N^o d'ordre : 40421

THÈSE

présentée par

Hachem HALAWANA

pour obtenir le grade de

Docteur de l'Université Lille 1

Discipline : Automatique, Génie informatique, Traitement du signal et des images

Dématriçage partiel d'images CFA pour la mise en correspondance stéréoscopique couleur

Soutenue publiquement le 14 decembre 2010 devant le jury composé de :

M. Theo GEVERS	Professeur à l'Université Amsterdam	Rapporteur
M. Abdelaziz BENSRHAIR	Professeur à l'INSA de Rouen	Rapporteur
M. Denis HAMAD	Professeur à l'Université Littoral	Examinateur
M. Yassine RUICHEK	Professeur à l'Université UTBM	Examinateur
M. François CABESTAING	Professeur à l'Université Lille 1	Co-Directeur
M. Ludovic MACAIRE	Professeur à l'Université Lille 1	Co-Directeur

Thèse préparée au Laboratoire d'Automatique, Génie Informatique et Signal

(LAGIS FRE-CNRS-3303)

N° d'ordre : 40421

PHD THESIS

presented by

Hachem HALAWANA

to obtain the title of

PhD of University Lille 1

Discipline : Automatic control, Computer Engineering, Image and Signal Processing

Partial demosaicing of CFA images for stereo matching

Publicly defended 14 December 2010

Jury :

M. Theo GEVERS	Professeur à l'Université Amsterdam	Rapporteur
M. Abdelaziz BENSRHAIR	Professeur à l'INSA de Rouen	Rapporteur
M. Denis HAMAD	Professeur à l'Université Littoral	Examinateur
M. Yassine RUICHEK	Professeur à l'Université UTBM	Examinateur
M. François CABESTAING	Professeur à l'Université Lille 1	Co-Directeur
M. Ludovic MACAIRE	Professeur à l'Université Lille 1	Co-Directeur

Thesis prepared in laboratory of Automatic control, Computer Engineering and Image and Signal Processing

(LAGIS FRE-CNRS-3303)

À mes parents, à Maymouna,
à Adam, à toute ma famille,
à mes amis...

Acknowledgements

Je voudrais tout d'abord témoigner mes sincères remerciements et ma profonde reconnaissance à Monsieur Ludovic MACAIRE et Monsieur François Cabestaing, Professeurs à l'Université Lille 1, pour m'avoir accueilli au sein de l'équipe Vision & Image du LAGIS et m'avoir encadré durant la durée de réalisation de ce travail.

Je voudrais remercier Monsieur Theo GEVERS, Professeur à l'Université Amsterdam, et Monsieur Abdelaziz BENSRHAIR, Professeur à l'INSA de Rouen, qui ont eu la gentillesse d'accepter la tâche importante d'être rapporteurs. Je les remercie pour l'intérêt qu'ils ont porté à mon travail et les conseils avisés qui m'ont aidé à améliorer la version finale du manuscrit.

Je tiens aussi à remercier Monsieur Denis HAMAD, Professeur à l'Université Littoral, et Monsieur Yassine RUICHEK, Professeur à l'Université UTBM, d'avoir accepté d'être examinateurs de ma thèse.

Un grand merci s'adresse à tous les membres de l'équipe VI à Lille . Je remercie en particulier Monsieur Olivier LOSSON, MCF à l'Université de Lille 1, pour l'outil BibTeX dont il est responsable. Cet outil m'a beaucoup aidé à gérer mes références bibliographiques. Durant ces années de préparation de thèse, j'ai partagé des moments exceptionnels avec tous les collègues. Je leur exprime ma profonde sympathie et leur souhaite beaucoup de succès.Je souhaite beaucoup de courage à tous les collègues doctorants qui n'ont pas encore soutenu leur thèse , en particulier à : Mariam, Halima, Molka, Tarik, et Gaël.

Je ne peux terminer mes remerciements sans exprimer ma sincère gratitude à mes parents, à mes frères et à ma soeur pour leur affection et leurs encouragements. Je remercie du profond de mon coeur ma chérie Maymouna pour son soutien, son amour, pour ces années de compréhension, de patience et d'efforts communs.

Table des matières

Table des figures

General introduction

Context of the study

In the field of automotive vehicles, main recent applications require the perception of the three-dimensional real world. In this field, the goal is not necessarily the automatic steering of the vehicle, which will require further years of research and development, but only assistance to the driver. In this case, the system overcomes the defects of the driver such as the lack of vigilance and poor appreciation of distances by warning him when there is a potential danger. So, automotive vehicles have to detect the different objects that are on the road and represent them in the three-dimensional scene map.

To reach this goal, various sensor technologies can be used. For example, the radar emits electromagnetic waves toward the environment and determines the presence of objects by analyzing the return time of reflected waves. Sonar and lidar operate on the same principle, with a sound signal for the first, and a laser beam for the second. These sensors are classified as active because they emit a signal, then analyze signals reflected by objects in the scene.

Another type of sensors uses light from an outside source that has been reflected by objects in the scene. They are called passive sensors, and one example is the camera. The vision system objective is to mimic the visual system of animals or humans. A single camera provides an image of the scene viewed from a single point of view. In this case, the content of the image is analyzed to recognize, to a certain level of accuracy, the objects present in the scene.

However, most systems based on cameras, for the purpose of environment perception, use two cameras to observe the scene. Then, they use a technique similar to that used by most animals whose vision system has two eyes. Analyzing two images of the scene observed at the same time allows, under certain conditions, to recover the three-dimensional information which has been lost during the image formation process.

Many projects today propose to combine information from various active and passive sensors. The work in this thesis fits within one of the three components of the RaViOLi project (Radar and Steerable Vision and Lidar) headed by F. Cabestaing and L. Duvieubourg, Researchers in the LAGIS laboratory, and supported by the Regional Group of Transportation Research of the Nord-Pas-de-Calais.

In this project a new stereoscopic sensor has been developed. It consists of a single-CCD color camera, combined with a set of mirrors arranged in an original geometry to obtain the best configuration for foveal vision (see figure 0.1). Each image of the stereo pair is formed on one half of the CCD color camera using a specific arrangement of mirrors. Such an arrangement can acquire color images of distant objects of the stereoscope, and this, with excellent spatial resolution. This forms two virtual cameras from a single real camera and for this reason it is called stereoscopic sensor. The two images formed by the two virtual cameras are used to reconstruct the three-dimensional map of the observed scene.

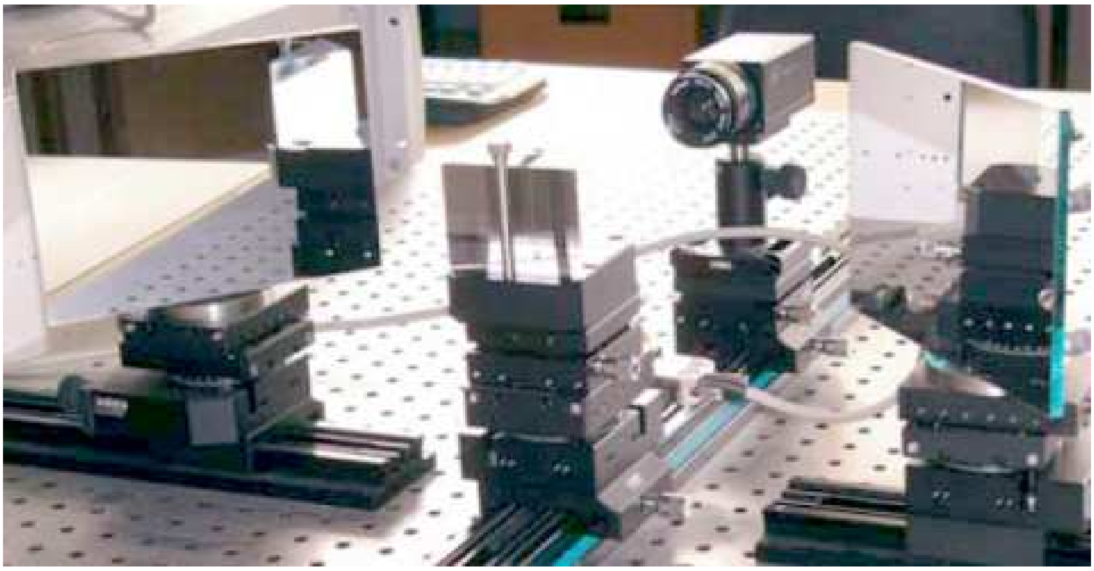


Figure 0.1 : Stereoscopic sensor equipped with one single-CCD color camera.

Stereovision

From two images (left and right) simultaneously acquired by two cameras, stereovision systems aims to recover the third dimension which has been lost during image formation.

The projections of the same scene point visible by the two cameras do not have the same coordinates in the two image planes. The exact position of this scene point can be retrieved if its two projections (called homologous) are identified. The problem of identifying homologous pixels in the two images is called stereo correspondence problem.

Stereo matching methods try to solve the stereo correspondence problem. Throughout this thesis, we concentrate our work on dense stereo matching methods that try to identify all homologous pixels in the two images to yield a dense three-dimensional map of the observed scene. Stereo matching methods are applied to pairs of stereo images that can be gray-level images or color images. In gray-level images each pixel is characterized by a gray-level value while in color images each pixel is characterized by three components Red (R), Green (G) and Blue (B).

Chambon *et al.* have compared widely used stereo matching methods applied to gray-level and color images [CC04]. They have shown that taking into account color information generally improves the performance reached by the matching process [PSZ05]. However, in their studies they did not mention how the color images had been acquired and it is well known that the acquisition device influences the performance reached by any color image analysis scheme.

Color images can be acquired by two types of cameras : those including three sensors associated with beam splitters and color filters providing the so-called full color images (each pixel is characterized by Red, Green and Blues levels), and those including a single-sensor.

The single-sensor cameras do not provide a full color image but actually deliver a color filter array (CFA) image, where each pixel is characterized by a single color component which can be Red, Green or Blue. So, one has to estimate the missing color components at each pixel. The process of estimating the missing color components is commonly referred to as CFA demosaicing, and yields a demosaiced color image where each pixel is characterized by an estimated color point [BGMT08].

Objective of the thesis

In the RaViOLi project, we use a single-CCD color camera to perceive the color of the observed objects. A straightforward method to match pixels in stereo images acquired using single-CCD color cameras is to reconstruct color images by demosaicing and then apply the matching process to the demosaiced color images.

As the demosaicing methods intend to produce "visually pleasing" demosaiced color images, they attempt to reduce the presence of color artifacts, such as false colors or zipper effects, by filtering the images [YLD07]. So, some useful color texture information for stereo matching may be altered in the color demosaiced images. Thus, the quality of stereo matching with demosaiced color stereo image pairs may suffer either from color artifacts or from the alteration of color texture caused by demosaicing schemes.

The objective of our thesis is to find an alternative solution to match pixels by analyzing CFA stereo images without reconstructing the color images by demosaicing.

Outline of the thesis

This PhD manuscript is divided into four chapters and a conclusion.

In the first chapter, we present the fundamental concepts of stereovision. Then, we define the stereo correspondence problem. We present dense stereo matching methods for solving this problem. We describe the two main classes of dense stereo matching methods : local and global ones. Global methods identify homologous pixels by analyzing the whole image while local ones analyze local neighborhood of pixels. Since stereovision scheme must respect real-time video constraints, we retain local methods which are less time consuming than global ones.

In the second chapter, we introduce the use of color information in dense stereovision. After having described why the demosaicing task is required by single-CCD color cameras, we present the Bayer CFA solution. Then, an introduction to the demosaicing issue and to its major solutions is exposed. After that, we show, thanks to a basic experiment, that the loss of information and/or the non-specific demosaicing step have a strong influence on the efficiency of color stereo matching techniques.

In the third chapter, we introduce a partial demosaicing scheme designed for images acquired by single-CCD cameras. In the partial demosaicing scheme, each pixel is characterized by two components and not three components as in a full demosaicing scheme. We show that this partial demosaicing scheme is more adapted for stereo matching than a full demosaicing scheme.

In the fourth chapter, we compare the matching results by applying local stereo matching method to partially demosaiced color images and fully demosaiced color images on benchmark images. Then, we try to show the reasons of the enhancement brought by partial demosaicing. We also examine if using single-CCD cameras instead of gray-level ones leads the stereovision schemes to improve their quality of matching.

Finally, we summarize all the topics discussed in this PhD and the main contributions of our work. Several ideas for future works are also proposed.

Chapitre 1

Fundamentals of computational stereovision

Résumé

Construire une description 3D d'une scène observée par des caméras représente un véritable challenge. A partir de deux images obtenues simultanément, nous pouvons, sous certaines conditions, retrouver partiellement l'information 3D qui a été perdue pendant le processus de formation de l'image.

La stéréovision consiste à calculer l'information 3D de la scène à partir des images de saisie 2D. Il s'agit d'une imitation d'une des capacités du système de vision de l'être humain dans lequel cette tâche est automatiquement effectuée par le cerveau.

Dans cette thèse, nous porterons notre attention sur la stéréovision binoculaire qui utilise deux caméras, appelées caméra gauche et caméra droite qui observent la scène à partir de différents points de vue, comme cela est illustré dans la figure 1.1a. Dans cette figure, qui montre la vue d'en haut d'une installation stéréovision, nous pouvons voir que chaque caméra a son propre point de vue. Ainsi, la scène est divisée en quatre points de vue :

- Le point de vue de gauche seulement, qui contient les points de la scène qui peuvent apparaître sur l'image de gauche ;
- Le point de vue de droite seulement, qui contient les points de la scène qui peuvent seulement apparaître sur l'image de droite ;

- Le point de vue commun, qui contient les points de la scène qui peuvent apparaître sur les images de gauche et de droite ;
- Le point de vue invisible par les deux caméras.

Par exemple, deux images capturées par les caméras gauche et droite sont montrées respectivement dans les figures 1.1b and 1.1c. L'image de gauche (respectivement de droite), est décomposée en deux parties, l'une correspondant à la projection des points de la scène qui proviennent du point de vue de gauche (respectivement de droite), et l'autre correspondant à la projection des points de la scène qui proviennent d'un point de vue commun (voir figure 1.1d).

Examinons les parties des images de gauche et de droite qui correspondent au point de vue commun. Dans le point de vue commun, une surface élémentaire d'un objet observé par les deux caméras, peut être représentée par un pixel dans la partie de gauche et un pixel dans la partie de droite. La localisation dans l'espace de ces pixels dans les images de gauche et de droite est différente. La différence entre ces localisations est utilisée pour estimer la distance qui sépare la surface élémentaire des caméras. En étendant cette procédure à toutes les surfaces élémentaires observées par les deux caméras, nous pouvons estimer les distances séparant chaque objet de la scène 3D des caméras.

Dans la première partie de ce chapitre, nous introduirons quelques propriétés de la projection perspective. Ces propriétés seront ensuite utilisées pour décrire le modèle de Pinhole d'une caméra. Puis, nous introduirons tous les paramètres nécessaires pour définir la transformation d'un point d'une scène 3D en un point de l'image 2D dans l'image plane de la caméra. Après cela, les différentes propriétés de la stéréovision binoculaire seront décrites.

Dans une seconde partie, nous introduirons le problème de correspondance stéréovision qui est accentué par plusieurs phénomènes. Ensuite, nous présenterons plusieurs hypothèses qui seront utilisées pour simplifier sa résolution.

Enfin, nous présenterons les méthodes de correspondance stéréovision éparse et dense utilisées pour résoudre le problème de correspondance stéréovision. Cependant, dans cette présentation de l'état de l'art, nous limiterons notre description aux méthodes de correspondance stéréovision dense utilisant les approches locales et globales.

1.1 Introduction

Building a three-dimensional description of a scene observed by cameras is a challenging task. From two images simultaneously acquired, we can under certain conditions recover partially the 3D information which has been lost during the image formation process.

Stereovision consists in computation of the three-dimensional information of the scene from two-dimensional input images. It is an imitation of one of the abilities of the human vision system where this task is automatically performed by the brain.

In this thesis, we focus our attention on binocular stereovision which uses two cameras, called left and right cameras, that observe the scene from different points of view, as illustrated in figure 1.1a. In this figure, which shows the top view of a stereovision setup, one can see that each camera has its own field of view. So, the scene is divided into four fields :

- left field of view only, which contains scene points that may only appear on the left image ;
- right field of view only, which contains scene points that may only appear on the right image ;
- common field of view, which contains scene points that may appear on both left and right images ;
- field invisible by both cameras.

