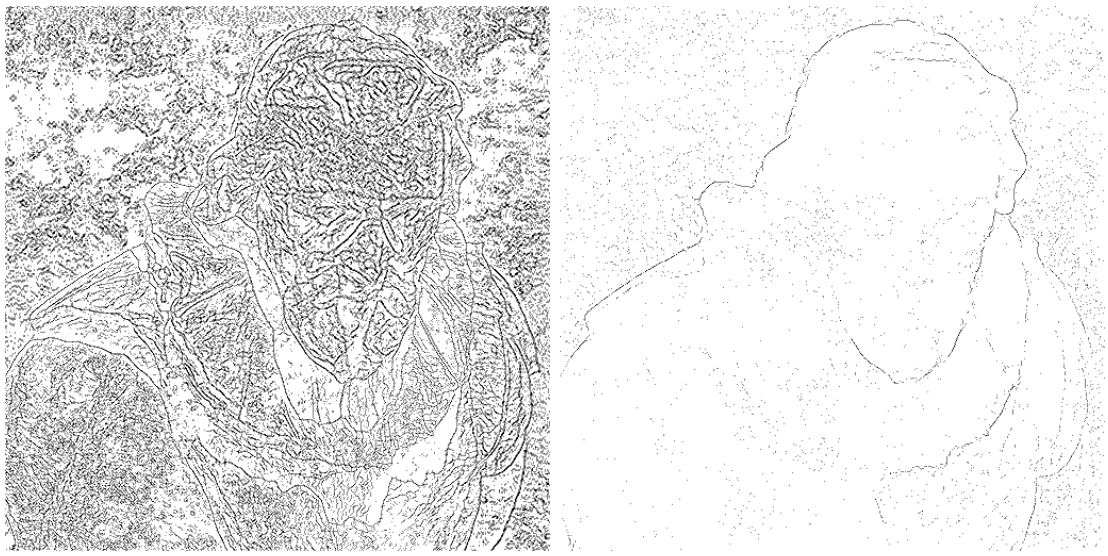


Figure 3.7: Comparison of the edge map obtained from the Canny Edge Detector on the left, and the Silhouette Map, obtained after excluding texture edges, on the right.

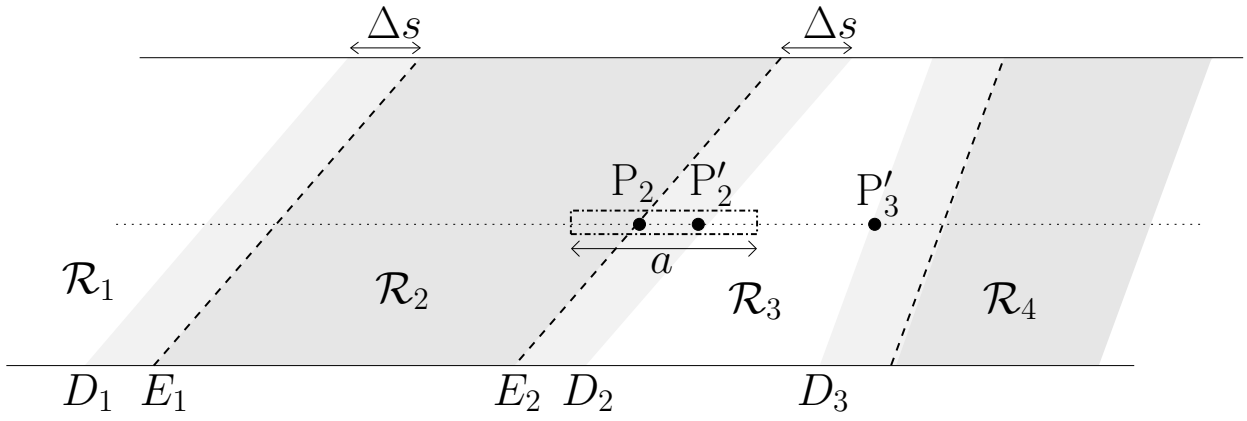


Figure 3.8: Silhouette enlargement: the original EPI lines (E_1; E_2) and the corresponding lines on the disparity map (D_1; D_2). Working out the region (width Δs) involves starting from an EPI edge (P_2) and search over window of size a to determine the position of P'_2. (Lourenço *et al.* [6])

map of the chosen view, $S(x, y)$, as shown in Figure 3.7, that contains only the outline of different objects. A change in both disparity and luminance is expected to be found at an object transition. Therefore, an edge is considered a silhouette edge only when corresponding image and disparity edges are found. Assuming that image edge points are roughly matching the real position of transitions, the algorithm for finding potential silhouette edges is simple.

As represented in Figure 3.8, starting from an edge point at the EPI representation (P_2), a search is performed on a window of width a , in order to find out the corresponding edge on the disparity map (P'_2). The dimension of the search region (a) is related to the window size (w) used when computing the structure tensor, as that is the theoretical maximum of silhouette enlargement, according to the processes described in Section 3.2.1. Additionally, it is known that the occluding region is always the one that increases in size from the silhouette enlargement effect. Therefore, the region beyond the disparity edge point should always have lower disparity (i.e. larger depth) than the region where the image edge point is found.

Thus, after identifying P_2 and P'_2, the search is extended beyond the window a , in order to find a new region ($\mathcal{R}_3$ in Figure 3.8) with a lower disparity than that of P'_2.

Summarizing, an image edge point P_2 is only identified as a silhouette edge, if it has a corresponding edge P'_2 for which a region $\mathcal{R}_3$ with lower disparity is found.

There are two special cases that deserve consideration. The first case where two different colored objects are located at adjacent positions, at an indistinguishable depth from the camera can be safely ignored. This strives from the fact that such object boundary, by definition, is not revealed in the disparity map, and thus any silhouette enlargement present would be undetectable. Such edges are equivalent to texture edges.

The second case occurs when an object transition has a well-defined disparity edge, but the luminance of both objects is similar to the point that no strong image edge appears in the corresponding EPI. While simply considering all disparity edges to be silhouette edges would be accurate in theory, in practice the method for detecting disparity edges gives too many false positives. Additionally, since disparity edges are displaced from their accurate position, without a precise image edge, it is impossible to estimate the real position of the edge point. In the final silhouette map, this translates to discontinuous lines, where certain portions of a silhouette edge are not correctly identified as such. This leads to silhouette artifacts in the final result. A method designed to improve the silhouette map by linking this discontinuous lines is described in the following section.

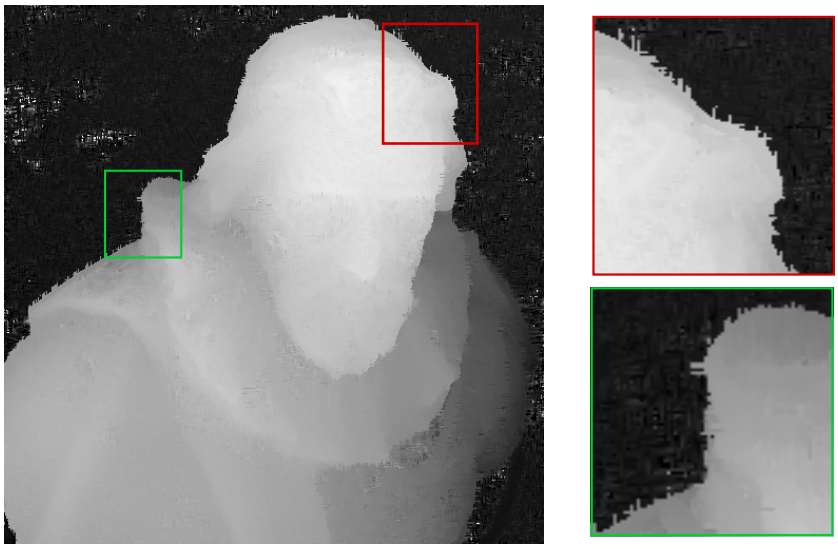


Figure 3.9: Silhouette Enhancement Artifacts on the *Cotton* dataset.

3.3.4 Silhouette Map Improvement

The silhouette map $S(x, y)$ estimated in Section 3.3.3 is a binary map describing the outline of the silhouette of all objects in a scene. As can be seen in Figure 3.7, the initial silhouette map contains discontinuous lines where a single continuous line would better describe the silhouette of a given object. Allowing these discontinuities can result in additional artifacts in the final disparity map, as shown in Figure 3.9. In order to fix this problem, an algorithm is proposed to connect discontinuous lines in $S(x, y)$, estimating the silhouette edge in regions where the edge detecting algorithm failed to detect edge points.