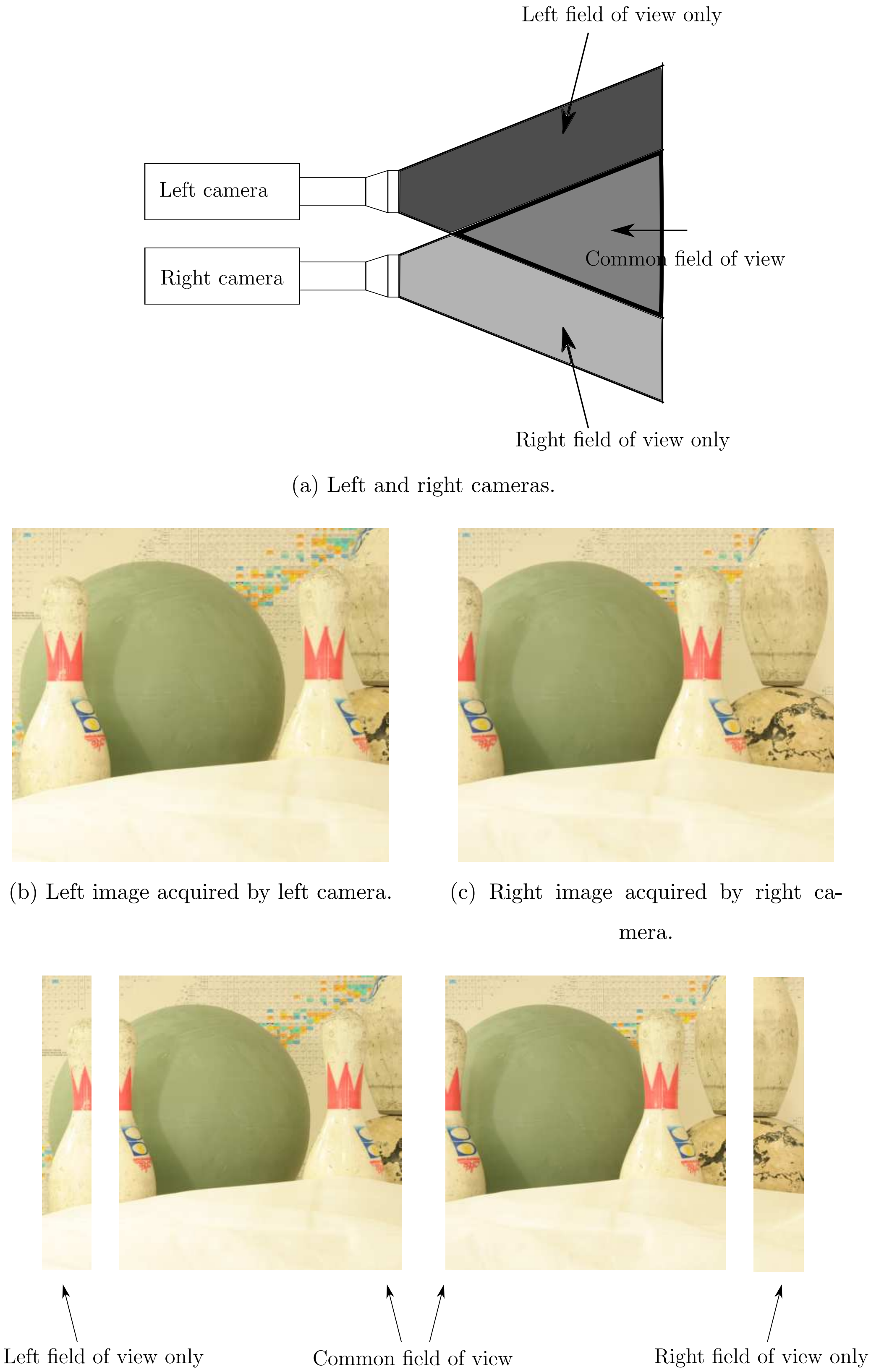
For example, two images captured by left and right cameras are shown in figures 1.1b and 1.1c, respectively. The left (right, respectively) image is decomposed into two areas, one that corresponds to the projection of scene points that belong to the left (right, respectively) field of view only and one that corresponds to the projection of scene points that belong to the common field of view (see figure 1.1d).

Let us examine the areas in the left and right images that correspond to the common field of view. In the common field of view, an elementary surface of an object that is observed by the two cameras, can be represented by one pixel in the left area and one pixel in the right area. The spatial locations of these pixels in the left and right images are different. The difference between these spatial locations is used to estimate the distance which separates the elementary surface from cameras. By extending this scheme to all elementary surfaces observed by both cameras, we can estimate the distances separating each object in the three-dimensional scene from the cameras.

In the first part of this chapter, we introduce some properties of the perspective projection. These properties are then used for describing the pinhole model of a camera. Then, we introduce all the camera parameters necessary to define the transformation of a three-dimensional scene point into a two-dimensional image point in the camera image plane. After that, the various properties of the binocular stereovision setup are described.

In the second part, we introduce the stereo correspondence problem that is complicated by many phenomena. Then, we present several assumptions that are used to simplify its resolution.

Finally, we present the sparse and dense stereo matching methods used to solve the stereo correspondence problem. However, in this presentation of the state of the art, we limit our description to dense stereo matching methods using local and global approaches.



(a) Left and right cameras.

(b) Left image acquired by left camera.

(c) Right image acquired by right camera.

(d) Images representing the different fields of figure 1.1a.

Figure 1.1 : Stereovision example.

1.2 Perspective projection

In this section, we present some basic concepts of perspective projection which are necessary for defining the geometrical model of a camera. The simplest approach to describing the properties of optical systems is the perspective projection. We begin by showing how to determine the projection of space points onto one projection plane. Then, the case of two projections onto two planes is described. Finally, we present fundamental properties of the epipolar geometry.

1.2.1 Single perspective projection

The projection of a space point P is located at the intersection of the straight line connecting it with the center of projection O and the projection plane. This is shown in figure 1.2. All the points on the straight line OP are projected onto the same point p. So, this is a non inversible transformation.

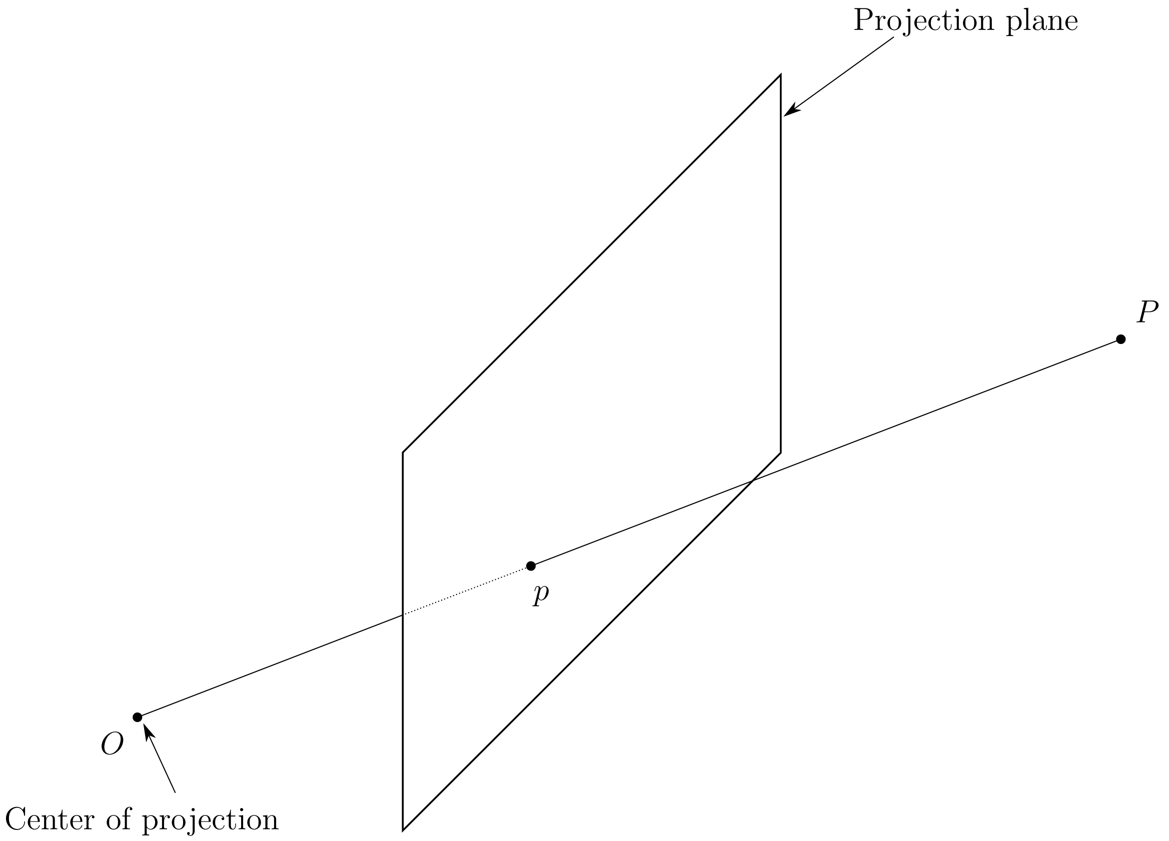


Figure 1.2 : Single Projection.

1.2.2 Two perspective projections

Let us consider now two projection planes. In order to define notations that will be adapted to stereovision, we will call these planes respectively the left and right projection planes and denote them l and r, respectively. The projection centers are respectively denoted O_l and O_r. The straight line connecting them is called the baseline. A space point P is projected onto the left projection plane at p_l and onto the right one at p_r as shown in figure 1.3.

As explained in the previous section, the two projections of P are given by the intersections of the straight lines O_lP and O_rP with the projection planes l and r. In contrast with single projection, the location of the space point P can be deduced from its two projections p_l and p_r, since P lies at the intersection of the straight lines O_lp_l and O_lp_r.

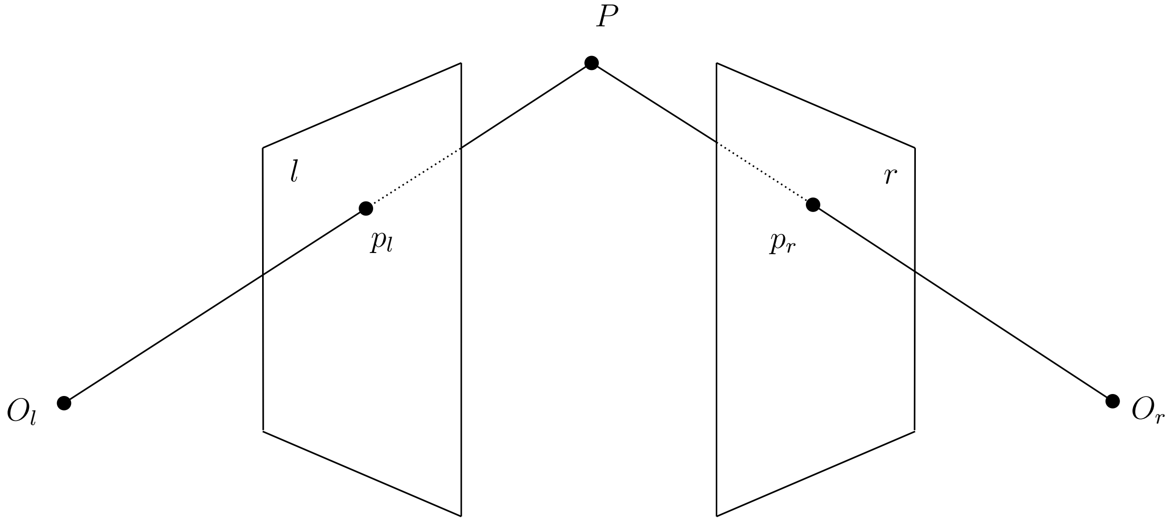


Figure 1.3 : Two perspective projections.

1.2.3 Epipolar geometry

A very interesting set of properties, called epipolar geometry, relates the two perspective projections shown in figure 1.3. We define several terms :

- the **epipolar plane** of a space point P is the plane determined by P and the projection centers O_l and O_r ;
- the **left epipole** is the projection onto the left projection plane of the right projection center and the **right epipole** is the projection onto the right projection

plane of the left projection center. Eipoles are denoted e_l and e_r. In other words, the left and right epipoles e_l and e_r are the two intersections of the baseline with the projection planes ;

– to any space point, we associate two **epipolar lines**. The epipolar lines are the intersections of the epipolar plane of the point with the projection planes. Analyzing figure 1.4, we can deduce that the left and right epipolar lines are the projections of the straight lines O_rP and O_lP onto the left and the right projection planes, respectively. For this reason, all left and right epipolar lines pass through left and right epipoles, respectively. We call the right epipolar line e_rp_r, the epipolar line corresponding to the left point p_l and vice versa.

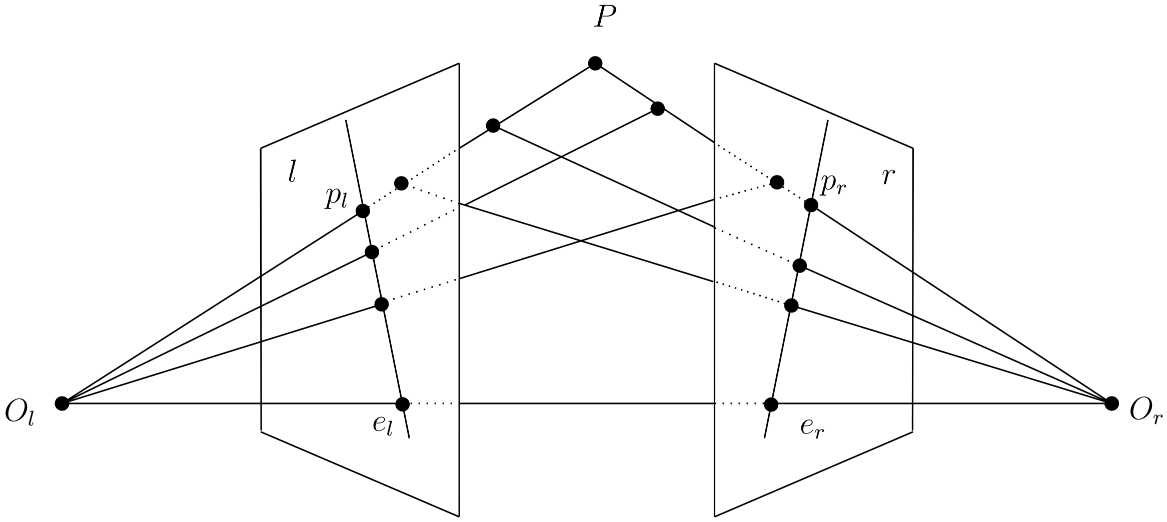


Figure 1.4 : Epipolar Geometry.

The epipolar geometry describes the relation between right and left projections of a space point P. Hence, the very important property, called epipolar property : **given a space point P, its right projected point p_r lies on the right epipolar line corresponding to its left projected point p_l and vice versa**.

1.3 Stereovision setup

1.3.1 Introduction

In this part, we introduce the geometry of the stereovision setup by first modeling the image formation by a pinhole camera. The perspective projection explained in the previous section is used for this purpose. The intrinsic and extrinsic parameters of the camera are then defined. Using these parameters, we can define the transformation from scene points to image points. Then, we present a binocular stereoscopic system. The various properties of this system are detailed.

1.3.2 Camera model and image formation

An image results from the optical projection of a three-dimensional scene onto the image plane. Modeling of image formation has been the subject of numerous works. The pinhole camera model is the simplest and the most used for describing the geometrical properties of cameras equipped with a thin lens. Geometrical properties of image formation can be explained thanks to the perspective projection described in section 1.2. Light rays reflected by each surface element of a scene object pass through the pinhole, called hereafter optical center, before reaching the image plane of the camera. This model is illustrated in figure 1.5.

In this figure, the pinhole camera produces an inverted image. In order to avoid the inversion of coordinates, we can describe the projection using a direct configuration, i.e. by placing the image plane in front of the optical center instead of behind. Doing this, we no longer have to rotate the obtained image by 180° in order to well represent the scene observed by the camera.

We write scene points with capital letters and their projected image points with small letters. Unlike the perspective projection described in figure 1.2, which just defines the geometrical transformation, each point p in the image plane can only correspond to one scene point P that is *visible* by the camera.

The pinhole camera can be modeled by providing two set of parameters : the intrinsic and extrinsic parameters. In order to clearly describe these parameters, let us define three orthogonal coordinate systems (see figure 1.5) :

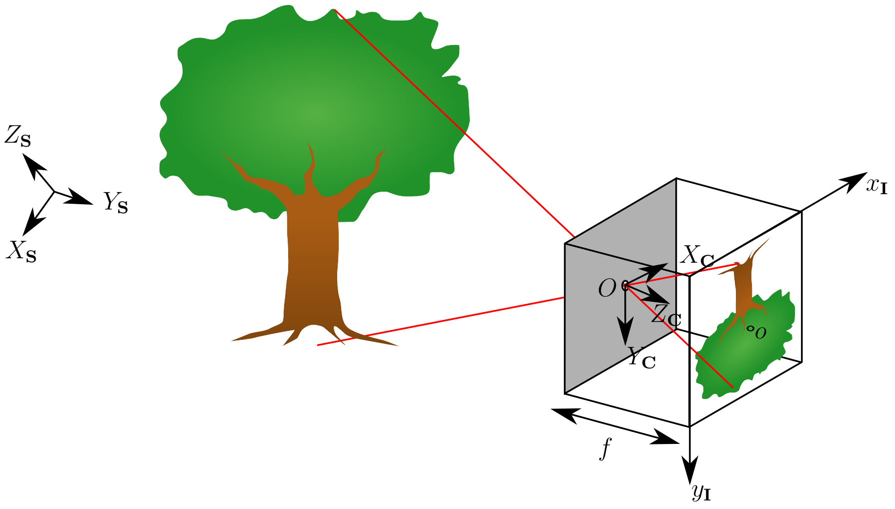


Figure 1.5 : Pinhole camera model.

- the scene coordinate system $(X_{\mathbf{S}},Y_{\mathbf{S}},Z_{\mathbf{S}})$;
- the camera coordinate system $(X_{\mathbf{C}},Y_{\mathbf{C}},Z_{\mathbf{C}})$;
- the image coordinate system $(x_{\mathbf{I}},y_{\mathbf{I}})$.

The camera coordinate system is adjusted so that its $X_{\mathbf{C}}$-axis and $Y_{\mathbf{C}}$-axis are parallel to the $x_{\mathbf{I}}$-axis and $y_{\mathbf{I}}$-axis of the image coordinates system, respectively. The $Z_{\mathbf{C}}$-axis is the optical axis of the camera.