The first step consists in removing of all lines with length smaller than two pixels. This eliminates scattered noise with negligible impact on accurate silhouette edge estimations.

The coordinates of line endpoint in $S(x, y)$ are found through morphological operations and organized as a $2 \times N$ matrix, where $\frac{N}{2}$ equals the number of separate lines in $S(x, y)$. The aim is to associate each endpoint with a direction α in degrees. To achieve this, the coordinates of the first three points of the line associated with each endpoint are found by iteratively searching for $S(x, y) = 1$ in all 8-connected neighbors of the endpoint.

The difference between the coordinates of an endpoint (x_0, y_0) with the next point in the line (x_1, y_1) are stored in two vectors $\boldsymbol{x}^{dif}$ and $\boldsymbol{y}^{dif}$ such that $x_i^{dif} = (x_i - x_{i+1})$ and $y_i^{dif} = (y_i - y_{i+1})$. This is repeated for all $i < 3$, which is equivalent to considering only the four points of the line closest to the chosen endpoint. α is then calculated as:

$$\alpha = \arctan\left(\frac{\bar{\boldsymbol{y}}^{dif}}{\bar{\boldsymbol{x}}^{dif}}\right) \tag{3.1}$$

Figure 3.10: Comparison of the Silhouette map before and after the silhouette improvement step.

where $\bar{\boldsymbol{x}}^{dif}$ and $\bar{\boldsymbol{y}}^{dif}$ represent the expected values of the vectors $\boldsymbol{x}^{dif}$ and $\boldsymbol{y}^{dif}$. α represents the angle, in degrees, associated with the starting endpoint.

An exhaustive search for every combination of endpoints would heavily increase the complexity of the algorithm. As an alternative, the Euclidean distances from each endpoint to all others are pre-computed and stored in matrix $E_{N\times N}$.

The algorithm iteratively searches for the lowest element of matrix $\boldsymbol{E}$, which provides the closest pair of endpoints. A maximum distance threshold is defined as the stop condition. Originally connected endpoints are attributed a distance larger than this value, excluding them from our search.

The viability of a connection between each pair of endpoints is tested based on the difference between the direction α associated with one endpoint, and the direction that is symmetric to the one associated with the other endpoint, calculated as $\alpha' - 180$, where α' represents the direction associated with the second endpoint. If this difference is above the predefined threshold, the pair is discarded, and the respective element in matrix $\boldsymbol{E}$ is increased to above the maximum distance threshold. If the difference is below the threshold, that pair is considered connected and the rows and columns of $\boldsymbol{E}$ representing these edges are increased to above the maximum distance.

The Brasenham algorithm is used to link edges that are considered connected with a straight line in $S(x, y)$. Figure 3.10 shows the improvements to the silhouette map in this step. The improved map on the left shows significantly less false positives, and less discontinuities in the lines of the map outlining the silhouette. A more complete overview of the results is given in Chapter 4.

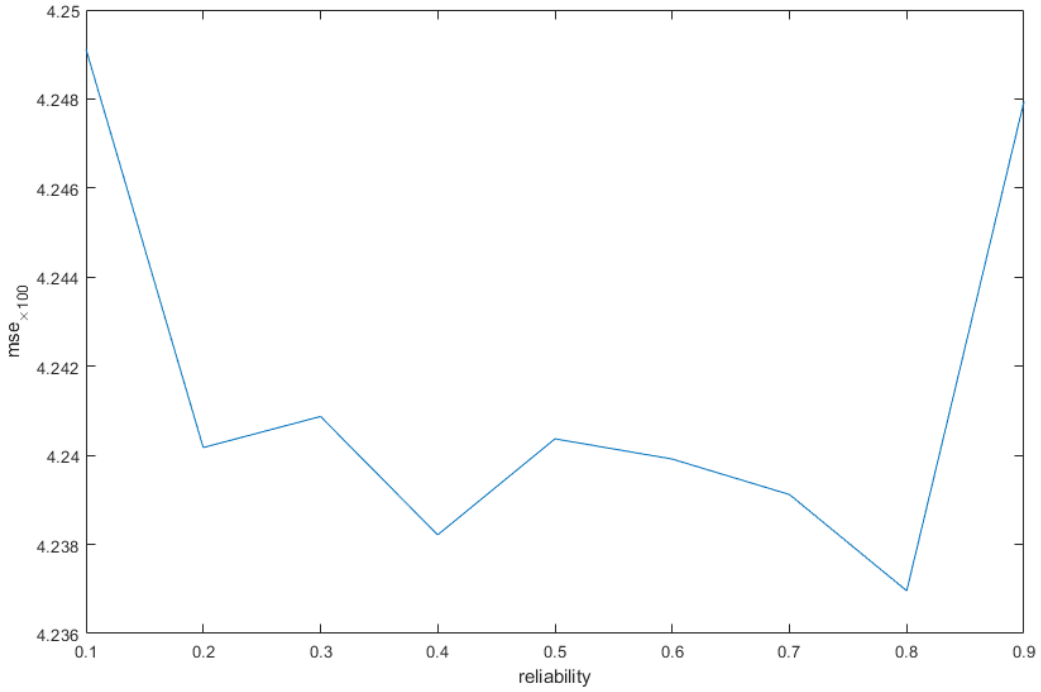


Figure 3.11: Average MSE for four different datasets using different reliability thresholds in the value replacement process.

3.3.5 Correction of Disparity Estimation

With a better Silhouette Map available, the process for matching image edge points with disparity edge points outlined in Section 3.3.3 is repeated, using the silhouette edge points from the improved silhouette edge map as image edge points.

Once again using Figure 3.8 as a reference, for each silhouette edge point P_2 a disparity edge point P'_2 is found so that region $\mathcal{R}_3$ has lower disparity than region $\mathcal{R}_2$. The region a between these two points, is thus the region affected by the silhouette enlargement effect.

The next step is to correct the disparity values by substituting them by the mean disparity of the occluded region (R_3), which requires determination of its far-end edge (P'_3 on D_3). To reduce the influence of noise, only points with a high reliability indicator (r) are considered to calculate this mean value. Reliability values from $r = 0.4$ to $r = 0.9$ were tested for four different data-set, $r = 0.8$ was chosen as the one that provided the best results in terms of mean square error, but small variations of this value do not have critical impact on the performance.The results of these tests are shown in the graph of Figure 3.11. The reliability r of the substituted values is also changed to equal the average of the reliability of the disparity values used for the substitution.

Using only the mean disparity value to substitute the disparity values of the erroneous

region is often a poor match in slanted surfaces, yet it proves more beneficial than attempting to predict the orientation of the corresponding surface, as local noise prior to the Matting optimization step is too high.

3.3.6 Analysis of vertical and horizontal disparity estimations

Following the above steps for horizontal and vertical EPIs, two improved disparity maps $d_{xs}(x, y)$ and $d_{yt}(x, y)$ are obtained, as well as two reliability maps $r_{xs}(x, y)$ and $r_{yt}(x, y)$. To build a final disparity map $d(x, y)$, for each point (x, y) the disparity estimation associated with a higher degree of reliability is chosen.

3.3.7 Optimization Step

At points where the algorithm fails to find accurate edges, the disparity map exhibits jagged edges, as shown in Figure 3.9. To limit the impact of these artifacts, and to reduce local noise, an optimization step is applied by considering the Laplacian-based image matting correction described in Section 2.2.4.

Mean shift segmentation is used to separate the image into different segments based on color similarity, which are optimized separately. The goal is to keep different objects in different segments in order to avoid over-smoothing of edges, as the optimization algorithm tends to smoothen steep transitions, which is not adequate for this purpose. However, as the mean shift segmentation algorithm is not totally accurate, over-smoothing occurs in a great number of situations.

Over-smoothing occurs whenever one of the segments is wrongly calculated, and includes a silhouette edge as well as a portion of a different image object or background region. The disparity of the high reliability foreground object is then associated with the lower reliability values common in occlusion regions and, in that manner, the transition between these values is smoothed over by the optimization algorithm.

In section 2.2.4, the Laplacian Matrix is derived, but no meaning was defined for its elements other than their mathematical derivation. However, the Matting Laplacian Matrix can be understood as an application of the Graph Laplacian to images. In this manner, the principal diagonal of L provides a measure of how strongly connected each pixel is to the rest of the image, while other elements represent the strength of specific connection, or the affinity, between two vertices.

To minimize the This of over-smoothing, improved silhouette edge maps were used to perform an edge-aware optimization where any pixels in a window containing an edge were attributed zero affinity in the Laplacian matrix.

While this proved to achieve some success at minimizing over-smoothing, even com-

pletely removing it, in some cases, the use of edges did not consistently eliminate over-smoothing in the entire image as the improved silhouette maps, while accurate, are not entirely perfect, and a single strong affinity value between a pixel in the foreground and a pixel in the background areas propagates this affinity across the entire segment, which translates to the over-smoothing of object transitions in the optimized disparity map.

3.4 Plane Noise Reduction

As addressed in Section 3.2.2, structure tensor-based disparity maps do not enforce local smoothness. Therefore, the error in each adjacent sample is disassociated, giving an impression of noise. It is thus clear that estimating the surface orientation based on the difference between adjacent pixels leads to inaccurate estimates, as the difference related to noise can be significant in comparison with the difference between the true estimates. Nevertheless, the noise amplitude is generally low. Therefore, if the distance between two points in the same region is much larger than the noise level, an accurate estimation of the average surface orientation in the region between the two points can be found. It is therefore possible to estimate the average orientation of a pixel from a structure tensor based estimation. If such region describes a single planar surface, then the average surface orientation is equal to the local orientation of each point in the surface. Thus, it is possible to estimate the local orientation of planes from disparity maps obtained from the structure tensor.

Estimating planes in the 3D scene requires the conversion of discrete image coordinates into scene coordinates. This conversion depends on lens parameters and the depth at each point. This depth depends on the same parallax effect that permits the detection of disparity using Light Field cameras. The distance of a pixel in the horizontal direction is smaller for objects closer to the camera, and longer for objects further from the camera.

This depth dependency is a base-line problem of the plane estimation algorithm presented in this section. While the algorithm accurately estimates planes in the 3D scene, re-projecting such planes onto the screen requires the use of the non-smooth disparity map, as the operation once again depends on the parallax effect previously described, providing results that maintain some level of noise in areas close to the camera where the generally small projection error is magnified.

Having converted the disparity map into a scene map $s(x, y, z)$, a seed-based approach is followed, in line with the algorithms described in [36]. Seeds are fit to planes using a least squares approach, and then grown to the size of the plane they represent in the disparity map. Several steps are taken to avoid fitting planes to curved surfaces. Finally, a segmentation map of image planes is created, and silhouette corrected areas are assigned

to the relevant planes before replacing the original disparity values with the results of the equation for their respective planes.

3.4.1 Initial Seeds

Initial seeds represent small patches of the scene map. In this algorithm, every possible seed of the same size is tested by sectioning the disparity map into all possible overlapping square patches of equal size, and utilizing the corresponding pixels of the scene map s.

Seed growth based algorithms like the one described in [36], assume locally smooth disparity maps as their input, which enables them to fit a plane on 2×2 initial seeds. However, as mentioned in Section 3.2.2, the initial estimation of the structure tensor provides noisy disparity maps. Therefore, using only four adjacent samples of a depth map obtained from the structure tensor provides highly inaccurate plane fitting. Nevertheless, to be able to detect small plane structures accurately it is important to use a small seed size so that the initial seeds can be contained in every image plane. The initial seed size is thus an adjustable parameter of this algorithm, that can be changed according to the the characteristics of a scene. Throughout our tests an initial seed size of twenty proved to be the most reasonable for every scene.

The goal is to rank all seeds by how closely they describe a plane. To do so, a plane is fitted to the samples in each seed using a least squares approach. The distance from each point in the seed to the estimated plane is calculated. The variance of these distances is used to rank the fit of the plane. It could seem that directly using the average of the differences would provide a better ranking of the fit but that is not necessarily the case. This is justified by considering two examples. In the case where all points are at the same large distance from the fitted plane, it is clear that these points still describe a plane, parallel to the plane they fit. In opposition, the case where all points are at shorter but different distances from the plane that best fits them would have a smaller average difference, yet those points would be a worse description of a plane. Since the actual estimation of the plane is not relevant, as the fit will change as the seed grows, it is more important to guarantee that the starting point of the growth algorithm is a planar seed.

3.4.2 Seed Growth

The goal is to grow some of the seeds, so that all planes in the scene are accurately described by a single seed indicating the points that represent the planar surface and a plane $ax + by + c = z$ that best fits those points. Starting with the seed that best fits a plane, the points adjacent to the seed in every direction are added. The distance from these new points to the previously fitted plane is calculated and only those below a certain

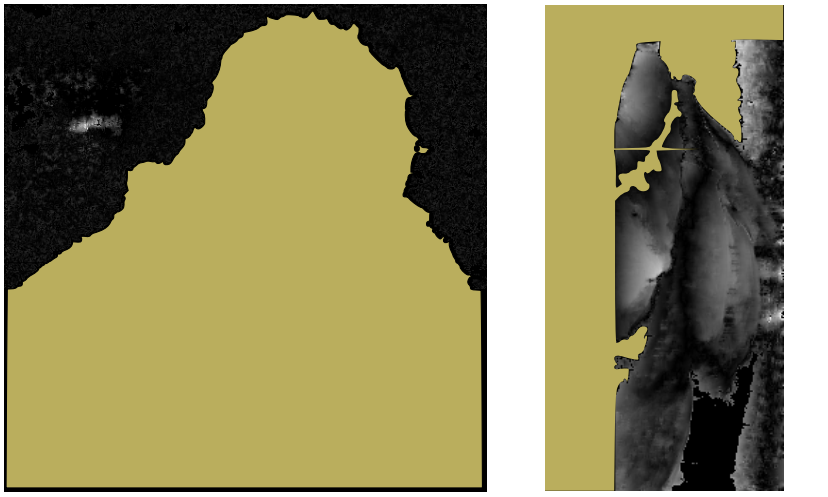


Figure 3.12: The distance between the original estimation of the scene and an estimated plane. White is the largest absolute distance. Points outside the estimated plane are in color.

threshold are kept. In this manner, a slightly larger seed is obtained and the fitted plane is adjusted.

This process is repeated iteratively until one direction of expansion (i.e. up, down, left or right) finds no further points below the threshold or reaches an image boundary, at which point growth is stopped along this direction. This guarantees that the seeds remain contiguous. When all four directions are halted, the seed is considered fully grown. When a fully grown seed is computed, a lot of the initial seeds are now overlapping this seed, and are thus redundant. Cleaning up these seeds is important to keep run-time manageable.

This process is repeated for every seed until there are no initial seeds left above a certain variance threshold. To avoid confusion in the terminology, fully grown seeds are designated regions. Unfortunately, not all estimated regions are best described by planes.

3.4.3 Distinguishing Planar from Non-Planar Regions

Regions are not always best described by planes. Smaller patches of slightly curved surfaces can also provide very good fits for a plane. Some other regions are also best described as the region of intersection of a plane with a scenic surface. Allowing these would cause curved surfaces in our disparity map to become stratified, and decrease the quality of the disparity map in those areas. Additionally, regions described by planar but noisy areas of a scene can erroneously prove poor fits to a plane using regular metrics.

The high variance in planar regions due to local noise makes metrics like the variance insufficient to differentiate between the two cases listed above. To the human eye, the

Figure 3.13: Map of the different planar regions estimated for the dataset *Dino*. Black represents the regions of the image for which no planar region was found.

distance from plane images look fairly distinguishable. Nevertheless, common metrics tend to fail as can be seen in Figure 3.12 [2]. The difference image on the left refers to a region of the image that best describes a plane. The image on the right describes to a heavily non-planar surface. However, due to noise, the average of the distances in the non-planar region is lower.

Noting that in planar regions, local noise was responsible for the high average distance values, while in non-planar regions, the distance to the plane increased continuously, TV-L1 denoising [26] of the difference images was utilized. This made the average difference of planar regions distinguishable from non-planar regions. The variance of the initial depth map, weighed to reflect the usual values for the average difference of planar regions, is used as a threshold.

3.4.4 Improving the disparity map

To change the initial disparity values into an improved version based on the plane estimations, a region map is built. The region map provides, for each pixel of the original disparity map, the index of a corresponding estimated region. The region map is zero for pixels where no corresponding planar region was estimated. Figure 3.13 shows the map of the different planar regions found for the dataset *Dino*.

In order to build this map, whenever a plane-described region finishes growing, its index is inserted into all corresponding pixels positions in the plane map covering that

[2] This figure is best observed in color.

region. In the case of overlapping regions, the seed that describes the closest plane to the original estimation is kept on a pixel by pixel basis.

Finally, by solving the plane equation $z = ax + by + c$ for every point of each region, the distance values of our initial estimation are updated with the value of the estimated plane. The map with the z value for each point is thus an improved depth map. In order to better compare results with other algorithms and benchmarks, the depth results are converted back to disparity values.