The intrinsic parameters represent the internal characteristics of the camera. This includes scale change coefficients on the image, and the focal distance f of the camera lens, i.e. the distance separating the optical center from the center of the image plane. Using these parameters, we can define the transformation of a scene point in the camera coordinate system into its projected point in the image coordinate system.

Let us denote by $(X_{\mathbf{S}}^P,Y_{\mathbf{S}}^P,Z_{\mathbf{S}}^P)^T$ the coordinates of P in the scene coordinate system and $(X_{\mathbf{C}}^P,Y_{\mathbf{C}}^P,Z_{\mathbf{C}}^P)^T$ its coordinates in the camera coordinate system. The coordinates of its projected point p are denoted $(x_{\mathbf{I}}^p,y_{\mathbf{I}}^p)^T$.

In order to define this transformation with linear equations, we use the homogeneous coordinates. The homogeneous coordinates of a scene point P in the camera coordinate system are defined by the vector $(X_{\mathbf{C}}^P,Y_{\mathbf{C}}^P,Z_{\mathbf{C}}^P,1)^T$ and the homogeneous coordinates of its projected point p in the image coordinates system are defined by the vector $(x_{\mathbf{I}}^p,y_{\mathbf{I}}^p,1)^T$.

Equation (1.1) defines the intrinsic matrix $\mathbf{M_{int}}$ of intrinsic parameters in homogeneous coordinates :

$$\mathbf{M_{int}} \ = \ \begin{pmatrix} f\cdot\alpha_x & 0 & x^o_{\mathbf{I}} & 0 \\ 0 & f\cdot\alpha_y & y^o_{\mathbf{I}} & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad , \tag{1.1}$$

where α_x and α_y are scale change coefficients, and $(x^o,y^o,1)^T$ are the homogeneous coordinates of the center o of the image in image coordinate system. In most cases, the center o is the intersection of the optical axis with the image plane.

The extrinsic parameters of the camera are all the geometrical parameters necessary to model the transformation from the camera coordinate system to the scene coordinate system and vice versa. These parameters are described by the extrinsic matrix $\mathbf{M_{ext}}$, presented in equation (1.2). This block matrix corresponds to the association of a rotation matrix $\mathbf{R^C_S}$ and a translation vector $\mathbf{T^C_S}$:

$$\mathbf{M_{ext}} \ = \ \begin{pmatrix} \mathbf{R^C_S} & \mathbf{T^C_S} \\ \mathbf{0_3}^T & 1 \end{pmatrix} \quad , \tag{1.2}$$

where :

- the matrix $\mathbf{R^C_S}$, of dimension 3×3, represents the orientation of the scene coordinate system with respect to the camera coordinate system ;
- the column vector $\mathbf{T^C_S}$, of dimension 3×1, represents the coordinates of the center of the scene coordinate system in camera coordinate system ;
- $\mathbf{0_3}$ is a zero column vector of dimension 3×1 and $\mathbf{0_3}^T$ is its transpose.

Thus, given a scene point P and its projected point p, we can use the equations (1.1) and (1.2) to formulate the following relation :

$$Z^P_{\mathbf{C}} \cdot \begin{pmatrix} x^p_{\mathbf{I}} \\ y^p_{\mathbf{I}} \\ 1 \end{pmatrix} = \mathbf{M_{int}} \cdot \begin{pmatrix} X^P_{\mathbf{C}} \\ Y^P_{\mathbf{C}} \\ Z^P_{\mathbf{C}} \\ 1 \end{pmatrix} = \mathbf{M_{int}} \cdot \mathbf{M_{ext}} \cdot \begin{pmatrix} X^P_{\mathbf{S}} \\ Y^P_{\mathbf{S}} \\ Z^P_{\mathbf{S}} \\ 1 \end{pmatrix} = \mathcal{M} \cdot \begin{pmatrix} X^P_{\mathbf{S}} \\ Y^P_{\mathbf{S}} \\ Z^P_{\mathbf{S}} \\ 1 \end{pmatrix} , \tag{1.3}$$

where $\mathcal{M}$, called the projection matrix, is the product of the intrinsic and extrinsic matrices.

1.3.3 Binocular stereoscopic vision

Binocular stereoscopic vision systems use two cameras. Let P be a scene point visible by the two cameras and projected onto p_l in the left image plane and onto p_r in the right image plane. Since these image points p_l and p_r correspond to the same point of the scene, they are called homologous points. We add the subscripts l and r to the notations of projected points, coordinates and lines in the left and right image planes, respectively.

In this section, we will explain some properties of the binocular stereoscopic vision systems that are very useful for the rest of our study in this chapter.

1.3.3.1 Epipolar property

Using properties of the epipolar geometry listed in section 1.2.3, we know that the homologous points p_l and p_r lie in the epipolar lines corresponding to the same scene point P. In homogeneous image coordinates, a straight line in the image plane can be represented by a vector of dimension 3×1. Let vector $\mathbf{L}$ represent a line in an image plane. Then, a point p with homogeneous image coordinates $(x^p_{\mathbf{I}}, y^p_{\mathbf{I}}, 1)^T$ in this plane belongs to line $\mathbf{L}$ if :

$$\mathbf{L}^T \cdot \begin{pmatrix} x^p_{\mathbf{I}} \\ y^p_{\mathbf{I}} \\ 1 \end{pmatrix} = 0. \tag{1.4}$$

So, let us denote by :

- $\mathbf{L}_l$ the vector representing the left epipolar line in homogeneous image coordinate system of the left image ;
- $\mathbf{L}_r$ the vector representing the right epipolar line in homogeneous image coordinate system of the right image ;
- $(x^p_{\mathbf{I}l}, y^p_{\mathbf{I}l}, 1)$ the homogeneous coordinates of p_l in the left image coordinate system ;
- $(x^p_{\mathbf{I}r}, y^p_{\mathbf{I}r}, 1)$ the homogeneous coordinates of p_r in the right image coordinate system.

Transposing the properties of epipolar geometry to cartesian formalism, it can be shown that the equation of the epipolar line corresponding to p_l in the right image is defined by the following equation :

$$\mathbf{L}_r = F \cdot \begin{pmatrix} x^p_{\mathbf{I}l} \\ y^p_{\mathbf{I}l} \\ 1 \end{pmatrix}, \tag{1.5}$$

where F, called the fundamental matrix, fully describes the epipolar geometry of the stereo system.

The fundamental matrix can be calculated in function of the projection matrices of both camera by using the following equation [HZ04] :

$$F = skew(\mathcal{M}_r \cdot \mathbf{O}^{\mathbf{S}}_l) \cdot \mathcal{M}_r \cdot \mathcal{M}^T_l \cdot (\mathcal{M}_l \cdot \mathcal{M}^T_l)^{-1}, \tag{1.6}$$

where :

- $\mathcal{M}_r$ is the projection matrix of the right camera ;
- $\mathcal{M}_l$ is the projection matrix of the left camera and $\mathcal{M}^T_l$ its transpose ;
- $\mathbf{O}^{\mathbf{S}}_l$ represents the homogeneous coordinates of left camera optical center in scene coordinate system ;
- $skew(\mathbf{a})$ is the skew-symmetric matrix corresponding to a vector $\mathbf{a} = (a_1,a_2,a_3)^T$ defined as :

$$skew(\mathbf{a}) \quad = \quad \begin{pmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{pmatrix} \quad . \tag{1.7}$$

1.3.3.2 Half-occlusion phenomenon

In binocular stereoscopic vision, we want to project a scene point onto two image planes. However, this is not possible for every point of the scene. The reason is that there are scene points that are visible by only one camera [H0̈7, OFA05]. We refer to those points as half-occluded. The half-occlusion phenomenon is illustrated in figure 1.6. In this figure, the scene point B is projected onto b_l in the left image plane but is not visible by the right camera. Similarly, the scene point C is projected onto c_r in the right image plane but is not visible by the left camera. Analyzing figure 1.6, we deduce that all the scene points between A and B are invisible for the right camera, while the scene points between C and D are invisible for the left camera.

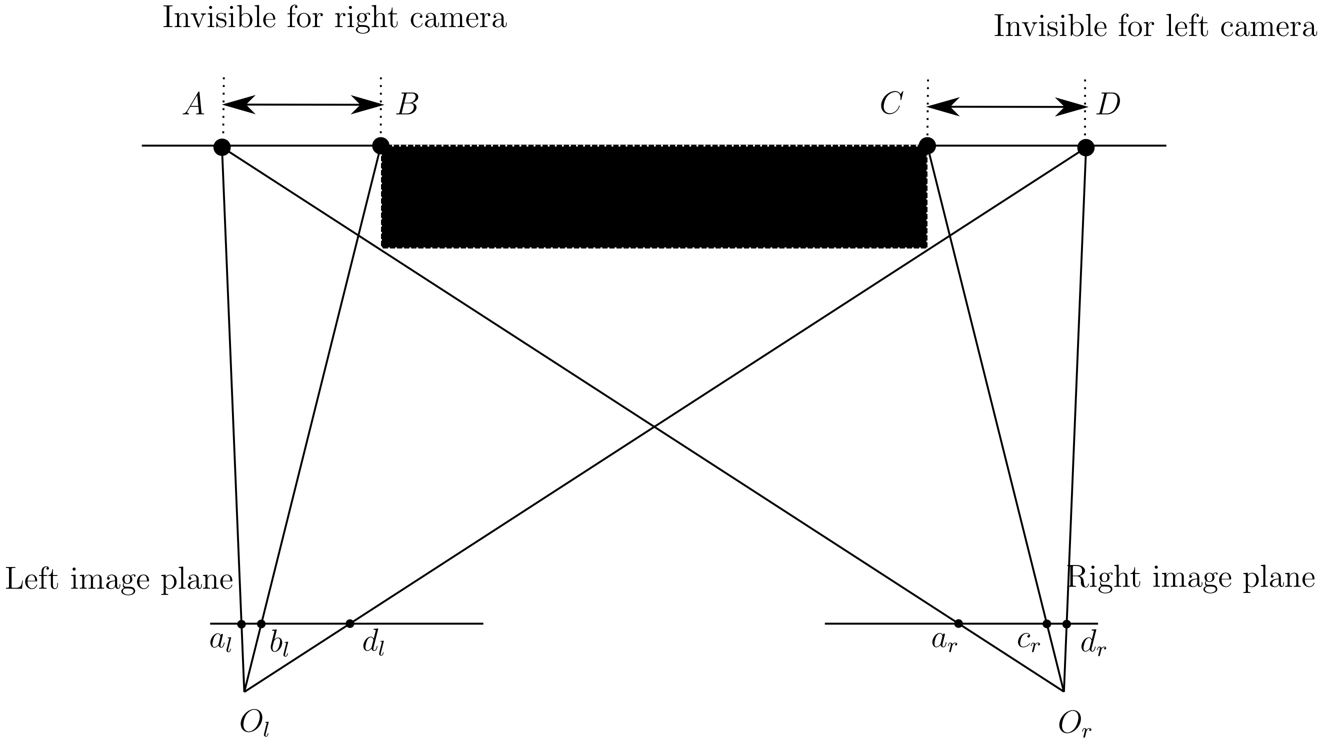


Figure 1.6 : Half-occlusion phenomenon.

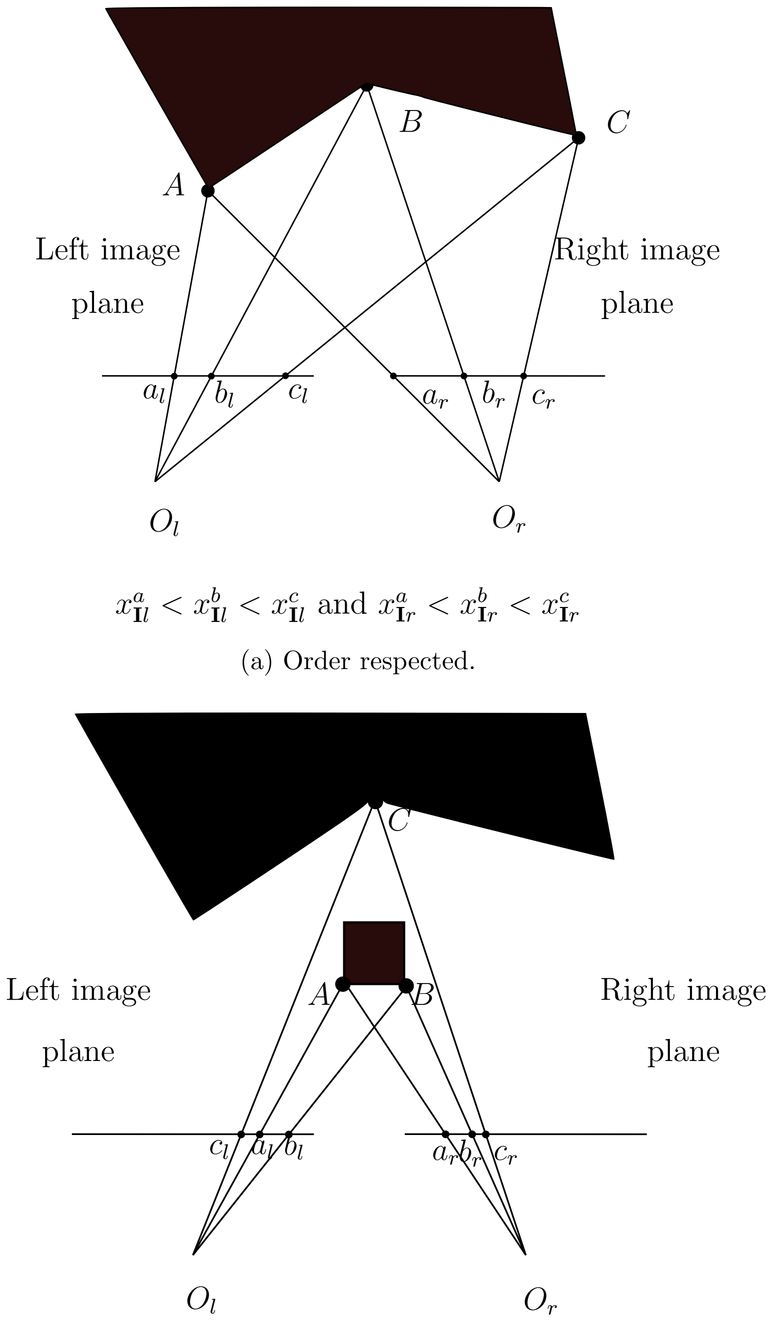
1.3.3.3 Order phenomenon

In stereoscopic vision, one usually assumes that the rank order of $X_{\mathbf{C}}$-coordinates of scene points is the same as that of their projections in both image planes. In figure 1.7a, the rank order of the projections of scene points ABC is conserved in the left and right image planes. However, the hypothesis of order conservation does not always hold true. For example, when the scene points belong to objects which are half-occluded, the rank order may not be respected, as shown by figure 1.7b.

1.3.3.4 Calibration of stereo setup

The calibration step consists in calculating the parameters required to compute precisely the transformation from coordinates of a three-dimensional scene point observed by the two cameras to coordinates of its projected points onto the left and right image planes [HS97].

These parameters are the intrinsic and extrinsic parameters of each camera of the stereo system. Accurate estimates of these parameters are necessary in order to relate the



$x^a_{\mathbf{I}l} < x^b_{\mathbf{I}l} < x^c_{\mathbf{I}l}$ and $x^a_{\mathbf{I}r} < x^b_{\mathbf{I}r} < x^c_{\mathbf{I}r}$

(a) Order respected.

$x^c_{\mathbf{I}l} < x^a_{\mathbf{I}l} < x^b_{\mathbf{I}l}$ and $x^a_{\mathbf{I}r} < x^b_{\mathbf{I}r} < x^c_{\mathbf{I}r}$

(b) Order not respected.

Figure 1.7 : Order phenomenon.

scene coordinate system to the image coordinate system. From these parameters, we can compute the fundamental matrix using equation (1.6).

However, we can also estimate the fundamental matrix without any prior knowledge of the cameras internal parameters or the relative position between them, by using the eight point algorithm proposed by Longuet-Higgins [LH87]. This algorithm analyzes a predefined set of eight pairs of homologous points. The problem of performing stereo calibration is well understood and high-quality software toolkits are available. For a detailed presentation of recent works on calibration, the reader is invited to read [FL01, HZ04, Hen00].

1.4 Stereo correspondence problem

1.4.1 Introduction

In the previous section, we presented the binocular stereoscopic vision. From a scene point P, using the projection matrices of the left and right cameras, we can find the location of the projected points onto the left and right image planes. Suppose now that the problem is inverted. The two projected points are identified, and we want to find the location of the scene point P. P is the intersection of the straight lines O_lp_l and O_rp_r. So, the scene point P can be recovered if the pair of left and right projected points are identified.

Given only one projection of a scene point P, the correspondence problem consists in determining its homologous one in the other image plane if it exists. It is a complex problem that has received a huge attention for more than 40 years and till now. Inspite of this, finding homologous points is not always possible.

Suppose that the spatial location of the projected point p_l of P is known and that p_r exists. We want to find the location of p_r in the right image plane. This seems to be a two-dimensional search problem. However, thanks to the properties of the epipolar geometry explained in section 1.2.3, we know that the right projection p_r lies on the epipolar line corresponding to p_l in the right image plane. So, the correspondence problem is in fact reduced to a one-dimensional search problem.

However, the homologous point might not even exist in case of half-occlusions as explained in section 1.3.3.2. In this case, there are several methods that handle this problem. These methods are classified into three classes :

- methods that detect half-occlusion ;
- methods that reduce sensitivity to half-occlusion ;

– methods that model the half-occlusion geometry.