Chapter 4

Results

In this chapter, the results of the algorithms presented in Chapter 3 are presented, compared with other state of the art methods, and discussed. The results are analyzed according to metrics discussed in [37], in particular the mean square error (MSE), badpix and median angle error (MAE). Badpix measures the percentage of pixels with disparity difference to the ground truth above 0.07. The MAE measures the median angle error, in degrees, of the surface normal vectors calculated from a given disparity map. The algorithms were tested using the Heidelberg Light Field dataset [38].

In order to better compare the results of the proposed methods with the state of the art, it was necessary to combine all the improvements into a fully functional framework. This framework can be divided into four steps: disparity map estimation, silhouette enhancement, matting based optimization and plane noise reduction are applied sequentially. This is in accordance with the diagram for the silhouette enhancement method shown in Figure 3.5.

In the disparity estimation step, the structure tensor is used as described in Chapter 3. In the silhouette enhancement step, the Krawtchouk polynomials based edge detector is used. A question arises in the ordering of the matting based optimization and plane noise reduction steps. While applying the optimization step later would guarantee any possible artifacts from plane noise reduction would be reduced, applying it earlier provides smoother planes, enabling a better plane estimation in the plane noise reduction step. The second alternative was chosen. Figure 4.1 shows a diagram of the framework used.

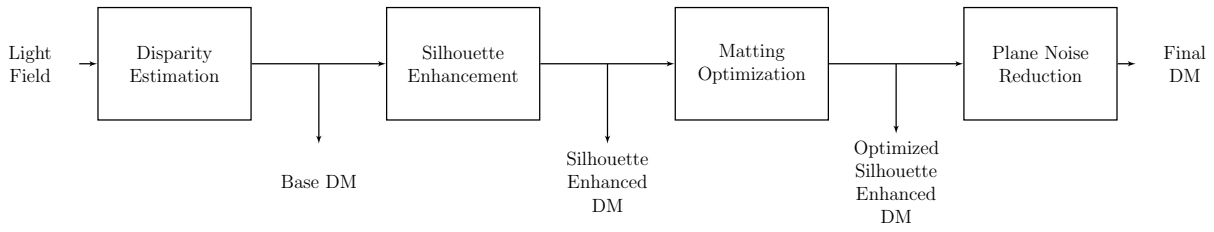


Figure 4.1: Diagram of the full framework used for comparison with the state-of-the-art.

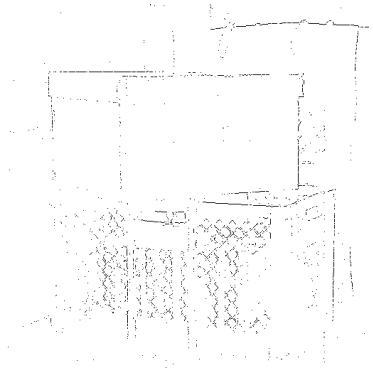
4.1 Silhouette Enhancement

The selection of edge detecting algorithm is very relevant to the results of the silhouette enhancement method. The two algorithms discussed in Section 3.3.2, one based on Krawtchouk polynomials, henceforth referred to as the Kraw method and the Canny edge detector, were tested with and without the silhouette map improvement step, described in Section 3.3.4. In this section, the use of these two edge detecting methods is compared, and the results of the silhouette map improvement process are analyzed. Finally, the overall results of the silhouette enhancement framework are compared with the initial structure tensor-based estimation.

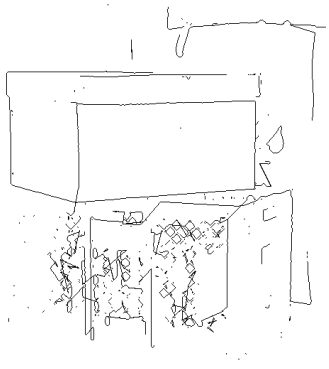
Figure 4.2 compares the silhouette maps of four different light field images, obtained from the silhouette map estimation process (see Section 3.3.3), with silhouette maps that underwent the silhouette map improvement process described in Section 3.3.4. The Kraw method was utilized for edge detection. Figure 4.3 shows the same comparison using the Canny edge detector.

For both methods, the improved images (on the right) show more continuous lines that outline a more complete silhouette of the objects in the images. For the Kraw method, this can be seen, for example, in the *Cotton* dataset. In the left portion of Figure 4.2(c), one can see the lines that describe the silhouette of the shoulder are pointy and incomplete, while in figure 4.2(d), the same line is strong and continuous. The same can be seen for the Canny edge detector, taking the *Sideboard* dataset as an example, in Figure 4.3(g), one can see discontinuities in the leftmost vertical line, outlining the leftmost edge of the cabinet. In Figure 4.3(h), the same feature is described by a continuous vertical line. It is also visible that the leftmost images (the ones before the improvement step), are much less noisy (less false positive silhouette edges) when using the Kraw method rather than the Canny edge detector. The use of the Canny edge detector, before improvement, also tends to produce silhouette maps with a larger number of small discontinuities along bigger lines, which the silhouette improvement algorithm is designed to correct.

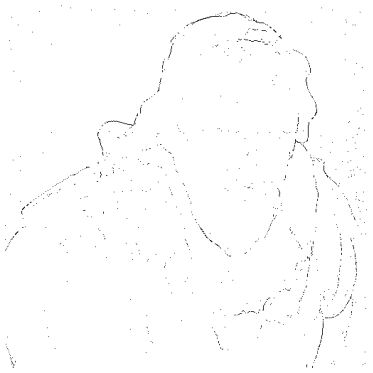
Table 4.1, 4.2, 4.3 and 4.4 respectively show MSE×100, MSE×100 in border regions, badpix, and badpix in border regions results for the disparity maps obtained from four



(a) Boxes Before Improvement



(b) Boxes After Improvement

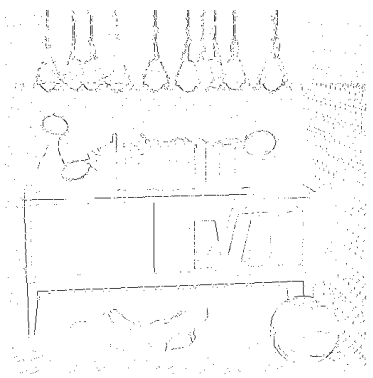


(c) Cotton Before Improvement

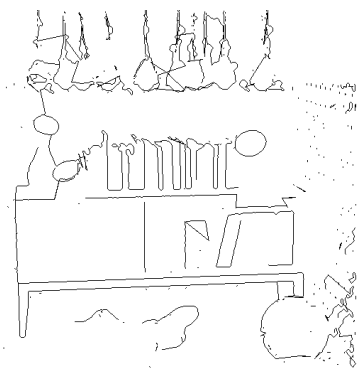
(d) Cotton After Improvement

(e) Dino Before Improvement

(f) Dino After Improvement

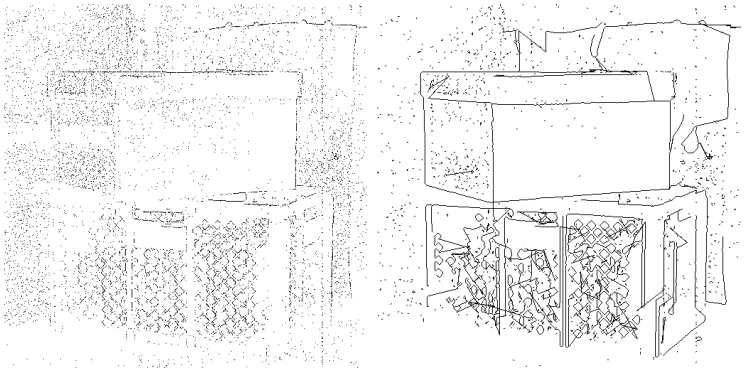


(g) Sideboard Before Improvement

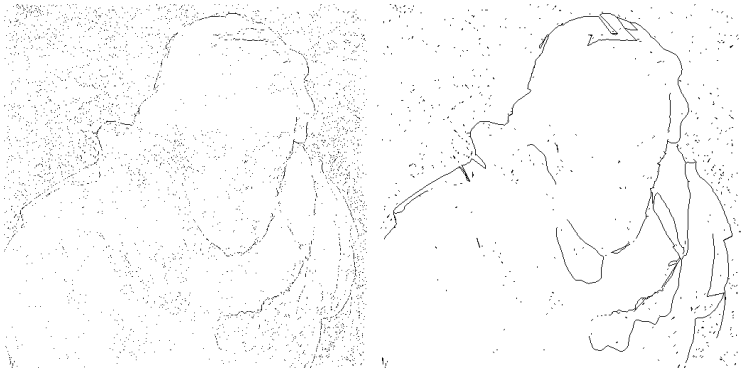


(h) Sideboard After Improvement

Figure 4.2: Comparison of Silhouette Maps obtained from the Kraw Edge Detector before and after Silhouette Improvement

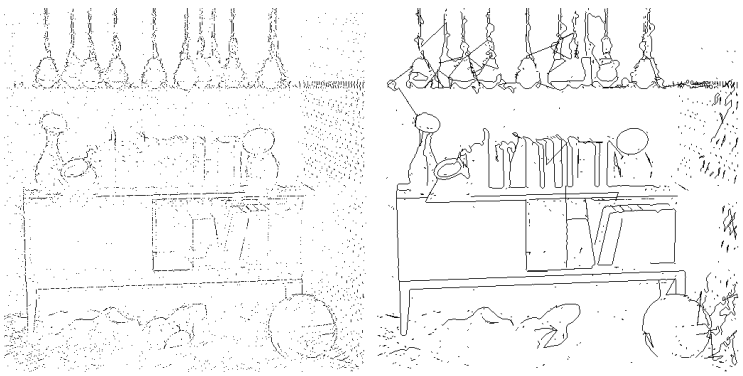


(a) Boxes Before Improvement (b) Boxes After Improvement



(c) Cotton Before Improvement (d) Cotton After Improvement

(e) Dino Before Improvement (f) Dino After Improvement



(g) Sideboard Before Improvement (h) Sideboard After Improvement

Figure 4.3: Comparison of Silhouette Maps obtained from the Canny edge detector before and after Silhouette Improvement

Table 4.1: MSE ×100 comparison for four different images

	Initial Estimation	Kraw	Canny	Kraw - SI	Canny - SI
Boxes	15.943	12.699	12.705	12.665	12.603
Cotton	4.166	1.616	1.921	1.603	1.550
Dino	1.611	0.799	0.927	0.782	0.863
Sideboard	2.924	1.8912	2.0319	1.898	1.863

different light field images, results are presented for the base structure tensor estimation, and silhouette enhanced disparity maps using the Canny edge detector and the Kraw Method, with and without silhouette map improvement(SI in the table).

Comparing the use of the Kraw method with the use of the Canny edge detector, without silhouette map improvement, the Kraw method proves superior in all metrics for the *Cotton*, *Dino* and *Sideboard* images, while being slightly worse than the Canny edge detector for the *Boxes* image in terms of MSE ×100 and badpix close to occlusion regions. In terms of the overall effectiveness of the silhouette map improvement step (described in Section 3.3.4), the MSE and badpix close to borders show some minor improvements, with the MSE in border regions when using the Canny edge detector rivaling the values of the superior Kraw method. For example, the badpix in border regions for the *Cotton* image improved from 33.85% to 31.92%. However, the Kraw method obtained little benefit overall and no benefit or even detriments in border regions, as can be seen by the results for the badpix in border regions for the *Dino* image, that went from a badpix of 29.45% without silhouette map improvement, to 30.03%. Nevertheless, as the goal of the silhouette map improvement step is to fix artifacts that while having a big visual impact, are small in size, it is to be expected that the numeric improvements would never be outstanding.

As for the results of the overall improvements to the entire silhouette enhancement framework, described in Section 3.3 and outlined in Figure 3.5. From the tables one can see a consistent improvement from the initial estimation to the results obtained from our framework for all datasets and using either edge detection algorithm. For example, the badpix in border regions for the disparity map of the *Cotton* image drops 12.68% from the initial disparity estimation to the silhouette enhanced disparity estimation, using the Canny edge detection method with silhouette map improvement.

The improvements made are hard to appreciate when observing the disparity maps directly. Therefore, Figure 4.4[1] compares the absolute difference between a disparity map with enhanced silhouette (the Kraw method with silhouette improvement is used) and the ground truth disparity. The images are in a scale of dark blue to red, where dark blue indicates no difference to the ground truth, and dark red indicates a difference in

[1]The Figure must be observed in color.

Table 4.2: MSE ×100 in border regions for four different images

	Initial Estimation	Kraw	Canny	Kraw - SI	Canny - SI
Boxes	0.522	0.492	0.490	0.492	0.491
Cotton	0.446	0.319	0.339	0.320	0.319
Dino	0.364	0.294	0.302	0.298	0.300
Sideboard	0.341	0.284	0.288	0.286	0.284

Table 4.3: Badpix 0.07 comparison for four different images

	Initial Estimation	Kraw	Canny	Kraw - SI	Canny - SI
Boxes	41.179%	35.64%	35.42%	35.58%	35.53%
Cotton	21.880%	16.42%	16.58%	16.32%	16.29%
Dino	13.612%	10.21%	10.79%	10.07%	10.43%
Sideboard	17.972%	15.07%	15.47%	15.00%	15.18%

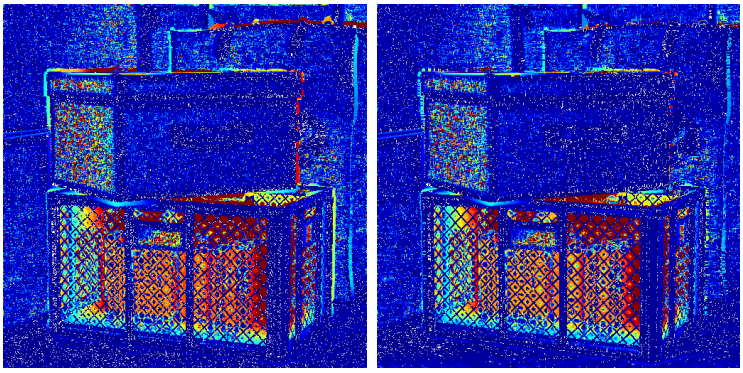
disparity larger than 2.

In the difference images obtained from the base estimations (left), one can see a high difference outlining the silhouette of objects in the image. This indicates a large error around objects due to the silhouette enlargement problem described in Section 3.2.1. In the difference images of the silhouette enhanced disparity maps, one can see either a reduction in the amplitude of the error, or a complete elimination of the silhouette enlargement. This is visible, for example for the *Dino* image, on the left side of the Figure 4.4(e) one can see a strong red outline of the silhouette of the cabinet, this error is completely corrected in 4.4(f).

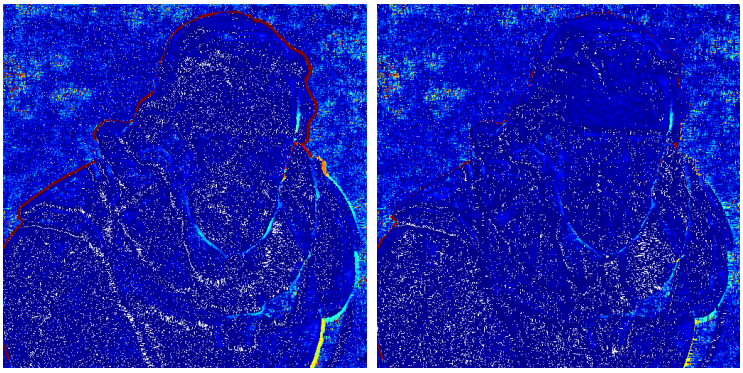
While the Kraw method proves superior if the silhouette map improvement is not applied, the Canny edge detector method receives greater benefits from silhouette map improvement. Overall the silhouette enhancement framework produces great results at limiting the effects of silhouette enlargement using either the Kraw method or the Canny edge detector.