A complete review of these methods was made by M.Z. Brown [BBH03] and by Chambon [Cha05].

To identify homologous points, we need an extra information about them in addition to their coordinates. This information, related to photometry, is delivered by digital cameras which are presented in the following. Then, a variety of assumptions commonly exploited to make the resolution of stereo correspondence problem possible are introduced. We prefer to use the word 'assumptions' instead of 'constraints' since they do not always hold true. On the opposite, the epipolar property imposes what is called the epipolar constraint, since it always holds true unless the calibration data are erroneous.

1.4.2 Image digitizing

Till now, we talked about points projected onto image planes. However, in order to be able to process an image by computers, it should be digital. Nowadays, digital images are acquired by digital cameras. In these cameras, the image plane is digitalized into picture elements called pixels. Each pixel is associated with a photosensor element which measures the perceived light stimulus intensity. The gray-level value denoted I and associated to the corresponding pixel depends on the measured intensity and on the camera bias and gain.

The image sensor is composed of M columns and N lines of photosensors, hence the digital image is composed of $M \times N$ pixels. Let us denote h_x and h_y the side lengths of image sensor along $x_{\mathbf{I}}$-axis and $y_{\mathbf{I}}$-axis, respectively. Then, the scale change coefficients α_x and α_y of the intrinsic matrix of the camera (see equation (1.1) are defined as :

$$\alpha_x = \frac{M}{h_x} \tag{1.8}$$

$$\alpha_y = \frac{N}{h_y}. \tag{1.9}$$

So, each projected point p with coordinates $(x_{\mathbf{I}}^p, y_{\mathbf{I}}^p)^T$ in the image coordinate system is associated to the pixel with integer coordinates (x^p, y^p) in pixel coordinate system. The relation between the two coordinates is expressed as :

$$x^p = int[x_{\mathbf{I}}^p], \tag{1.10}$$

$$y^p = int[y_{\mathbf{I}}^p], \tag{1.11}$$

where $int[x]$ is the integer part of x. In other words, all image points in the surface of a photosensor corresponding to a pixel are associated to it.

From now and on, we will use pixels instead of points in the image planes. For simplicity purpose, we suppose that a visible scene point P is projected onto a pixel p_l in the left image with coordinates (x_l^p, y_l^p) and onto a pixel p_r in the left image with coordinates (x_r^p, y_r^p) where x_l^p, y_l^p, x_r^p and y_r^p are integer values. Pixels p_l and p_r are characterized by gray-levels $I_l(x_l^p, y_l^p)$ and $I_r(x_r^p, y_r^p)$, respectively. The superscript p in x and y coordinates is deleted when its absence does not make confusion.

Pixels p_l and p_r are called homologous pixels, since they correspond to homologous image points. It is important to notice that, because of the spatial sampling, one pixel can have more than one homologous pixel.

In order to find homologous pixels, we analyze the gray-levels of pixels in the two images.

1.4.3 Photometric consistency assumption

We assume that the Lambertian model is used for describing incident light reflection by the surface of an object. This model explicits that the object's surface reflects incident light identically in all directions. When using this physical constraint, the same light stimulus reflected by the surface element of an observed object reaches the two homologous pixels in the two images. So, the gray-level values $I_l(x_l, y_l)$ and $I_r(x_r, y_r)$ of these two pixels should be equal, assuming that the gains and biases of both cameras are the same.

In practice, the assumption of Lambertian surfaces is violated by specular reflections. The light stimulus perceived by each camera may substantially change depending on the viewpoint. Therefore, the gray-levels of two homologous pixels can be different.

Even if the Lambertian assumption is not violated, there are other sources of variation of the gray-levels, such as the difference in sensor characteristics of the two cameras. Nevertheless, image acquisition is also commonly corrupted by a certain amount of noise introduced by the camera electronic devices [WTTW06].

1.4.4 Unicity assumption

The unicity assumption simply states that a pixel in one image has at most one homologous pixel in the other image. We have seen that the spatial sampling process theoretically removes the unicity property existing for double perspective projection in the case of continuous images. However, when image sampling is fine enough, the unicity assumption allows one to search a single homologous pixel rather than several ones.

This assumption is often used to identify half-occlusions by enforcing one-to-one correspondences for scene points which are visible by the two cameras [Ble06].

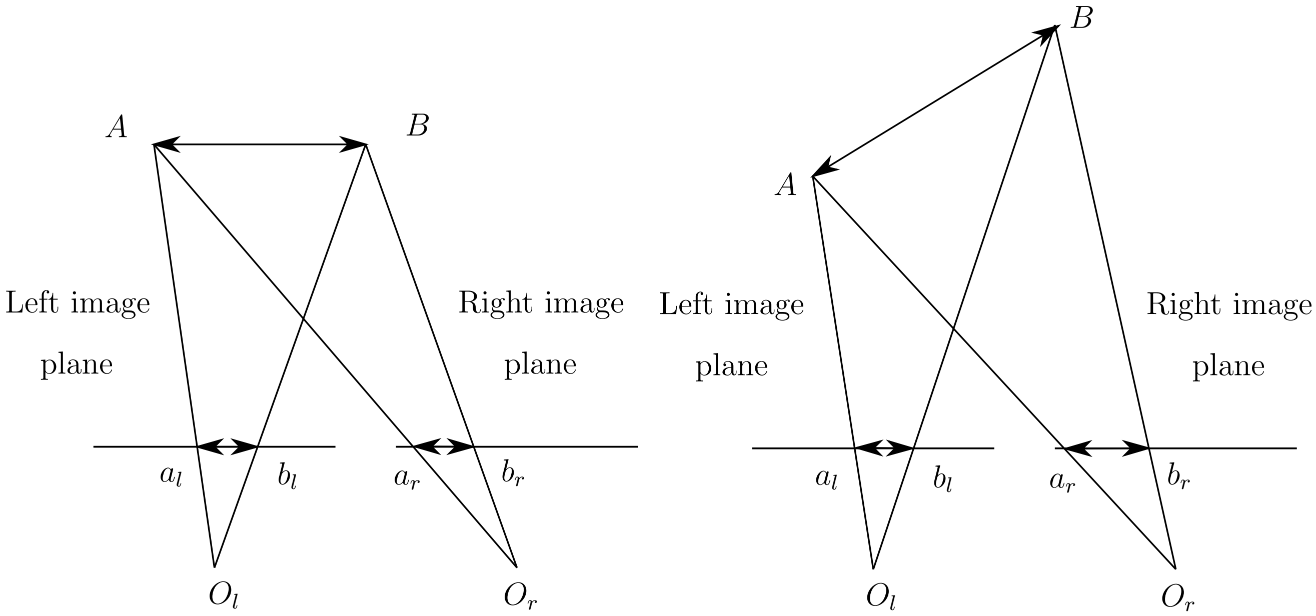
Even in the case of fine sampling, the unicity assumption is violated for horizontally slanted surfaces [OA05]. Let us consider a slanted segment and a fronto-parallel segment bounded by points A and B which are observed by two the cameras (see figure 1.8). The point A is projected onto pixel a_l with coordinates (x_l^a, y_l^a) in the left image and onto pixel a_r with coordinates (x_r^a, y_r^a) in the right image. Similarly, the point B is projected onto pixel b_l with coordinates (x_l^b, y_l^b) in the left image and onto pixel b_r with coordinates (x_r^b, y_r^b) in the right image.

In the fronto-parallel case, the segment AB is projected onto the same number of pixels in the left and right images (see figure 1.8a). Hence, the unicity assumption is not violated.

However, in the slanted case, the segment AB is projected onto different number of pixels in the left and right images (see figure 1.8b). Hence, there are pixels in one image that are homologous to more than one pixel of the other image.

1.4.5 Order assumption

The order phenomenon in stereoscopic vision has been introduced in section 1.3.3.3. When solving the stereo correspondence problem, many algorithms assume that the rank order of pixels on which are projected the scene points is respected.



(a) Projection of a fronto-parallel slanted line. (b) Projection of a horizontally slanted line.

Figure 1.8 : Unicity assumption.

1.5 Stereo matching strategies

Techniques for solving the stereo correspondence problem by analyzing a pair of stereo images have a high computational complexity. For each pixel in the left image, there are a lot of possible candidate right pixels to be examined in order to determine the best correspondence. We assume that the homologous right pixel corresponds to the best correspondence. Stereo matching methods, that intend to solve the correspondence problem, can be divided into two classes : sparse and dense methods [Wor07].

Sparse methods match features that have been identified in the stereo image pair. The used features are edges, line segments, curve segments, etc. For this reason, these methods are also called feature-based methods. The matching process is applied only on the detected features [Wu00]. Through 1980s, feature matching methods for stereo correspondence received significant attention. It is mainly due to their low computational complexity. So, they are well suited for real time applications [MPP06].

However, the interest in this type of methods has declined in the last decade. This is due to the increase of applications that need an accurate identification of all the homologous pixels in the stereo image pair [BBH03]. The stereo matching methods that provide all the homologous pixels in the stereo image pair are called dense methods. We will only present the dense stereo matching methods that are proposed for solving the correspondence problem.

Since there is a large amount of papers about solving the stereo correspondence problem, an exhaustive review is difficult to be achieved. We therefore focus our summary on a few techniques that we consider as important ones. A review of recent dense stereo matching methods has been published by Scharstein and Szeliski [SS02]. The authors identify four steps that are usually performed by a dense stereo matching method :

- matching cost computation ;
- cost aggregation where the initial matching costs are spatially aggregated over a pixel neighborhood ;
- optimization in which, the best correspondence is determined at each pixel ;
- matching results refinement where miss-matches are removed.

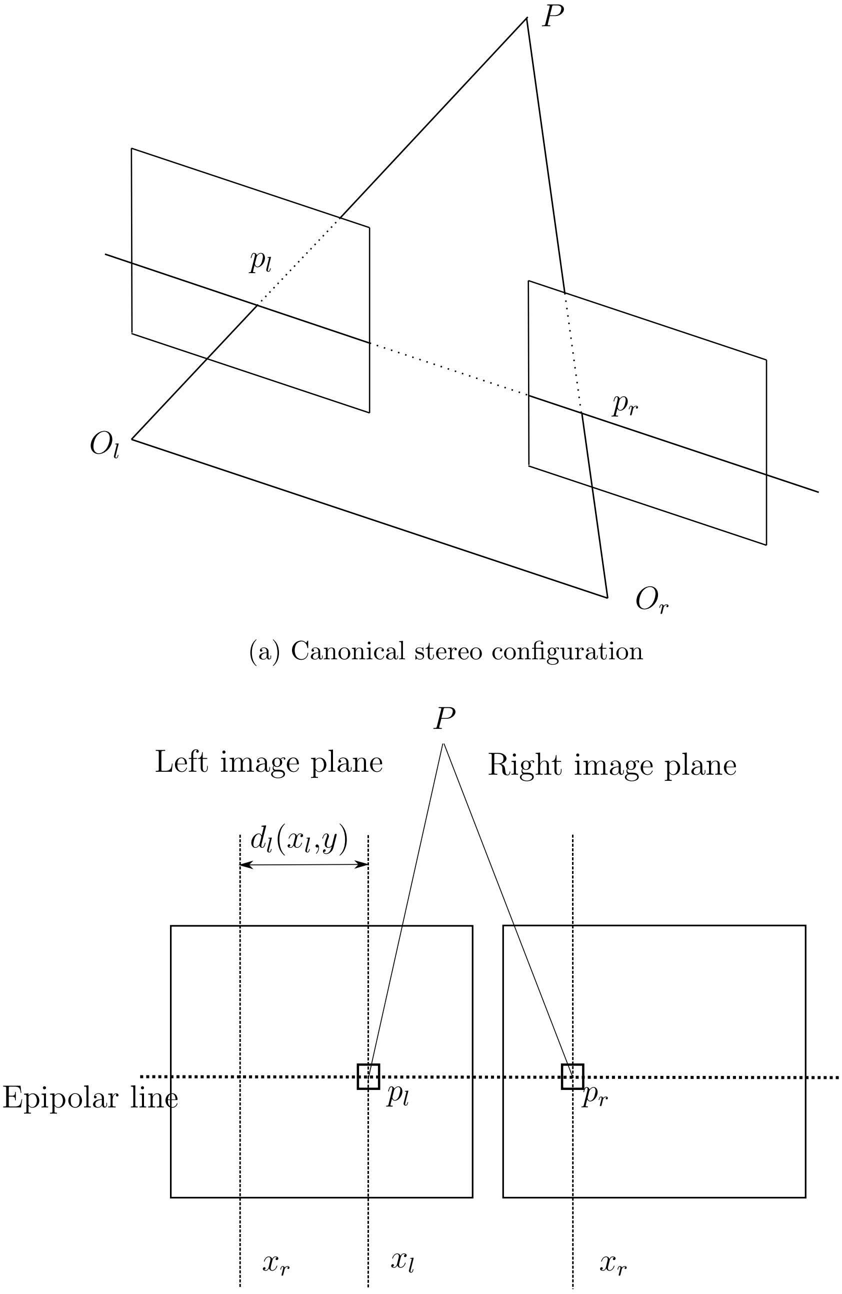
Some methods also add a preprocessing step, such as image segmentation, before computation of the matching cost [BG07]. We focus our analysis on the optimization step and split the description of dense stereo matching methods into local and global methods.

The optimization step of global methods usually involves a high computational effort which does not make them suitable for real-time applications. Although global methods provide the best results at the current state-of-the-art [SS02, BBH03], the interest in local ones does not decrease due to their simplicity and low computational complexity.

Before introducing the local and global methods, we will present the canonical configuration of a stereo setup. The use of canonical configuration simplifies the definition of neighborhoods and matching costs for local methods. Then, we describe the local and global methods.

1.5.1 Canonical configuration

Most binocular stereovision systems are based on the canonical configuration of cameras. To obtain a canonical configuration geometry, we set up two identical cameras according to the configuration shown in figure 1.9a. In this configuration, the two image planes l and r coincide. The baseline is parallel to the two image planes so that the epipolar geometry is simplified and the corresponding epipolar lines coincide. Hence, the homologous pixels p_l and p_r can then be found on the same straight line but at different locations.



(a) Canonical stereo configuration

(b) Disparity between two homologous pixels

Figure 1.9 : Canonical configuration, i.e. the two image planes coincide, the focal distances are equal and the baseline is parallel to the two image planes.

The spatial location of pixel p_r in the right image appears shifted to the left compared to that of the pixel p_l in the left image. This shift between the coordinates of homologous pixels is called disparity (see figure 1.9b). The disparity value $d_l(x_l,y)$ at p_l, whose coordinates are (x_l,y) in the left image plane, is denoted as :

$$d_l(x_l,y) = x_l - x_r, \tag{1.12}$$

where (x_r,y) are the coordinates of p_r in the right image plane. We use y instead of y_l and y_r in the coordinates of p_l and p_r since both pixels lie on the same horizontal line.

It can be easily proved that the disparity value is inversely proportional to the distance separating the scene point from the camera planes. Moreover, depending on the parameters of the stereo setup, the maximum possible disparity between the left and right images can be defined. The correspondence problem in this configuration can be reformulated as : given one projected pixel, we should find its disparity value.

Marr *et al.* [MP79] state that if the scene is separated into objects whose surface geometries are smooth enough, then the disparity varies smoothly almost everywhere, except at depth discontinuities. In other words, the disparity is a piecewise continuous function. This assumption, called smoothness assumption, is often used with assumptions explained in section 1.4.

Even when the canonical configuration in not used in the stereovision system, we can still benefit from this simple geometry by projecting back the image planes onto a plane that is parallel to the baseline as shown in figure 1.10. Each pixel p_l of the left projection plane is reprojected onto a virtual parallel left image plane l'. This new projected pixel lies at the intersection of O_lp_l with l'. Similarly, each pixel p_r in the right projection plane is reprojected onto a virtual parallel right image plane r'. Since the reprojected pixels may have non-integer coordinates, interpolation methods are required to estimate their gray-levels.

This process is known as rectification or epipolar rectification [FL01]. By rectifying the images, the corresponding epipolar lines coincide. Also, both computational complexity and the possibility of false matches are greatly reduced [Zha96]. Various methods for rectification have been proposed by [ATV00, Har99].

Throughout this thesis, we assume that the geometry of the stereovision setup is precisely adjusted so that the epipolar lines correspond to horizontal lines in the image planes. Therefore, each pair of homologous pixels lies in the same horizontal line.

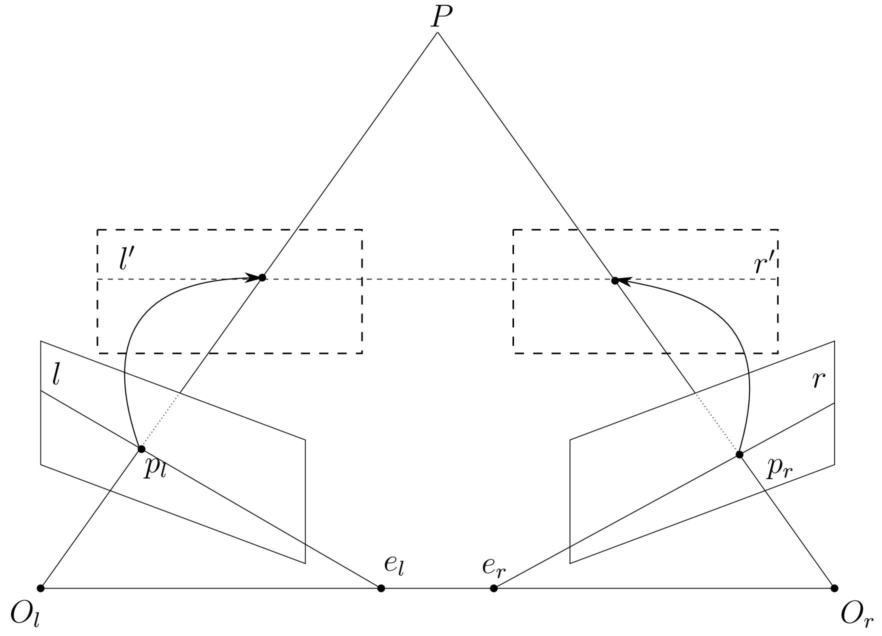


Figure 1.10 : Rectification process.