Table 4.4: Badpix 0.07 in border regions for four different images

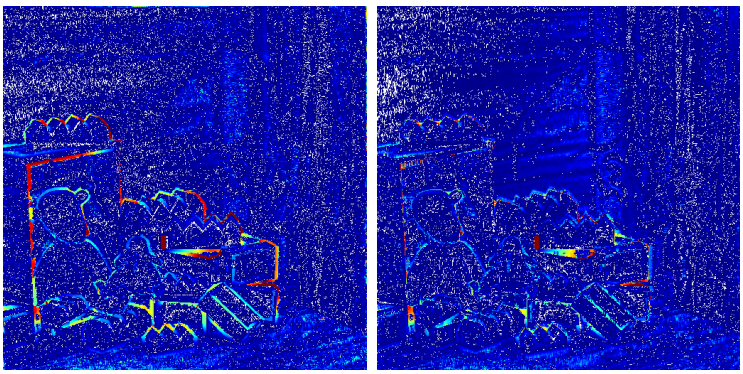
	Initial Estimation	Kraw	Canny	Kraw - SI	Canny - SI
Boxes	52.15%	49.21%	49.00%	49.18%	49.12%
Cotton	44.60%	31.90%	33.85%	31.98%	31.92%
Dino	36.39%	29.45%	30.28%	29.80%	30.03%
Sideboard	34.11%	28.47%	28.84%	28.58%	28.36%



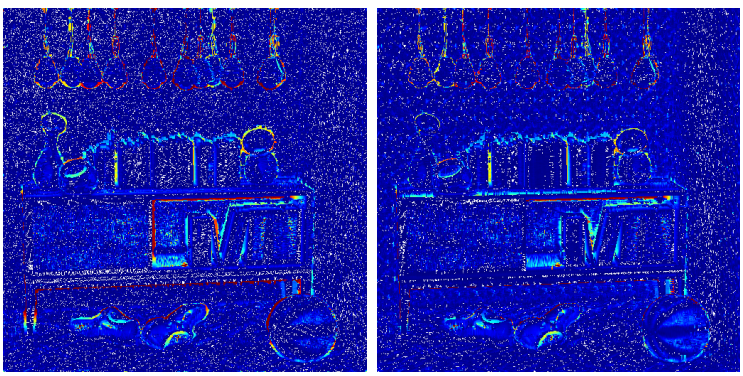
(a) Boxes (b) Boxes



(c) Cotton (d) Cotton



(e) Dino (f) Dino



(g) Sideboard (h) Sideboard

Figure 4.4: Comparison of the absolute differences between the ground truth and the base structure tensor estimation (left) and the silhouette enhanced estimation (right). Red indicates a higher difference.

4.2 Matting Optimization

This section discusses the results of the matting optimization step, and compares the simple method first proposed in [13] with the edge-aware method described in 3.3.7. To obtain these results, the Kraw method with silhouette improvement was used.

As discussed in Section 3.3.7, the goal of the edge-aware method was to reduce the over-smoothing of edges that occurs when using the simple method. This is illustrated in Figure 4.5. The figure shows two plots of the disparity in an horizontal line of a disparity map (*cotton* image). The top plot shows the disparity obtained using the simple method, the bottom plot shows the disparity of the same horizontal line using the edge-aware method.

It is visible that the top plot converges smoothly from around -1.2 disparity to 0.2 disparity, the transition occurring in the space of six pixels. Meanwhile, the bottom plot shows a steep transition from -1.2 to 0.2 in the space of a single pixel. For this case, the edge-aware method completely removes the over-smoothing problem that occurs when using the simple method. However, as mentioned in 3.3.7, while in some cases, as is illustrated in the figure, the edge-aware method reduces over-smoothing, there are other cases where there's no difference between the edge-aware method and the simple method.

In Table 4.5 one can see the results of the edge-aware algorithm, when compared with standard matting based optimization, when applied to a Silhouette Enhanced disparity map, using the Kraw Edge Detection algorithm with edge improvement.

For all images the MSE for the simple method is smaller than the MSE for the edge-aware method, yet the badpix and specially the badpix near border regions are smaller for all images when using the edge-aware method. This is to be expected as the edge-aware method essentially prevents the optimization of areas near edges, so while previously correct values are not being distorted, errors related to noise in the area near the edges of the figure will remain unchanged.

The edge-aware method reveals a clear trade-off between sharp accurate edges and

Table 4.5: Results of the matting based optimization, both simple (SM) and edge-aware (EAM)

	MSE×100		Badpix 0.7		Badpix Borders	
	SM	EAM	SM	EAM	SM	EAM
Boxes	8.345	8.421	27.25%	26.78%	54.60%	53.44%
Cotton	0.524	0.648	6.89%	6.36%	39.50%	32.88%
Dino	0.549	0.605	7.36%	7.08%	31.06%	29.23%
Sideboard	1.245	1.412	12.91%	12.14%	33.53%	30.66%

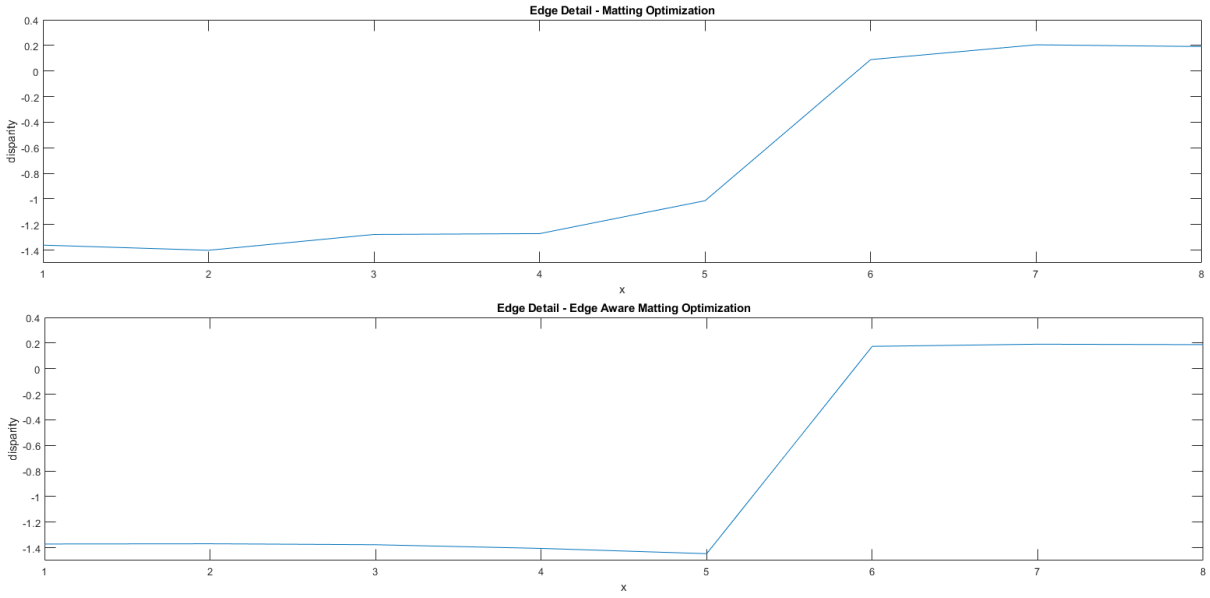


Figure 4.5: Comparison between a line of an optimized disparity map using the simple method of matting optimization and the edge-aware method of matting optimization.

noise removal near edges. Therefore, the choice between optimization algorithm should depend mostly on the context of utilization. The edge-aware method should be utilized in contexts highly dependent on edge precision, but the simple method could be more useful if sensitivity to noise in the disparity map is a problem.

4.3 Plane Noise Reduction

The main objective of the plane noise reduction algorithm is to reduce the amount of local noise in planar regions of the disparity map. In this section this is analyzed by measuring the median angle error (MAE) of the surface normals in planar regions of disparity maps and by visually comparing the normal maps (see the definition of normal map in Section 3.2.2) obtained with and without the plane noise reduction improvement step.

Table 4.6 compares the MAE results for the disparity map before (Matting Opt) and after (Plane-NR) plane noise estimation. Using as a starting point the disparity maps obtained after silhouette enhancement using the Kraw method with silhouette map improvement and the simple method of matting. The plane noise reduction method is tested for four different light field images.

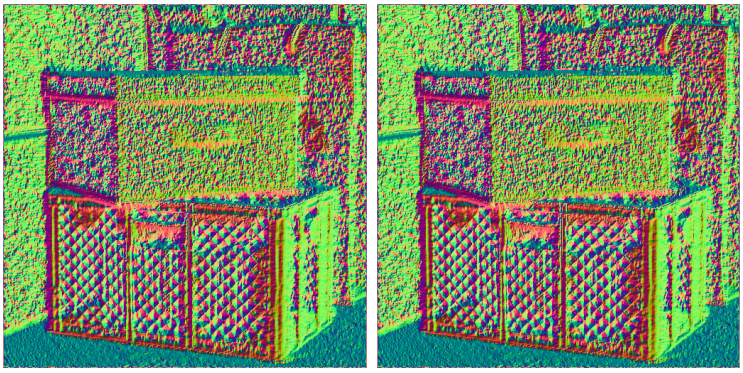
The MAE of planar regions were greatly improved without interfering with the non-planar regions. Using the *Cotton* image as an example, we can see a reduction of the MAE in planar regions from 76.23° to 1.09°, while keeping the mean angle error in non planar regions nearly identical. The algorithm does not function for the *boxes* image, as

Table 4.6: Median Angle Error in planar and non planar regions for disparity maps having only matting optimization and having an additional plane noise reduction (Plane-NR) step.