1.5.2 Local dense stereo matching methods

Local methods, also called window-based approaches, assume that the gray-levels configureation are similar in the neighborhood of homologous pixels. More particularly, they assume that the gray-levels of neighbors of a left pixel are close to those of the same neighbors of its homologous right pixel in the right image. So, a matching cost is defined between the window around the left pixel to be matched and the windows around the candidate right pixels in the corresponding epipolar line in the right image. This epipolar line is a horizontal image line in the case of canonical stereo configuration.

The window is shifted over all possible candidate pixels so that a matching cost between the left pixel and each candidate one in the right image is computed. The final disparity estimation is obtained by selecting the window with the lowest matching cost. Since local approaches match window areas, they are often called area-based or window-based methods.

Window-based approaches exploit the concept of a support region. Each pixel receives a support from its neighboring pixels. One assumes that pixels inside this support region are likely to have the same disparity and can therefore help to resolve matching ambiguities. Usually, rectangular or square windows centered on the pixel to be matched implement this concept.

Let us consider an example, the sum of squared differences matching cost. In this cost, the squared difference (SD) dissimilarity measure between gray-levels of pixels is used. The SD between the gray-level $I_l(x_l,y)$ of pixel p_l with coordinates (x_l,y) in the left image and the gray-level $I_r(x_l - s,y)$ of a candidate pixel p_r at a shift s with coordinates $(x_l - s,y)$ in the right image can be written as :

$$SD_g(x_l,y,s) = (I_l(x_l,y) - I_r(x_l - s,y))^2, \tag{1.13}$$

where the subscript g refers to gray-level images. The aggregation of the SD costs in the window of size $(2w + 1) \times (2w + 1)$ centered at p_l and a similar window centered at p_r is called sum of squared differences cost (SSD) and is defined for gray-level images as :

$$SSD_g^w(x_l,y,s) \quad = \quad \sum_{i=-w}^{w} \sum_{j=-w}^{w} (I_l(x_l + i,y + j) - I_r(x_l + i - s,y + j))^2, \tag{1.14}$$

where w is the window half-width.

We compute the SSD matching cost for all possible right candidates. Then, the homologous right pixel is derived based on the winner-takes-all principle (WTA) as illustrated in figure 1.11. The shift for which the matching cost is the lowest is selected. Thus, the estimated disparity $\hat{d}_l^w(x_l,y)$ at the pixel p_l corresponds to the shift s of the right pixel at which the matching cost is the lowest. It is expressed as :

$$\hat{d}_l^w(x_l,y) = \arg\min_s(SSD_g^w(x_l,y,s)). \tag{1.15}$$

We use the l subscript in the estimated disparity symbol since the left image pixel is used as the reference in the cost computation, s ranging from s_{min} to s_{max}. We also use the superscript w because the aggregation is based on a neighborhood window whose half-width is w.

Once the disparity has been estimated at each pixel in the left image, the left dense estimated disparity map is formed. The disparity map is the array of disparity values computed for each pixel, which has the same size $M \times N$ as the digital images.

When there are untextured areas in the images, the matching cost does not reach a global extremum at the correct disparity. Figure 1.12 shows an example of untextured area. In this figure, the gray-levels of the left and right images are represented. If we examine the gray-levels of nine pixels, we are not able to determine a correspondence

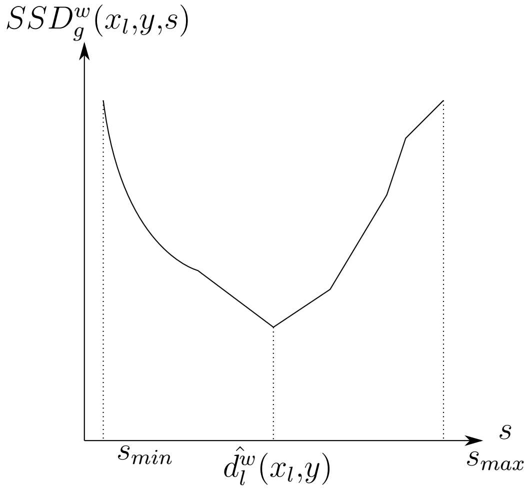
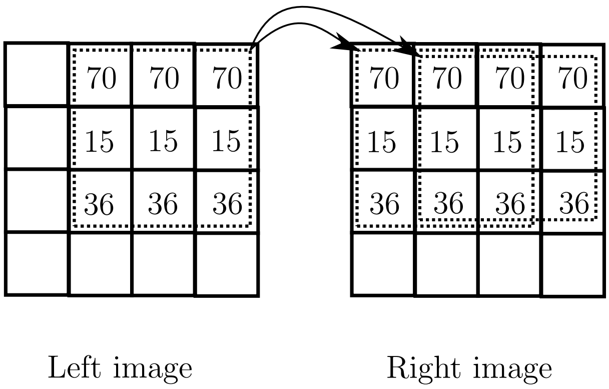


Figure 1.11 : Winner-takes-all principle, $[s_{min},s_{max}]$ represents the possible shifts of the searched pixel. At shift s equal to $\hat{d_l^w}(x_l,y)$ the matching cost value is minimum.



	70	70	70
	15	15	15
	36	36	36

70	70	70	70
15	15	15	15
36	36	36	36

Figure 1.12 : Matching untextured areas.

in the other image because the gray-level pattern is the same along the corresponding epipolar lines in the two images.

Another source of ambiguity arises from the presence of a repetitive pattern [Cha05]. To avoid this problem, the matching cost and the aggregation area should be carefully chosen to correctly match pixels.

1.5.2.1 Matching costs

To identify the right pixel which is homologous to a considered left pixel, a matching cost is computed for each candidate right pixel. The simplest matching costs assume that the gray-levels at homologous pixels are equal. More robust costs model explicitly or implicitly certain radiometric changes and/or noise [HS07]. Common pixel-based matching costs include absolute differences, squared differences, sampling-insensitive absolute differences [BT98], or truncated versions of these. Since costs are used by window-based stereo methods, they are usually defined for a given window shape.

Some costs are insensitive to differences in camera gain or bias, for example gradient-based measures [Sch94] and non-parametric measures such as rank and census transforms [ZW94]. More robust similarity measures have been proposed, such as mutual information but they require more computation time [Rey00, Hir05].

In this section, we present the most used window-based matching costs. For a detailed review about matching cost and its sensitivity to radiometric differences, one can read [HS07].

The Sum of Absolute Differences (SAD), the Sum of Squared Differences (SSD) and Normalized Cross Correlation (NCC) are the most used costs by stereo matching methods. SSD is defined by equation (1.14) while the SAD cost for gray-level images can be expressed as :

$$SAD_g^w(x_l,y,s) \;=\; \sum_{i=-w}^{w}\sum_{j=-w}^{w} |I_l(x_l+i,y+j) - I_r(x_l+i-s,y+j)|. \tag{1.16}$$

The NCC matching for gray-level images is defined as :

$$NCC_g^w(x_l,y,s) \;=\; \frac{\sum_{i=-w}^{w}\sum_{j=-w}^{w} |I_l(x_l+i,y+j) \times I_r(x_l+i-s,y+j)|}{NormC_{lg}^w(x_l,y) \times NormC_{rg}^w(x_l-s,y)}, \tag{1.17}$$

where $NormC_{lg}^w(x_l,y)$ and $NormC_{rg}^w(x_l-s,y)$ are normalization coefficients of gray-levels corresponding to pixels of a $(2w+1)\times(2w+1)$ aggregation window, centered respectively on (x_l,y) in the left image and on (x_l-s,y) in the right image. The normalization coefficients are expressed as :

$$NormC_{lg}^w(x_l,y) \;=\; \sqrt{\sum_{i=-w}^{w}\sum_{j=-w}^{w} I_l(x_l+i,y+j)^2}, \tag{1.18}$$

and

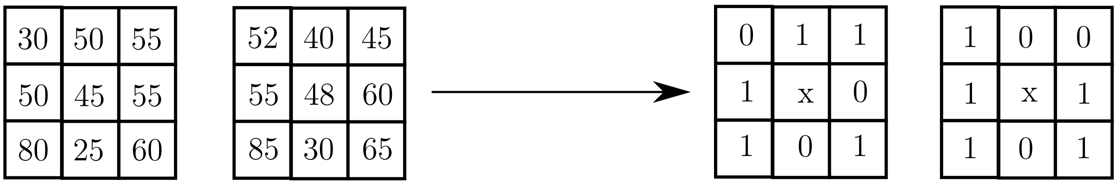
$$NormC_{rg}^w(x_l-s,y) \;=\; \sqrt{\sum_{i=-w}^{w}\sum_{j=-w}^{w} I_r(x_l+i-s,y+j)^2}, \tag{1.19}$$

where the subscripts lg and rg refer to the left and right gray-level images, respectively.

Birchfield and Tomasi have proposed a matching cost based on absolute difference which is insensitive to image sampling [BT98]. Rather than just comparing gray-levels of pixels shifted by integral amounts, which may miss a valid match, they compare the gray-levels of each pixel in the left image with a linearly interpolated function of the right image.

Zabih and Woodfill [ZW94] have introduced non-parametric measures. They propose to use ordering information among the gray-levels of pixels, rather than the gray-levels themselves. A non-linear transform as the rank or census transform is applied to the image, then an adequate matching cost is used. Matching costs using such transforms may improve performance near object boundaries [PC03].

The census transform is based on a non linear transform of gray-levels in order to compare them. It maps the local neighbors of a pixel p to a binary string, each bit representing one neighboring pixel. The bit associated to a neighboring pixel is set to one when its gray-level is higher than that of the central pixel, and set to zero otherwise. This representation is largely immune to photometric variations, or variations of camera gain or bias. Figure 1.13 shows the census transform process for two example windows, which are finally described respectively by the strings $0111x0101$ and $1001x1101$.



30	50	55
50	45	55
80	25	60

Left neighborhood

52	40	45
55	48	60
85	30	65

Right neighborhood

0	1	1
1	x	0
1	0	1

Left neighborhood

1	0	0
1	x	1
1	0	1

Right neighborhood

Figure 1.13 : Neighborhood census transform.

The strings of the census transformed windows are compared for similarity by using the Hamming distance, i.e. the number of bits that differ in the two bit strings. For example, the distance between the two windows of figure 1.13 is equal to 4. The best matching is determined by searching the minimum of the Hamming distance between the window centered on the left pixel and those of the right windows centered at the right candidate pixels.

1.5.2.2 Aggregation area

A straightforward aggregation approach consists in using a square window centered at each pixel p. This square-window approach implicitly assumes that the disparity is similar over all pixels in the square window. This assumption does not hold true near disparity discontinuities. To overcome this problem, several works propose different aggregation areas. First, the shiftable window approach is proposed, which considers multiple square windows centered at different locations. It retains the window with the smallest cost [BI99, FRT97].

For both square-window and shiftable-window approaches, the size of the support window is fixed and is difficult to adjust. A small window may not include sufficient gray-level variations for a good matching. A large one may violate the assumption of constant disparity inside the window. By analyzing the image, we deduce that we can associate each region representing the object surface with a window size. Thus, the window size should be large for areas with low textures and small for areas with fine details. For this reason, Kanade *et al.* propose an adaptive-window method which automatically selects the window size and/or shape based on local information [KO94].

The window size is initially defined by another solution proposed by Perez, and called SBAN (Similarity-Based Adaptive Neighborhood) [PBCC04]. In this method, only a few neighboring pixels that are supposed to correspond to the same object in the window are selected by a segmentation scheme for computing the matching cost.

The influence of the pixel location in the chosen window is also studied. Fusiello *et al.* use several windows for cost aggregation, where the location of the pixel to be matched varies from the center to the corners [FRT00].

They compute the SSD matching cost on nine rectangular windows in which the current pixel is located at different places, and keep the window with the smallest SSD. The idea is that a window yielding the smallest matching cost is the most likely to cover a constant depth region. The multiple window approach decreases errors at object boundaries.

Instead of keeping only one window as in [FRT00], Hirschmüller combines the costs computed with different windows [HIG02]. For example, when using a central window with 4 surrounding ones, the matching cost can be computed by adding the values of the two best surrounding windows to that of the middle one.

In these algorithms, the modification of window configuration is based on one or several indices which define the confidence granted to the matching. The support regions in all aggregation approaches detailed above are two-dimensional ones since one assumes that the disparity is constant over a support region. So, these approaches favor fronto-parallel surfaces.

In images of some natural scenes, this fronto-parallel plane assumption for which pixels have the same disparity, is regularly violated because of the perspective effect. Furthermore, complex objects cannot be accurately described with planes. In this case, disparity errors and a blurring effect across depth discontinuities appear in the resulting estimated disparity map. Moreover, the larger the size of the two-dimensional window, the higher the error rate is. That led Lefebvre *et al.* to assume that the information located on the epipolar line is sufficient to provide a good matching [LAC07]. For each pixel, matching costs are computed for several widths of one dimensional windows and several locations of the current pixel.

Another one dimensional approach to handling slanted surfaces is the oriented-rod approach [KLCL05]. This approach aggregates costs along a group of line segments with different orientations and retains the segment that gives the smallest matching cost.

1.5.3 Global dense stereo matching methods

In this section, we present the principle of global methods. In these methods, a global cost function is evaluated between the left and right images. The estimated disparity map computed by a global method is the map that minimizes the global cost.

The global cost function, denoted E_{Global}, is composed of two cost functions :

– the correspondence cost $E_{correspondence}$;

– the assumption cost $E_{assumptions}$.

The correspondence cost is expressed as for the local methods in terms of matching costs applied to a chosen aggregation area. Generally, global methods make use of pixel-based matching costs.

The assumption cost depends on a combination of the stereo correspondence assumptions presented in section 1.4. Once the global cost has been defined, a variety of optimization methods can be used to find its extremum. These optimization methods include dynamic programming, neural networks, genetic algorithms, graph cuts, etc.

As stated above, global methods generally make use of all the pixels in the image [BBH03]. However, some authors propose to reduce the global cost computation to a single epipolar line [FKO$^+$04] or to several adjacent epipolar lines [GY05].

In the next section, we present global methods that make use of one-dimensional smoothness assumption and two-dimensional smoothness assumption.

1.5.3.1 Horizontal disparity smoothness assumption

Some global methods assume that the disparity smoothly varies along horizontal lines. In this case, it is not necessary to define the global cost over all the image. In other words, it is sufficient to define a global cost between two corresponding epipolar lines in both images. Using this assumption, the problem of cost minimization can be transformed to a minimum path search problem.

In this approach, the optimal path is searched in a two-dimensional matching array representing all the possible matches. An example of a matching array is presented in figure 1.14. Each cell in the two-dimensional matching array corresponds to a potential match of two pixels. A lot of matches are excluded thanks to the order and maximum allowed disparity assumptions.

For example, the order assumption forbids matching left pixels with x_l coordinates to right pixels with x_r coordinates larger than x_l. The cells of the matching array corresponding to these forbidden matches are above the diagonal (marked in gray in figure 1.14). Imposing a constraint on the maximum acceptable value of the disparity allows the removal of cells in the lower right corner of the matching array.

A local cost, such as SSD or SAD, is associated to each cell. The correspondence cost term $E_{correspondence}$ of a path is equal to the sum of the local costs of all the cells it crosses. The assumption cost $E_{assumptions}$, which represents the one-dimensional smoothness of disparity, is a linear sum of the differences between disparities estimated at neighboring pixels along the chosen path. Hence, the assumption cost penalizes disparity changes between adjacent pixels.

A cell that is connected with its predecessor by a horizontal or vertical link represents an occlusion. The task of a stereo matching based on one-dimensional smoothness assumption consists now in computing the path with minimum cost that connects the two opposite corners *start* and *end* of the matching array.

Global approaches based on this assumption are generally solved using the dynamic programming optimization method [GY07, CBR+07].

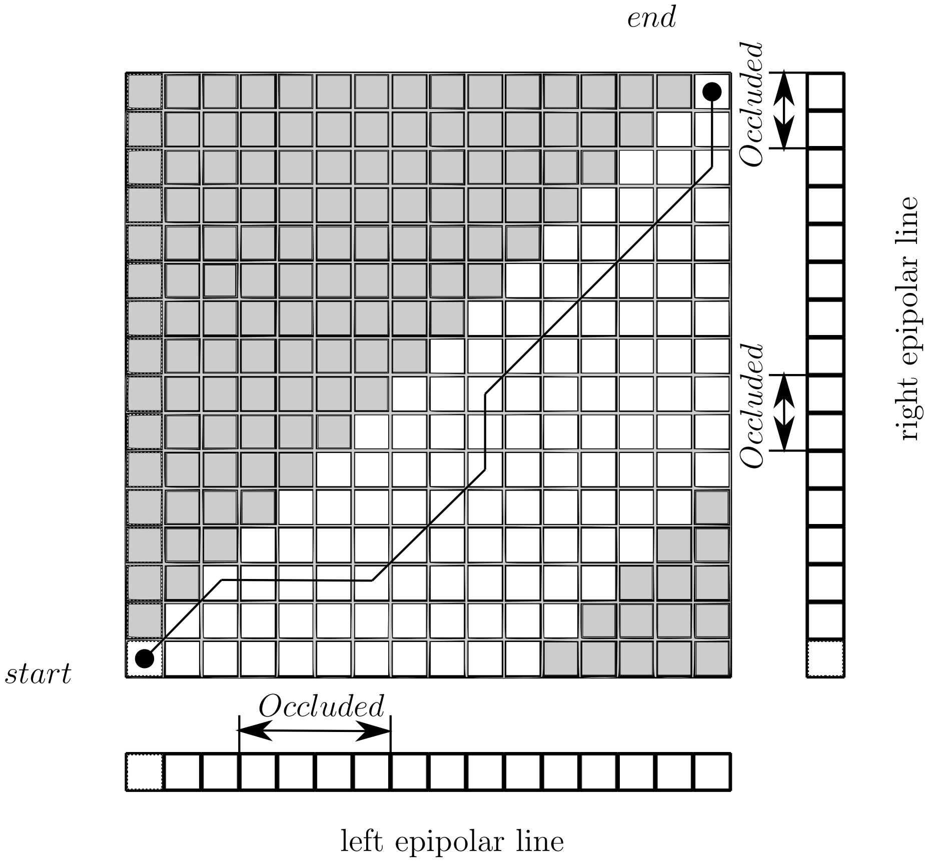


Figure 1.14 : Matching array used by a global stereo scheme based on one-dimensional smoothness assumption.

One major advantage of matching based on one-dimensional smoothness is to provide global support for regions that are weakly textured and which otherwise would be incorrectly matched . The computational cost of stereo matching algorithms based on this assumption and using dynamic programming are low compared to other global approaches imposing two-dimensional smoothness. However, a major disadvantage of this method is that a local error can be spread over the whole line and cause a lot of false pixel matches.

1.5.3.2 Horizontal and vertical disparity smoothness

This assumption imposes a two-dimensional smoothness of the estimated disparity map. Hence, the global cost should be defined over the whole image. This type of assumption has been proposed by Roy and Cox [RC98] and formalized by Veksler [Vek99] and Kolmogorov and Zabih [KZ01]. The best way to explain the global cost in this case is by using graph theory.

Nodes in the graph represent image pixels while the graph terminals represent all possible solutions, i.e. disparity affectation. Edges between nodes are weighted according to assumption cost term which is here the smoothness, while $E_{correspondence}$ is used to weight the edges between nodes and terminals. Finding a solution is possible by cutting the graph into many two-terminal graphs and determining the configuration with the lowest cost. Once each pixel is attached to one single terminal, it is assigned to that terminal's disparity value. In this case, graph cut optimization methods are the most suitable optimization techniques to find a solution to the global cost minimization problem.

At the current state-of-the-art in stereo matching, global approaches using this assumption give the most accurate experimental results. However, from a computational viewpoint, it is very time consuming and not suited to real time applications.

1.6 Left-right consistency check

Different methods allow the refinement of the obtained estimated disparity maps. They are used in the vast majority of stereo matching methods. These methods are used as a post-processing step to improve disparity maps by removing false matchings or for providing sub-pixel disparity estimation.

The most used method for detecting false matches is the left-right (LR) consistency check. In the matching process, one image is taken as a reference and for each pixel in this image (in the case of dense stereo matching), we seek its homologous in the other image. This approach may yield one estimated disparity map for each image. The first one based on the left image as reference and the second based on the right image as reference [FHZF93].

Then, the LR consistency check asserts that the pixels p_l and p_r are homologous if the following condition is verified :

$$\hat{d}_l(x_l,y) = \hat{d}_r(x_r,y) = \hat{d}_r(x_l - \hat{d}_l(x_l,y),y) \tag{1.20}$$

If this condition is not verified, then we consider that this position corresponds to a half-occlusion case and so, no disparity value can be estimated at these pixels [FRT00].

Throughout this thesis, we are not going to check the LR consistency. Since only the left estimated disparity map is computed, we will call it estimated disparity map, for simplicity. Hence, the left estimated disparity and the left actual disparity values will be denoted $\hat{d}$ and d, respectively, without the l subscript.

1.7 Real-time stereo correspondence methods

The constraint of time in stereo matching methods is very important for applications that run at video rate. So, the computation time has to be lower than the image acquisition time, which is generally 40ms. A lot of research about real-time stereovision follow two main axes :

- finding algorithms that do not need high computational time, while still providing good matching results ;
- developing hardware devices for efficient implementation in order to decrease the computation time.

These devices include special purpose hardware, such as digital signal processors (DSP) or field programmable gate arrays (FPGA), and extensions to recent PCs, such as the Multi-Media Extension (MMX) [FKO$^+$04, HIG02] and the pixel/vertex shading instructions for the Graphics Processing Units (GPUs) [GY05, MPL04].

Due to the complexity of global methods, most of them are limited to applications with non-real-time or near-real-time constraint. However, some global approaches can be implemented in real-time. Forstmann and Kanou have proposed a real-time stereo system based on dynamic programming [FKO$^+$04]. For efficient computation, the proposed system consists in a specific coarse to fine approach in combination with MMX implementation, to achieve real-time stereo matching. They reduce the research area and use compiler optimization strategies to reach real-time performance. Another algorithm based on dynamic programming for real-time stereo matching is proposed in [GY07]. The iterative best path tracing process used by traditional dynamic programming is replaced by a local minimum search process, making the algorithm suitable for parallel execution.

However, almost all real-time stereo algorithms use local WTA optimization. For rectangular aggregation windows of fixed size, a sliding window technique, as described by Faugeras *et al.* [FHZF93] can be employed to make the complexity linear with respect to the number of pixels and computed disparities. This gives rise to real-time implementations such as that of Point Grey Research [1]. For a detailed review about real-time stereo algorithms based on local methods, one can refer to [WGGY06].

1.8 Conclusion

In this chapter, the basics of stereovision have been presented. Then, we have detailed the stereo correspondence that consists in finding pairs of homologous pixels in the left and right images. We have studied the cases where stereo matching schemes fail, such as for untextured regions or half-occlusion. We have also detailed the most frequently used assumptions, such as smoothness, order and unicity.

We have described the local and global methods which are the two main classes of stereo matching methods. The performances of these methods have been compared by Scharstein *et al.* [SS02]. Local methods analyze local neighborhoods of pixels to identify pairs of homologous pixels whereas global ones analyze all the pixels. Although global methods give the best results in this taxonomy, the search for an optimal solution is very time consuming and very hard to be built in embedded systems. On the opposite, local methods based on WTA can easily be implemented in embedded systems using parallel architectures.

In our thesis, we use local methods based on WTA to calculate dense estimated disparity maps. Till now, we have limited our study to gray-level based stereo matching methods. In the next chapter we will see how color information can be used in stereovision. The different architectures of color cameras and their influence on pixel stereo matching performance will be also examined.

[1] http ://www.ai.sri.com/∼ konolige/svs/svm.htm

Chapitre 2

Color stereovision and demosaicing

Résumé

Dans le chapitre précédent, les principes de base de la stéréovision ont été introduits, tout comme les méthodes dont l'objectif est de résoudre le problème de correspondance à travers le voisinage de pixels. Cependant, nous nous sommes concentrés sur les techniques qui permettent de calculer les features à partir des niveaux de gris des pixels, c'est-à-dire des techniques qui peuvent s'appliquer seulement sur les images de niveau de gris.

Dans la première partie de ce chapitre, nous élargirons notre étude sur les méthodes de correspondance stéréovision qui traitent des paires d'images couleur. Dans ce but, nous détaillerons les quatre étapes suivies pour résoudre le problème de correspondance stéréo en prenant en compte les couleurs de pixels. La solution du problème de correspondance est basée sur une configuration et un modèle de caméras spécifiques. Ainsi, prendre en compte la couleur nécessite d'étudier comment les caméras couleur sont modelées.

Dans les caméras récentes, les images sont la plupart du temps acquises par un simple capteur associé à un color filter array CFA. Ce principe et les techniques de dematriçage qui donnent une image couleur à partir d'une image brute acquise par le capteur, seront décrits dans la seconde partie de ce chapitre. Cette description est basée sur une étude exhaustive du dématriçage effectuée par les membres de notre laboratoire. Puis, nous montrerons que puisque les images couleur interpolées à partir d'images brutes sont corrompues par des estimations artefacts, nous devrons les utiliser avec précaution pour résoudre le problème de correspondance stéréo.

2.1 Introduction

In the previous chapter, basic principles of stereovision have been introduced, as well as a review of methods which aim at solving the so-called correspondence problem through pixel, neighborhood, or feature matching. However, we stayed focused on techniques that compute features from pixel gray-levels, i.e. that can deal only with gray-level images. In the first part of this chapter, we extend the review to stereo matching methods that process pairs of color images. For this purpose, we detail how the four steps followed to solve a stereo correspondence problem take into account the color of pixels. We have also seen that solving the correspondence problem is based on a specific configuration and model of cameras. So, taking into account color requires to study how color cameras are modeled.

In recent cameras, images are most of the time acquired by a single sensor associated to a color filter array (CFA). This principle, and the demosaicing techniques that yield a color image from the raw image acquired by the sensor, are described in the second part of this chapter. This description is based on an exhaustive review of color demosaicing done by members in our laboratory [LMY10]. Finally, we show that since color images interpolated from raw images are corrupted by estimation artifacts, one should use them cautiously for solving the stereo correspondence problem.

2.2 Color stereo correspondence

Until the end of the 90's, color cameras were very expensive and using color images required a lot of processing power. Now, with the advances in camera technology and the increase of computational device speed, using color images has become more frequent in computer vision [CC04, OT92].

Many authors have reported that the use of color can highly improve the accuracy of stereo matching results [Cha05, CTB06, Kos96]. Indeed, color information can sometimes help remove matching ambiguities, notably in the case of metamerism [CTB06]. Anyway, it is clear that, in most situations, a *full color* image carries more information in its three chromatic components than a monochromatic image of the same scene.

Each pixel with coordinates (x,y) of a color image $\mathbf{I}$ is associated with a point in the three-dimensional RGB color space. The coordinates of this color point, denoted $R(x,y)$, $G(x,y)$ and $B(x,y)$ are the trichromatic components of the pixel. Therefore, a digital color image can be considered as an array of color points $\mathbf{I}(x,y) = (R(x,y),G(x,y),B(x,y))^T$. The color image $\mathbf{I}$ can be split into three *component planes* or *component images*, namely R, G and B. In each component image, a pixel is characterized by one single color component level.

In section 1.5 of chapter one, we stated that according to [SS02], most dense stereo matching methods intending to solve the correspondence problem can be decomposed into four steps. In the following, we present how color information can be exploited to improve the performance of each of these steps.

2.2.1 Color and cost computation

The generalization to digital color images of the sum of squared differences cost (SSD_g^w, equation (1.14)), computed between the left pixel with coordinates (x_l,y) and a candidate pixel in the right image, with s-shifted coordinates $(x_l - s,y)$, is expressed as :

$$SSD_c^w(x_l,y,s) \;=\; \sum_{i=-w}^{w}\sum_{j=-w}^{w} \|\mathbf{I}_l(x_l+i,y+j) - \mathbf{I}_r(x_l+i-s,y+j)\|^2 \;, \qquad (2.1)$$

where $\| \cdot \|$ is the euclidean norm. Therefore, $\| \,.\, - \,.\, \|^2$ is the squared euclidean distance between two points of the three dimensional RGB color space. In equation (2.1), $\mathbf{I}_l$ and $\mathbf{I}_r$ are the color points associated respectively with the left and right pixels, s is the spatial shift along the horizontal epipolar line, and w the half-width of a $(2w + 1) \times (2w + 1)$ aggregation window [Kos93].

Similarly, the gray-level version of the sum of absolute differences (SAD_g^w, equation (1.16)), can be generalized to deal with color images as follows :

$$\begin{aligned} SAD_c^w(x_l,y,s) \;=\; \sum_{i=-w}^{w}\sum_{j=-w}^{w} \Big(& |R_l(x_l+i,y+j) - R_r(x_l+i-s,y+j)| \\ & +|G_l(x_l+i,y+j) - G_r(x_l+i-s,y+j)| \\ & +|B_l(x_l+i,y+j) - B_r(x_l+i-s,y+j)| \Big) \;, \end{aligned} \qquad (2.2)$$

in which the squared euclidean norm used in SSD_c^w is in fact replaced by the L_1 norm.

The normalized cross-correlation cost (NCC_g^w, equation (1.17)) can be generalized to color images using the expression :

$$NCC_c^w(x_l,y,s) \quad = \quad \frac{\sum_{i=-w}^{w}\sum_{j=-w}^{w}|\mathbf{I}_l(x_l+i,y+j)\cdot\mathbf{I}_r(x_l+i-s,y+j)|}{NormC_{lc}^w(x_l,y)\times NormC_{rc}^w(x_l-s,y)}\ , \qquad (2.3)$$

where $NormC_{lc}^w(x_l,y)$ and $NormC_{rc}^w(x_l-s,y)$ are normalization coefficients of color points corresponding to pixels of a $(2w+1)\times(2w+1)$ window, centered respectively on (x_l,y) in the left image and on (x_l-s,y) in the right image :

$$NormC_{lc}^w(x_l,y) = \sqrt{\sum_{i=-w}^{w}\sum_{j=-w}^{w}\|\mathbf{I}_l(x_l+i,y+j)\|^2}\ , \qquad (2.4)$$

and

$$NormC_{rc}^w(x_l-s,y) = \sqrt{\sum_{i=-w}^{w}\sum_{j=-w}^{w}\|\mathbf{I}_r(x_l+i-s,y+j)\|^2}\ . \qquad (2.5)$$

Non-parametric costs, such as the census transform associated with the Hamming distance (see section 1.5.2.1), can easily be adapted to take into account color information. For instance, a census transform can be computed separately for each of the three channels R, G, and B. Then, in the case of a 3×3 window, the census string associated with a pixel is composed of 24 bits, instead of 8 bits for a gray-level image.

The matching costs described above use explicitly the RGB color space. However, the color components of a pixel can also be defined in other tri-chromatic color spaces. Chambon has tested the performance of matching techniques with three costs defined on nine color spaces, in order to determine which one gives the best results [Cha05]. According to this empirical work, $H_1H_2H_3$ and XYZ color spaces provide the best results on about 60% of the studied stereo pairs. However, since most color cameras acquire the raw color information through RGB filters, we will keep this specific but widely used color space in our study.

2.2.2 Color and cost aggregation

In section 1.5.2.2, we showed that one crucial problem posed by local matching methods was how to choose the shape and size of the cost aggregation window. One generally assumes that all the pixels in the aggregation window are equivalent, i.e. that they have the same *weight* in the expression used for aggregation. In the case of color images, some authors have proposed to weight the pixels in the aggregation window according to their color similarity and geometrical proximity with the central pixel [XWFS02, YK06, PCC05].

Yoon *et al.* measures the euclidean distance between the color point associated with the central pixel and those associated with its neighboring pixels in the aggregation window [YK06]. The weight of each neighboring pixel is then defined as a decreasing function of the color distance. Perez *et al.* proposes to include or not a pixel in the aggregation window if it belongs to the same color *class* as the central pixel [PCC05]. In fact, this technique corresponds to associating a binary weight to each neighbor pixel, determined by thresholding an implicit distance in the color space.

2.2.3 Color and optimization

In most of the papers dealing with color stereovision found in the literature, there is no dedicated optimization technique making use of color information. The optimization step is usually the WTA for local methods or graph cut and dynamic programming for global ones. However, some authors propose to split the color stereo correspondence problem into three gray-level stereo correspondence problems and then apply a fusion rule to compute the disparity map [BCP00, CC04].

Chambon *et al.* calculate a matching cost for each color channel and then apply the WTA selection technique to estimate three disparities for each pixel. If two or three of these disparities are equal, then this value is kept as the disparity estimation for the pixel. When this condition is not verified, several strategies are used to compute the final disparity estimation. For example, the minimum, maximum, average or weighted average of the three estimated disparities can be selected [BCP00].

2.2.4 Preprocessing

Color can also be used in the preprocessing step. In [BG07], color images are segmented into disjoint regions before computing matching costs, using a technique that yields over-segmented images. Then, a smoothness constraint is introduced to estimate disparities in each region. Region boundaries are supposed to coincide with depth discontinuities.

2.2.5 Conclusion

We have seen that gray-level based stereo matching methods can be modified to take color information into consideration. As far as sensor resolution and size are the same, the three components of a full color image contain more information than the single component of the equivalent gray-level image. Therefore, stereo matching methods that take advantage of this extra amount of information should yield a denser and/or more accurate disparity map.

However, this increase of performance is no more guaranteed if image resolutions are different, or if the quality of color images is lower than the quality of gray-level ones. In the following, we will see that, with some image sensors, color information is acquired to the detriment of resolution. In this case, one can no more consider that color based stereo matching is systematically more efficient than gray-level based stereo matching.