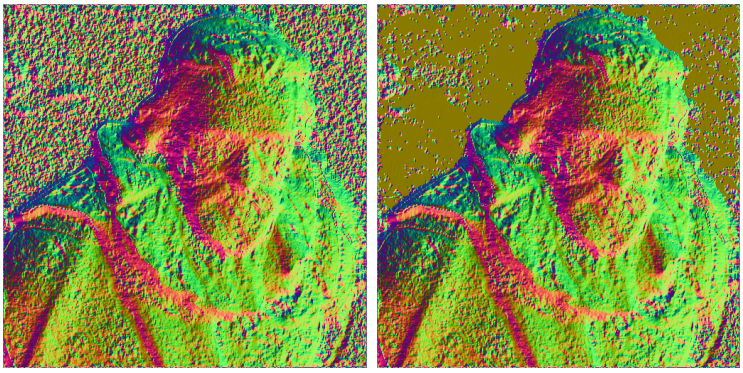
	MAE-Planar		MAE-Non Planar	
	Matting Opt.	Plane-NR	Matting Opt.	Plane-NR
Boxes	27.54	27.54	66.86	66.86
Cotton	76.23	1.09	27.91	27.92
Dino	17.81	7.80	22.33	22.33
Sideboard	23.75	5.30	49.19	45.89

the planar regions are too noisy, this translates to results that are exactly equal to the ones where plane noise reduction is skipped, with no additional errors being introduced.

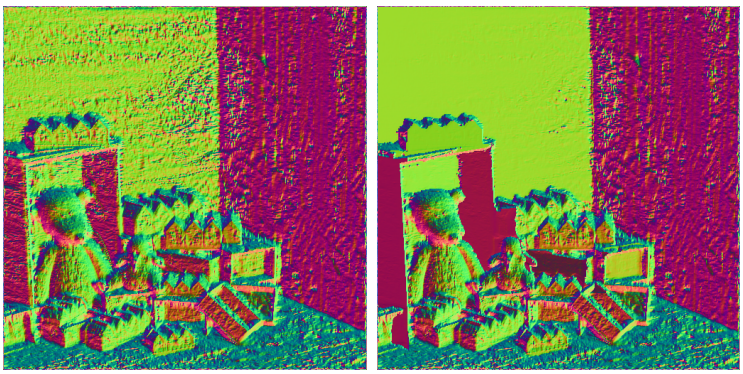
Figure 4.6 shows the normal maps of four datasets obtained from disparity maps before and after the plane noise reduction step. One can see the great improvement, for example, in the background region of the Cotton dataset, in Figure 4.6(c) the background is noisy, while in Figure 4.6(d) the background is an homogenous green-brown color, as would be expected from a planar region.



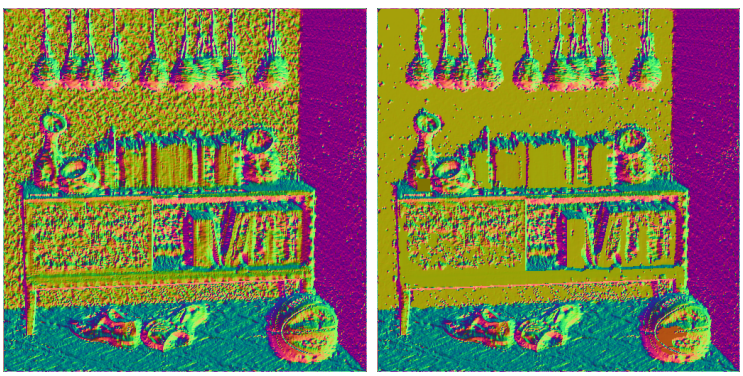
(a) Boxes Before PNR (b) Boxes After PNR



(c) Cotton Before PNR (d) Cotton After PNR



(e) Dino Before PNR (f) Dino After PNR



(g) Sideboard Before PNR (h) Sideboard After PNR

Figure 4.6: Comparison of the normal maps obtained from optimized disparity maps before and after plane noise reduction (PNR).

4.4 Comparison with State of the Art

In this section the output of the full framework, as shown in Figure 4.1, is compared with other methods published in the literature. Table 4.7 compares the proposed method in terms of MSE with the methods proposed in [19] (SPO) , [18] (OFSY_330/DNR), [20] (RM3DE) and [15] (EPI2). The different frameworks are compared in terms of MSE, MAE and badpix for four different light field images. These results can be seen in Tables 4.7, 4.8, 4.9, respectively.

The proposed method comes ahead of the other structure tensor-based framework, by Wanner *et al.* [15] in all metrics, proving to be a valuable improvement to the structure tensor framework. In terms of MSE, this work achieves competitive results in relation with the multi-resolution approach by Neri *et al.* [20] and the normal-map regularization based method by Willem *et al.* [18], while trailing the spinning parallelogram operator method [19]. In terms of MAE, the proposed framework proves superior to all methods but the one by Willem *et al.*, as this method applies a global optimization with constraints on the regularity of the produced normal maps, even so, the results of the proposed method still achieves comparable results in all but the *Boxes* image.

Due to the over smoothing of edges caused by the matting optimization, our framework lags behind in terms of badpix, where it is only superior to the structure tensor based method by Wanner *et al.*. Potential solutions to this issue are presented in the Chapter 5 section about future work.

Table 4.7: MSE $\times 100$ comparison with state of the art methods

	Proposed	SPO	OFSY_330/DNR	RM3DE	EPI2
Boxes	8.345	7.625	9.107	9.561	10.928
Cotton	0.507	0.341	1.313	2.653	4.318
Dino	0.575	0.36	0.31	0.782	2.076
Sideboard	1.242	1.071	1.024	2.478	4.651

Table 4.8: MAE comparison with state of the art methods

	Proposed	SPO	OFSY_330/DNR	RM3DE	EPI2
Boxes	27.541	20.269	3.574	20.842	16.443
Cotton	1.092	5.427	2.909	30.775	80.336
Dino	7.797	16.741	1.069	16.465	6.776
Sideboard	5.304	23.327	4.151	23.33	10.24

Table 4.9: BP (0.07) comparison with state of the art methods

	Proposed	SPO	OFSY_330/DNR	RM3DE	EPI2
Boxes	27.25 %	15.889%	19.246%	19.258%	29.795%
Cotton	5.81%	2.594%	3.036%	2.023%	16.694%
Dino	8.15%	2.184%	3.434%	3.892%	15.667%
Sideboard	12.93%	9.297%	10.355%	8.475%	18.953%

Chapter 5

Conclusion and future work

In this chapter we present the conclusions of our work and potential future work.

5.1 Conclusions

In this work several limitations of traditional state-of-the-art approaches were investigated and discussed. Then, techniques were developed to minimize their effects resulting in a complete disparity estimation framework achieving competitive results.

Firstly, the problem of enlarged silhouettes in the structure tensor based initial estimation was addressed by proposing a novel silhouette enhancement algorithm that matches the position of disparity map and image edges to detect and correct erroneous disparity. For this purpose, two different state-of-the-art edge detecting algorithms were used and compared. A novel method for improving the resulting silhouette maps based on the detection and correction of erroneously discontinuous edges was also developed.

Secondly, the low amplitude local noise that complicated the estimation of surface orientation from structure tensor obtained disparity maps, specially in smooth planar regions was addressed. By estimating planes in scene coordinates and projecting them back onto the camera plane, the median angle error of the surface normals generated from the resulting disparity maps was greatly improved.

The full framework proved very competitive in terms of MSE, and MAE.

5.2 Future work

While silhouette enhancement proved to be very beneficial, a simple median is used to inpaint the erroneous values, which is an over simplified approach, in some cases. However, the excessive local noise makes estimating the orientation of the relevant surfaces imprac-

tical. In the future, a possible method for interconnecting the Silhouette Enhancement algorithm with plane estimation could be based on reducing any error in the inpainting of erroneous areas, while simultaneously enhancing the accuracy of the borders of the estimated planes by including edge information.

Additionally, while the matting based optimization proved very reliable in terms of reducing the overall error of the image, over-smoothing problems remained one of the major issues of the framework, with the algorithm being too sensitive to errors in the segmentation process. An additional effort to reduce this sensitivity, or the introduction of a better segmentation algorithm could dramatically increase the performance of this entire framework. Alternatively, extensive testing of other state-of-the-art global optimization methods in the framework could prove fruitful.