2.3 Color image acquisition and color demosaicing

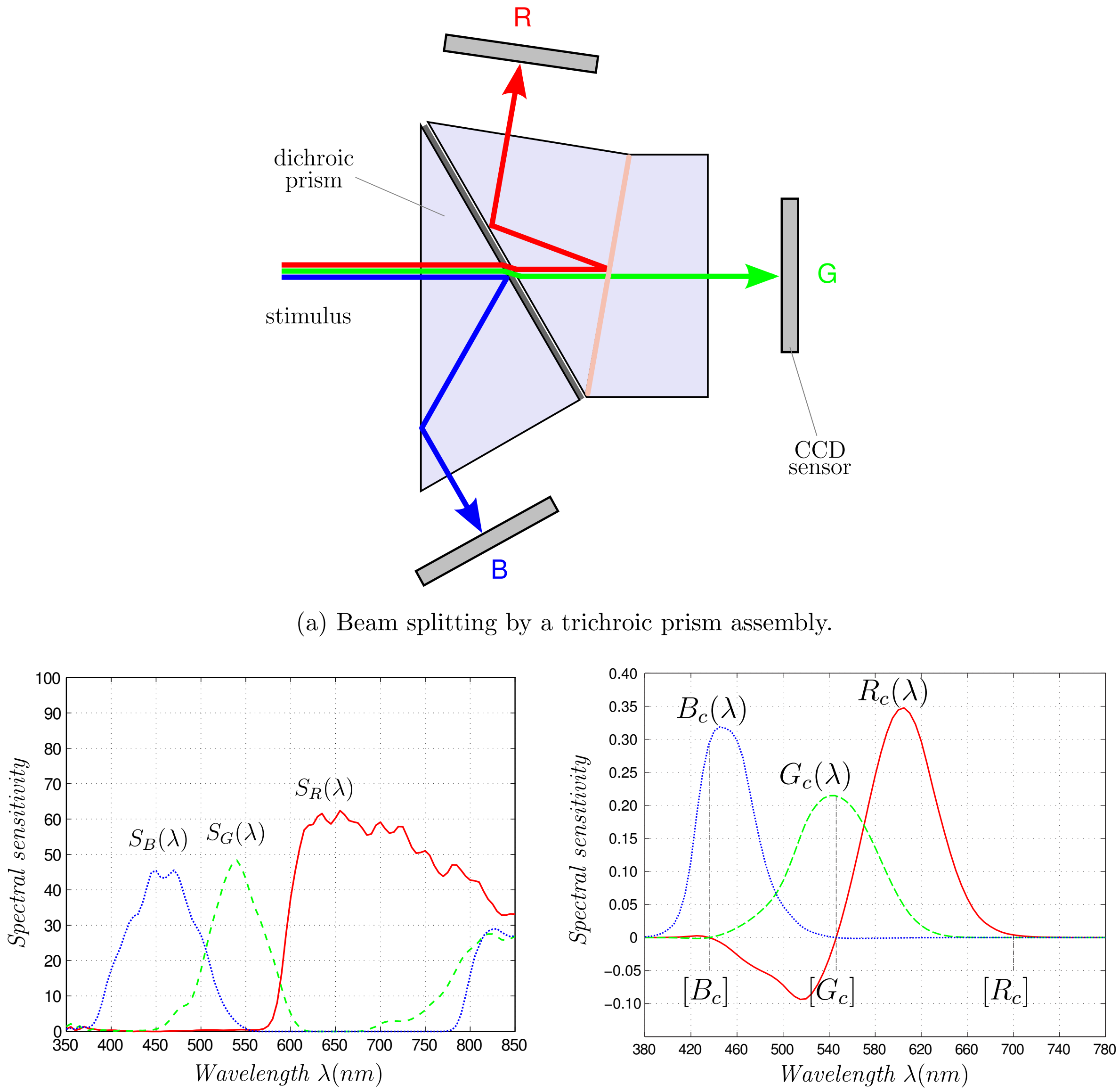
In this part, we describe the various image sensing and processing techniques used in recent color cameras. Then, we will show that the quality of estimated disparity maps strongly relies on the type of color camera used in the application.

Digital images or videos are currently a preeminent medium in environment perception. Acquisition techniques of color images in particular have involved much research work and undergone many changes. Despite major advancements, mass-market color cameras still often use a single sensor and require subsequent processing to deliver color images. This procedure, named *demosaicing*, is introduced in the following.

2.3.1 Single-CCD vs. three-CCD color cameras

Digital area scan cameras are devices able to convert color stimuli from the observed scene into a color digital image (or image sequence) thanks to photosensors. Such an output image is spatially digitized, i.e. composed of pixels. With each pixel is generally associated a single photosensor element, which captures the incident light intensity of the color stimulus.

The two main technology families available for the design of digital camera photosensors are CCD (*Charge-Coupled Device*) and CMOS (*Complementary Metal-Oxide Semiconductor*) technologies, the former being the most widespread one today. The CCD technology uses the photoelectric effect of silicon substrate, while CMOS is based on a photodetector and an active amplifier. Both photosensors overall convert the intensity of light reaching each pixel into a proportional voltage. Additional circuits then convert this analog voltage signal into digital data. For illustration and explanation purposes, the following text relates to the CCD technology.



(a) Beam splitting by a trichroic prism assembly.

(b) Relative spectral sensitivity of the Kodak KLI-2113 sensor.

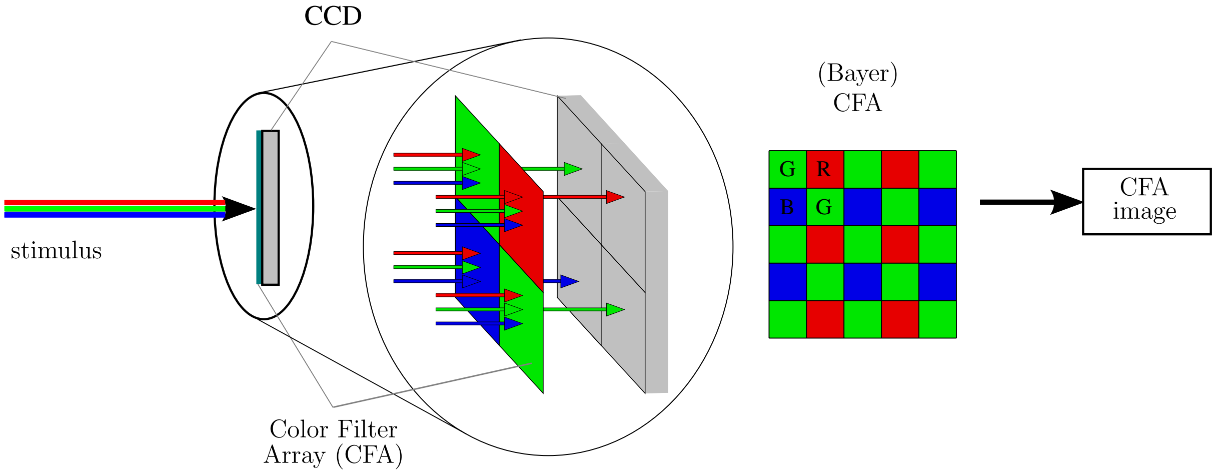
(c) CIE 1931 RGB color matching functions. $[R_c]$, $[G_c]$ and $[B_c]$ are the monochromatic primary colors.

Figure 2.1 : Three-CCD technology.

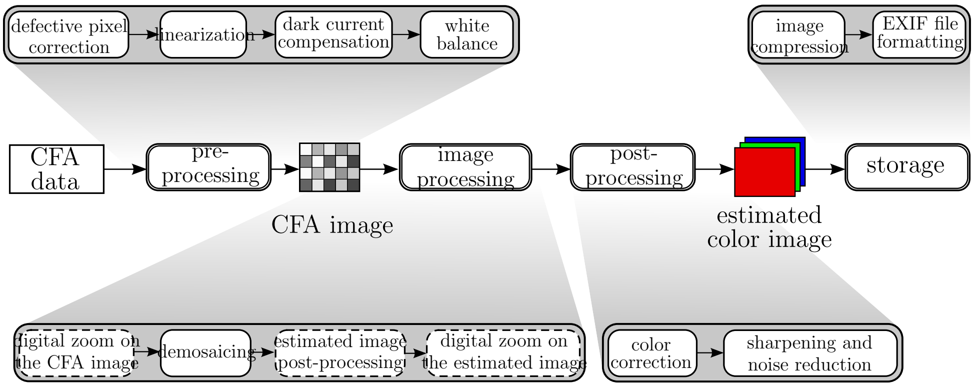
Various digital color cameras available on the market may also be distinguished according to whether they incorporate a single sensor or three sensors. In accordance with the trichromatic theory, three-CCD technology incorporates three CCD sensors, each one being dedicated to a specific primary color. In most devices, the color stimulus from the observed scene is split onto the three sensors by means of a trichroic prism assembly, made of two dichroic prisms (see figure 2.1a) [Lyo00]. Alternately, the incident beam may be dispatched on three sensors, each one being covered with a spectrally selective filter. The three component images R, G and B are simultaneously acquired by the three CCD sensors, and their combination leads to the final color image. Each digital three-CCD camera is characterized by its own spectral sensitivity functions $S_R(\lambda)$, $S_G(\lambda)$ and $S_B(\lambda)$ (see figure 2.1b for an example), which differ from the International Commission on Illumination (CIE) color matching functions $R_c(\lambda)$, $G_c(\lambda)$ and $B_c(\lambda)$ of the standard observer (see figure 2.1b).

Although three-CCD technologies yield high quality images, manufacturing costs of the sensor itself and of associated optical devices are high. Sensor size in such cameras is also limited by mechanical constraints. Calibrating the sensors of two cameras in order to reach the canonical configuration for stereovision (see figure 1.9a) is also very delicate. It requires a perfect parallelism of the three pairs of image planes, i.e. condition for epipolar lines to become parallel to horizontal lines.

In order to overcome cost constraints, a technology using a single sensor has been developed. The solution suggested by Bayer from the Kodak company in 1976 [Bay76] is still the most widely used today in commercial digital cameras. It uses a CCD or CMOS sensor covered by a filter (*Color Filter Array*, or CFA) designed as a mosaic of spectrally selective color filters, each of them being sensitive to a specific wavelength range. At each element of the CCD sensor, only one out of the three color components Red (R), Green (G) or Blue (B) is sampled (see figure 2.2a). Consequently, only one color component is available at each pixel of the image provided by the CCD charge transfer circuitry. This image is often related to as the *raw* image, but color filter array image (*CFA image*) is preferred hereafter in our specific context. In order to obtain a color image from the latter, two missing levels must be estimated at each pixel by a *demosaicing* algorithm (sometimes spelled *demosaicking*).



(a) Single-CCD technology outline, using the Bayer Color Filter Array (CFA).



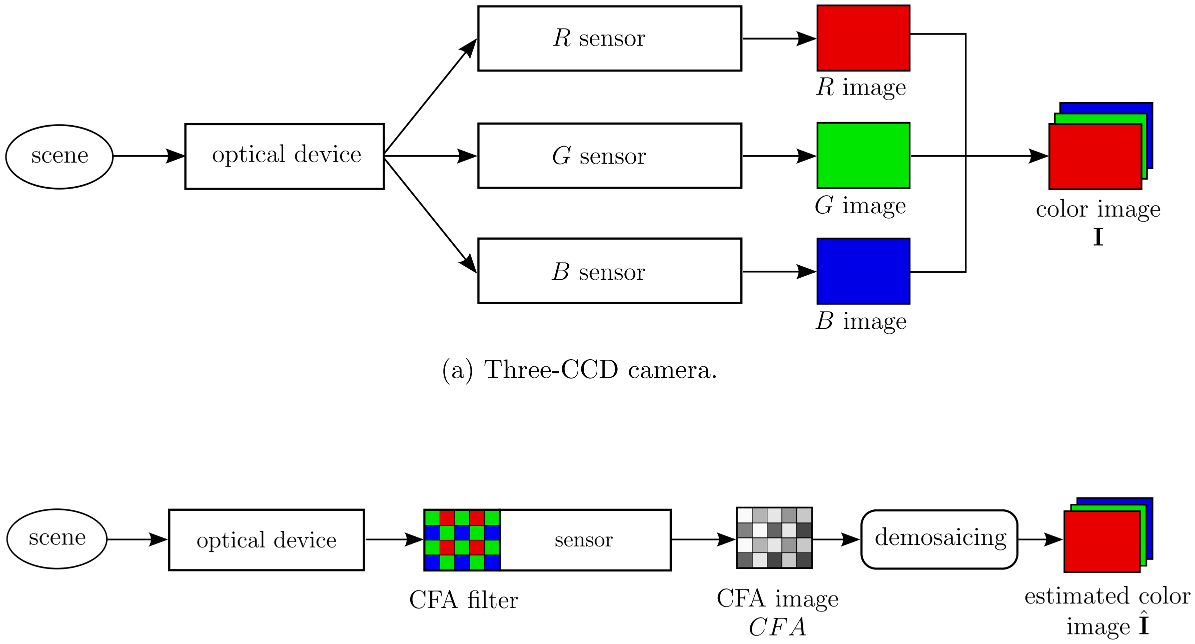
(b) Image acquisition within a single-CCD color camera (detailed schema). Dotted steps are optional.

Figure 2.2 : Internal structure of a single-CCD color camera.

A very large majority of cameras are equipped with Bayer filter arrays based on R, G and B primary color components. The Bayer array includes twice as many filters sensitive to the green primary as filters sensitive to blue or red color. This stems from Bayer's observation that the human eye has a greater resolving power for green color. Moreover, the photopic *luminous efficiency function* of the human retina – also known as the *luminosity function* – is similar to the CIE 1931 green matching function $C_G(\lambda)$, with a maximum reached in the same spectral domain. Bayer therefore both makes the assumption that green photosensors capture luminance, whereas red and blue ones capture chrominance, and suggests to fill the CFA with more luminance-sensitive (green) elements than chrominance-sensitive (red and blue) elements.

As shown in figure 2.2b, many other processing tasks are classically achieved within a single-CCD color camera [LP07]. They consist for instance in raw sensor data correction or, after demosaicing, in color improvement, image sharpening and noise reduction, so as to provide a “visually pleasing” color image to the user. These processing tasks are essential to the quality of the provided image and, as a matter of fact, discriminate the various models of digital cameras, since manufacturers and models of sensors are not so numerous. The related underlying algorithms have common features or basis, and parameter tuning is often a key step leading to more or fewer residual errors. Together with noise characteristics of the imaging sensor, such demosaicing schemes may incidentally be used to typify each camera model [BSM08].

2.3.2 Demosaicing formalization



(a) Three-CCD camera.

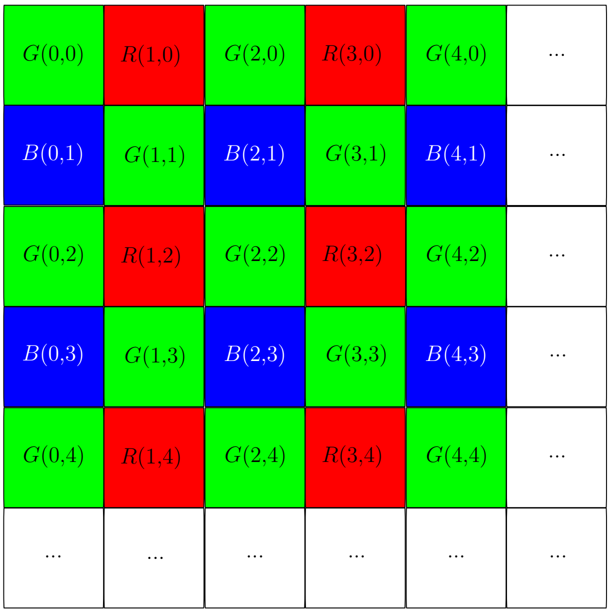
(b) Single-CCD color camera.

Figure 2.3 : Color image acquisition outline, according to the camera type.

In order to set a formalism for the demosaicing process, let us compare the acquisition process of a color image in a three-CCD camera and in a single-CCD camera. Figure 2.3a outlines the three-CCD camera architecture, in which the color image of a scene is formed by combining data from the three sensors. The resulting color image $\mathbf{I}$ is composed of three color component planes R, G and B.

In a color single-CCD camera, the color image generation is quite different, as shown in figure 2.3b : the single sensor delivers a CFA image denoted CFA. If the Bayer CFA is considered, with each pixel of coordinates (x,y) in CFA is associated a single color component $R(x,y)$, $G(x,y)$ or $B(x,y)$ (see figure 2.4) :

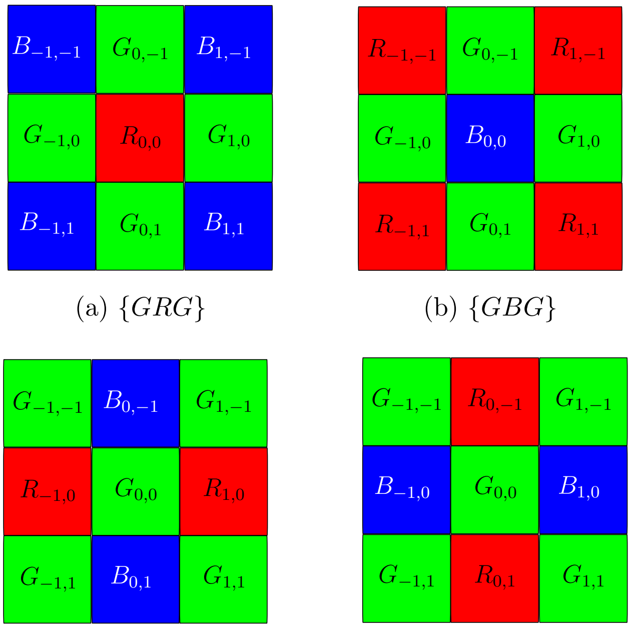
$$CFA(x,y) = \left\{ \begin{array}{lll} R(x,y) & \text{if } x \text{ is odd and } y \text{ is even,} & \quad (2.6a) \\ B(x,y) & \text{if } x \text{ is even and } y \text{ is odd,} & \quad (2.6b) \\ G(x,y) & \text{otherwise.} & \quad (2.6c) \end{array} \right.$$



$G(0,0)$	$R(1,0)$	$G(2,0)$	$R(3,0)$	$G(4,0)$	...
$B(0,1)$	$G(1,1)$	$B(2,1)$	$G(3,1)$	$B(4,1)$	...
$G(0,2)$	$R(1,2)$	$G(2,2)$	$R(3,2)$	$G(4,2)$	...
$B(0,3)$	$G(1,3)$	$B(2,3)$	$G(3,3)$	$B(4,3)$	...
$G(0,4)$	$R(1,4)$	$G(2,4)$	$R(3,4)$	$G(4,4)$	...
...	...	...	...	...	...