Finally, in the Plane Noise Reduction algorithm, it would be important to overcome the dependency on the initial disparity estimation in the image to scene coordinate transformation, as it would reduce the mean angle error of the surface normals of well estimated surfaces tremendously. Further improvements to the algorithm could also come from ensuring smoothness in plane contours.

Bibliography

[1] E. Adelson and J. Bergen, "The plenoptic function and the elements of early vision," *Computational Models of Visual Processing*, pp. 3–20, 1991.

[2] S. Wanner, "Orientation Analysis in 4D Light Fields," Ph.D. dissertation, Universitat Heidelberg, 2014.

[3] I. Tosic and K. Berkner, "Light field scale-depth space transform for dense depth estimation," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 441–448, 2014.

[4] M. W. Tao, S. Hadap, J. Malik, and R. Ramamoorthi, "Depth from combining defocus and correspondence using light-field cameras," in *Proceedings of the 2013 IEEE International Conference on Computer Vision*, ser. ICCV '13. Washington, DC, USA: IEEE Computer Society, 2013, pp. 673–680. [Online]. Available: http://dx.doi.org/10.1109/ICCV.2013.89

[5] J. Bigun, *Vision with Direction*. Springer, 2006.

[6] R. Lourenco, P. A. A. Assuncao, L. M. N. Tavora, R. Fonseca-Pinto, and S. M. M. Faria, "Silhouette enhancement in light field disparity estimation using the structure tensor," in *2018 25th IEEE International Conference on Image Processing (ICIP)*, Oct 2018, pp. 2580–2584.

[7] S. McDonagh, R. B. Fisher, and J. Rees, "Using 3d information for classification of non-melanoma skin lesions," in *Proc. Medical Image Understanding and Analysis*, no. 164-168. BMVA Press, 2008.

[8] R. GmbH. 3d optical inspection. https://raytrix.de/inspection/. [Online]. Available: http://web.archive.org/web/20080207010024/http://www.808multimedia.com/winnt/kernel.htm

[9] H. Mihara, T. Funatomi, K. Tanaka, H. Kubo, Y. Mukaigawa, and H. Nagahara, "4d light field segmentation with spatial and angular consistencies," in *2016 IEEE International Conference on Computational Photography (ICCP)*, May 2016, pp. 1–8.

[10] R. Ng, "Digital light field photography," Ph.D. dissertation, Stanford university, 2006.

[11] R. C. Bolles, H. H. Baker, and D. H. Marimont, "Epipolar-plane image analysis: An approach to determining structure from motion," *International Journal of Computer Vision*, 1987.

[12] J. Bigün, "Optimal orientation detection of linear symmetry," 1986.

[13] J. Li and Z. N. Li, "Continuous depth map reconstruction from light fields," in *2013 IEEE International Conference on Multimedia and Expo (ICME)*, July 2013, pp. 1–6.

[14] S. Wanner and B. Goldluecke, "Variational light field analysis for disparity estimation and super-resolution," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 3, pp. 606–619, March 2014.

[15] ——, "Globally consistent depth labeling of 4d light fields," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, June 2012, pp. 41–48.

[16] D. Dansereau and L. Bruton, "Gradient-based depth estimation from 4d light fields," in *2004 IEEE International Symposium on Circuits and Systems (IEEE Cat. No.04CH37512)*, vol. 3, May 2004, pp. III–549.

[17] M. Strecke, A. Alperovich, and B. Goldluecke, "Accurate depth and normal maps from occlusion-aware focal stack symmetry," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[18] Williem, I. K. Park, and K. M. Lee, "Robust light field depth estimation using occlusion-noise aware data costs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, no. 99, pp. 1–1, 2017.

[19] S. Zhang, H. Sheng, C. Li, J. Zhang, and Z. Xiong, "Robust depth estimation for light field via spinning parallelogram operator," *Computer Vision and Image Understanding*, vol. 145, pp. 148–159, 2016.

[20] A. Neri, M. Carli, and F. Battisti, "A multi-resolution approach to depth field estimation in dense image arrays," in *2015 IEEE International Conference on Image Processing (ICIP)*, Sept 2015, pp. 3358–3362.

[21] T. Lindeberg, *Scale-space theory in computer vision.* Kluwer Academic Publishers, 1994.

[22] ——, "Scale-Space," in *Wiley Encyclopedia of Computer Science and Engineering*, 2009, pp. 2495–2504.

[23] I. Tosic and K. Berkner, "3D keypoint detection by light field scale-depth space analysis," *2014 IEEE International Conference on Image Processing, ICIP 2014*, pp. 1927–1931, 2014.

[24] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, ser. ICCV '99. Washington, DC, USA: IEEE Computer Society, 1999, pp. 1150–. [Online]. Available: http://dl.acm.org/citation.cfm?id=850924.851523

[25] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Comput. Vis. Image Underst.*, vol. 110, no. 3, pp. 346–359, Jun. 2008. [Online]. Available: http://dx.doi.org/10.1016/j.cviu.2007.09.014

[26] A. Chambolle, "An algorithm for total variation minimization and applications," *Journal of Mathematical Imaging and Vision*, vol. 20, no. 1, pp. 89–97, Jan 2004. [Online]. Available: https://doi.org/10.1023/B:JMIV.0000011325.36760.1e

[27] H. Hirschmüller, P. R. Innocent, and J. Garibaldi, "Real-time correlation-based stereo vision with reduced border errors," *International Journal of Computer Vision*, vol. 47, no. 1, pp. 229–246, Apr 2002. [Online]. Available: https://doi.org/10.1023/A:1014554110407

[28] H. Scharr, "Optimal operators in digital image processing," Ph.D. dissertation, Heidelberg University, 2000.

[29] A. Levin, D. Lischinski, and Y. Weiss, "A Closed Form Solution to Natural Image Matting." [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.65.2293{&}rep=rep1{&}type=pdf

[30] V. Kolmogorov and R. Zabih, "Multi-camera scene reconstruction via graph cuts," in *Proceedings of the 7th European Conference on Computer Vision-Part III*, ser. ECCV '02. Berlin, Heidelberg: Springer-Verlag, 2002, pp. 82–96. [Online]. Available: http://dl.acm.org/citation.cfm?id=645317.756415

[31] J. Sun, N.-N. Zheng, and H.-Y. Shum, "Stereo matching using belief propagation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 7, pp. 787–800, July 2003.

[32] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, Nov 1986.

[33] D. R. Castillo, H. P. Cabrera, and P. Assunção, "Edge detection based on Krawtchouk polynomials," *Journal of Computational and Applied Mathematics*, vol. 284, no. 0, pp. 244–250, August 2015.

[34] J. Baik, T. Kriecherbauer, K. R. McLaughlin, and P. Miller, *Discrete Orthogonal Polinomials: Asymptotics and Applications*. Princeton University Press, 2007.

[35] A. F. Nikiforov, V. B. Uvarov, and S. K. Suslov, *Classical Orthogonal Polynomials of a Discrete Variable*. Springer, Berlin, Heidelberg, 1991.

[36] Z. Jin, T. Tillo, and F. Cheng, "Depth-map driven planar surfaces detection," in *2014 IEEE Visual Communications and Image Processing Conference*, Dec 2014, pp. 514–517.

[37] O. Johannsen, K. Honauer, B. Goldluecke, A. Alperovich, F. Battisti, Y. Bok, M. Brizzi, M. Carli, G. Choe, M. Diebold, M. Gutsche, H. Jeon, I. S. Kweon, J. Park, J. Park, H. Schilling, H. Sheng, L. Si, M. Strecke, A. Sulc, Y. Tai, Q. Wang, T. Wang, S. Wanner, Z. Xiong, J. Yu, S. Zhang, and H. Zhu, "A taxonomy and evaluation of dense light field depth estimation algorithms," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, July 2017, pp. 1795–1812.

[38] K. Honauer, O. Johannsen, D. Kondermann, and B. Goldlücke, "A dataset and evaluation methodology for depth estimation on 4d light fields," in *Computer Vision - ACCV 2016 : 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part III*, ser. Lecture Notes in Computer Science, S.-H. Lai, Ed., no. 10113. Cham: Springer, 2017, pp. 19–34.

Appendix A

Published paper

This appendix presents the published paper, resulted from the research work done during this dissertation.

- R. Lourenco, P. A. A. Assuncao, L. M. N. Tavora, R. Fonseca-Pinto and S. M. M. Faria, "Silhouette Enhancement in Light Field Disparity Estimation Using the Structure Tensor," 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, 2018, pp. 2580-2584.