Figure 2.4 : CFA image from the Bayer filter. Each pixel is artificially colorized with the corresponding filter main spectral sensitivity, and the presented arrangement is the most frequently encountered in the literature (i.e. G and R levels available for the first two row pixels).

The demosaicing scheme $\mathscr{F}$, most often implemented as an interpolation procedure, consists in estimating a color image $\hat{\mathbf{I}}$ from CFA. At each pixel of the estimated image, the color component available in the CFA image at the same pixel location is picked up,



(a) $\{GRG\}$ (b) $\{GBG\}$

(c) $\{RGR\}$ (d) $\{BGB\}$

Figure 2.5 : 3 × 3 neighborhood structures of pixels in the CFA image.

whereas the other two components are estimated :

$$\hat{\mathbf{I}}(x,y) = \left\{ \begin{array}{lll} (R(x,y),\hat{G}(x,y),\hat{B}(x,y))^T & \text{if } x \text{ is odd and } y \text{ is even,} & (2.7a) \\ (\hat{R}(x,y),\hat{G}(x,y), B(x,y))^T & \text{if } x \text{ is even and } y \text{ is odd,} & (2.7b) \\ (\hat{R}(x,y), G(x,y), \hat{B}(x,y))^T & \text{otherwise.} & (2.7c) \end{array} \right.$$

Each triplet in equations (2.7) stands for a color point, whose color component available at pixel with coordinates (x,y) in CFA is denoted $R(x,y)$, $G(x,y)$ or $B(x,y)$, and whose other two components among $\hat{R}(x,y)$, $\hat{G}(x,y)$ and $\hat{B}(x,y)$ are estimated for $\hat{\mathbf{I}}(x,y)$.

Before we get to the heart of the matter, let us still precise a few notations that will be most useful later in this section. In the CFA image (see figure 2.4), four different structures are encountered for the 3 × 3 spatial neighborhood, as shown on figure 2.5. For each of these structures, the pixel under consideration for demosaicing is the central one, at which the two missing color components should be estimated thanks to the available components and their levels at the neighboring pixels. Let us denote the aforementioned structures by the color components available on the middle row, namely $\{GRG\}$, $\{GBG\}$, $\{RGR\}$ and $\{BGB\}$. Notice that $\{GRG\}$ and $\{GBG\}$ are structurally similar, apart from the slight difference that components R and B are exchanged. Therefore, they can be analyzed in the same way, as can $\{RGR\}$ and $\{BGB\}$ structures. A generic notation is hence used in the following : the center pixel is considered having (0,0) coordinates, and its neighbors are referred to using their relative coordinates $(\delta x,\delta y)$. Whenever this notation bears no ambiguity, (0,0) coordinates are omitted.

2.3.3 Basic Schemes and Demosaicing Rules

The first solutions for demosaicing were proposed in the early eighties. They process each component plane separately and find the missing levels by applying linear interpolation on the available ones, in both main directions of the image plane. Such a bilinear interpolation is traditionally used to resize gray-level images [GB04]. Considering the $\{GRG\}$ structure, the missing blue and green component levels at the center pixel are respectively estimated by bilinear interpolation thanks to the following equations :

$$\hat{B} = \frac{1}{4}\left(B_{-1,-1} + B_{1,-1} + B_{-1,1} + B_{1,1}\right), \tag{2.8}$$

$$\hat{G} = \frac{1}{4}\left(G_{0,-1} + G_{-1,0} + G_{1,0} + G_{0,1}\right). \tag{2.9}$$

As for the $\{RGR\}$ structure, the missing red and blue component levels are estimated as follows :

$$\hat{R} = \frac{1}{2}\left(R_{-1,0} + R_{1,0}\right), \tag{2.10}$$

$$\hat{B} = \frac{1}{2}\left(B_{0,-1} + B_{0,1}\right). \tag{2.11}$$

Bilinear interpolation is easy to be implemented and not time consuming, but generates severe visible artifacts, as also shown in figure 2.6. The above scheme provides satisfying results in image areas with homogeneous colors, but many false colors in areas with spatial high frequencies – as for the fence bars in this extract.

One can state that two main conditions have to be verified so as to improve demosaicing results : spatial correlation and spectral correlation.

- Spectral correlation.

  For a natural image, Gunturk *et al.* show that the three color components are also strongly correlated [GAM02]. The authors apply a bidimensional filter built on a low-pass filter $h_0 = [1\ 2\ 1]/4$ and a high-pass one $h_1 = [1\ \ -2\ 1]/4$, so as to split each color component plane into four subbands, resulting from row and column

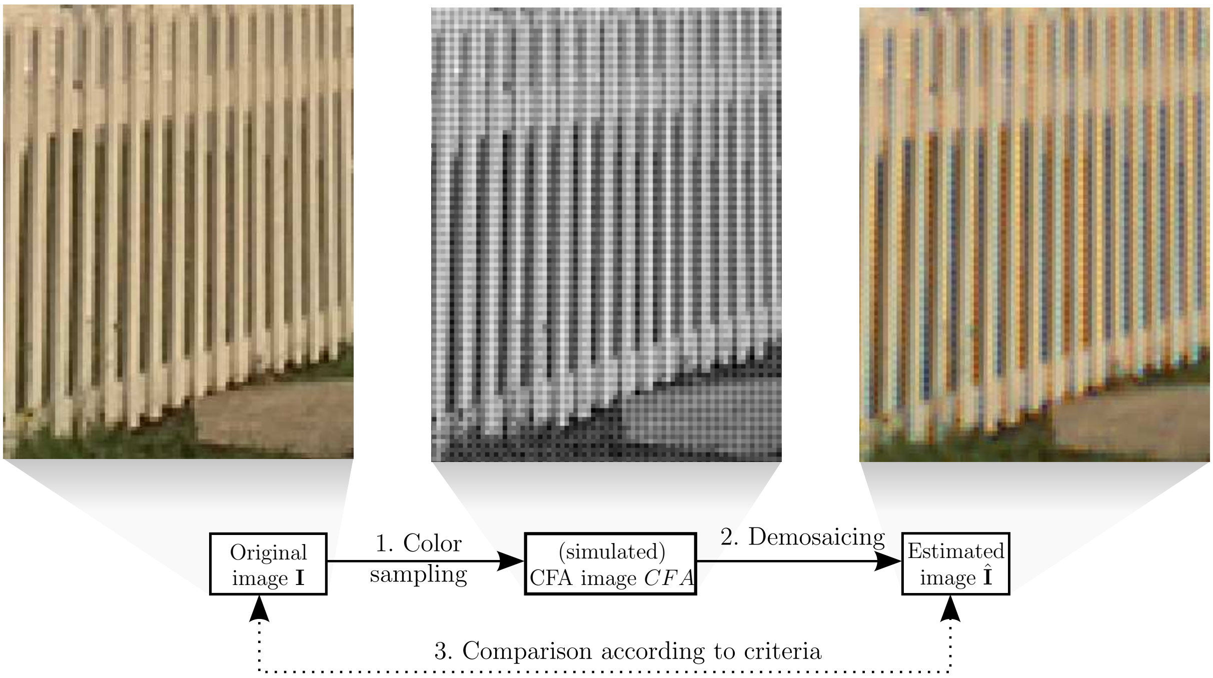

Original image **I**
1. Color sampling
(simulated) CFA image CFA
2. Demosaicing
Estimated image $\hat{\mathbf{I}}$
3. Comparison according to criteria

Figure 2.6 : Classical evaluation procedure for the demosaicing result quality (example of bilinear interpolation on an extract from the Kodak benchmark image "Lighthouse").

filtering : (LL) both rows and columns are low-pass filtered ; (LH) rows are low-pass and columns high-pass filtered ; (HL) rows are high-pass and columns low-pass filtered ; (HH) both rows and columns are high-pass filtered.

For each color component, four subband planes are obtained in this way, respectively representing data in rather homogeneous areas (low-frequency information), horizontal detail (high-frequency information in the horizontal direction), vertical detail (high-frequency information in the vertical direction) and diagonal detail (high-frequency information in both main directions). The authors then compute a correlation coefficient $r^{R,G}$ between red and green components over each subband according to the following formula :

$$r^{R,G} = \frac{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left(R(x,y)-\mu^R\right)\left(G(x,y)-\mu^G\right)}{\sqrt{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left(R(x,y)-\mu^R\right)^2}\sqrt{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left(G(x,y)-\mu^G\right)^2}}, \qquad (2.12)$$

in which $R(x,y)$ ($G(x,y)$, respectively) is the level at (x,y) pixel in the red (green, respectively) component plane within the same subband, μ^R and μ^G being the average of $R(x,y)$ and $G(x,y)$ levels over the same subband planes, M and N are the number of columns and rows, respectively.

The correlation coefficient between the blue and green components is similarly computed. Test results on twenty natural images show that those coefficients are always greater than 0.9 in subbands carrying spatial high frequencies at least in one direction (i.e. LH, HL and HH). As for the subband carrying low frequencies (LL), coefficients are lower but always greater than 0.8. This reveals a very strong correlation between levels of different color components in a natural image, especially in areas with high spatial frequencies. Using a wavelet coefficient analysis, Lian confirms that high-frequency information is not only strongly correlated between the three component planes, but almost identical [LCZT06]. Such *spectral correlation* between components should be taken into account to retrieve the missing components at a given pixel.

– Spatial correlation.

A color image can be viewed as a set of adjacent homogeneous regions whose pixels have similar levels for each color component. In order to estimate the missing levels at each considered pixel, one should therefore exploit the levels of neighboring pixels. However, this task is difficult at pixels near the boundary between two adjacent regions due to high local variation of color components. As far as demosaicing is concerned, this *spatial correlation* property avoids to interpolate missing components at a given pixel thanks to levels of neighboring pixels which do not belong to the same homogeneous region.

These two principles are generally taken into account sequentially by the demosaicing procedure. In the first step, demosaicing often consists in estimating the green component using spatial correlation. According to Bayer's assumption, the green component has denser available data within the CFA image, and represents the luminance of the image to be estimated. Estimation of red and blue components (assimilated to chrominance) is only achieved in a second step, thanks to the previously interpolated luminance and using the spectral correlation property. Such a way of using both correlations is used by a large number of methods in the literature. We also notice that, although red and blue component interpolation is performed after the green plane has been fully populated, spectral correlation is also often used in the first demosaicing step to improve the green plane estimation quality.

2.3.4 Demosaicing schemes

In this section, the most known and widely used demosaicing schemes proposed in the literature are described. We distinguish two main procedure families, according to whether they scan the image plane or chiefly use the frequency domain.

2.3.4.1 Edge-adaptive demosaicing methods

Estimating the green plane before R and B ones is mainly motivated by the double amount of G samples in the CFA image. A fully populated G component plane will subsequently make the R and B plane estimation more accurate. As a consequence, the G component estimation quality becomes critical in the overall demosaicing performance, since any error in the G plane estimation is propagated in the following chrominance estimation step. Important efforts are therefore devoted to improve the estimation quality of the green component plane – usually assimilated to luminance –, especially in high-frequency areas. Practically, when the considered pixel lies on an edge between two homogeneous areas, missing components should be estimated along the edge rather than across it. In other words, neighboring pixels to be taken into account for interpolation should not belong to distinct objects. When exploiting the spatial correlation, a key issue is to determine the edge direction from CFA samples.

Gradient computation is a general solution to edge direction selection. Hibbard's method [Hib95] uses horizontal and vertical gradients, computed at each pixel where the G component has to be estimated, in order to select the direction which provides the best green level estimation. Let us consider the $\{GRG\}$ CFA structure for instance (see figure 2.5a). Estimating the green level $\hat{G}$ at the center pixel is achieved in two successive steps :

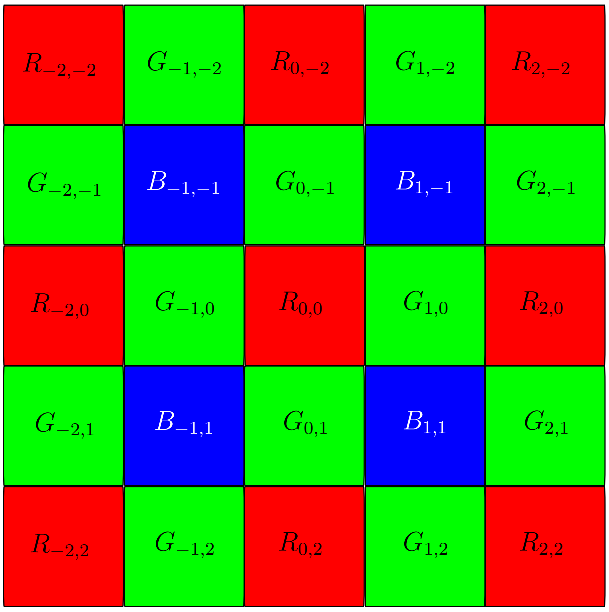
1. Approximate the gradient module (hereafter simply referred to as *gradient* for simplicity) according to horizontal and vertical directions, as :

$$\Delta^x = |G_{-1,0} - G_{1,0}|\,, \tag{2.13}$$

$$\Delta^y = |G_{0,-1} - G_{0,1}|\,. \tag{2.14}$$

2. Interpolate the green level as :

$$\hat{G} = \begin{cases} (G_{-1,0} + G_{1,0})/2 & \text{if } \Delta^x < \Delta^y \ , \quad (2.15a) \\ (G_{0,-1} + G_{0,1})/2 & \text{if } \Delta^x > \Delta^y \ , \quad (2.15b) \\ (G_{0,-1} + G_{-1,0} + G_{1,0} + G_{0,1})/4 & \text{if } \Delta^x = \Delta^y . \quad (2.15c) \end{cases}$$



$R_{-2,-2}$	$G_{-1,-2}$	$R_{0,-2}$	$G_{1,-2}$	$R_{2,-2}$
$G_{-2,-1}$	$B_{-1,-1}$	$G_{0,-1}$	$B_{1,-1}$	$G_{2,-1}$
$R_{-2,0}$	$G_{-1,0}$	$R_{0,0}$	$G_{1,0}$	$R_{2,0}$
$G_{-2,1}$	$B_{-1,1}$	$G_{0,1}$	$B_{1,1}$	$G_{2,1}$
$R_{-2,2}$	$G_{-1,2}$	$R_{0,2}$	$G_{1,2}$	$R_{2,2}$

Figure 2.7 : 5 × 5 neighborhood with central $\{GRG\}$ structure in the CFA image.

Laroche *et al.* suggest to consider a 5×5 neighborhood for partial derivative approximations thanks to available surrounding levels [LP93], for instance $\Delta^x = |2R - R_{-2,0} - R_{2,0}|$. Moreover, Hamilton and Adams combine both approaches [HA97]. To select the interpolation direction, these authors take into account both gradient and Laplacian second-order values by using the green levels available at nearby pixels and red (or blue) samples located 2 pixels apart. For instance, to estimate the green level at $\{GRG\}$ CFA structure (see figure 2.7), Hamilton and Adams use the following algorithm :

1. Approximate the horizontal Δ^x and vertical Δ^y gradients thanks to absolute differences as :

$$\Delta^x = |G_{-1,0} - G_{1,0}| + |2R - R_{-2,0} - R_{2,0}| \ , \quad (2.16)$$

$$\Delta^y = |G_{0,-1} - G_{0,1}| + |2R - R_{0,-2} - R_{0,2}| \ . \quad (2.17)$$

2. Interpolate the green level as :

$$\hat{G} = \begin{cases} \left(G_{-1,0} + G_{1,0}\right)/2 + \left(2R - R_{-2,0} - R_{2,0}\right)/4 & \text{if } \Delta^x < \Delta^y, \quad (2.18a) \\ \left(G_{0,-1} + G_{0,1}\right)/2 + \left(2R - R_{0,-2} - R_{0,2}\right)/4 & \text{if } \Delta^x > \Delta^y, \quad (2.18b) \\ \left(G_{0,-1} + G_{-1,0} + G_{1,0} + G_{0,1}\right)/4 & \\ \quad + \left(4R - R_{0,-2} - R_{-2,0} - R_{2,0} - R_{0,2}\right)/8 & \text{if } \Delta^x = \Delta^y. \quad (2.18c) \end{cases}$$

This proposal outperforms Hibbards' method. Indeed, precision is gained not only by combining two color component data in partial derivative approximations, but also by exploiting spectral correlation in the green plane estimation. We also notice that, in these equations, horizontal gradients are assumed to be similar for both red and blue components.

2.3.4.2 Demosaicing using the frequency domain

Some recent demosaicing schemes rely on a frequency analysis, by following an approach originated by [ASH05]. The fundamental principle is to use a frequency representation of the Bayer CFA image[1]. In the spatial frequency domain, such a CFA image may be represented as a combination of a luminance signal and two chrominance signals, all three being well localized. Appropriate frequency selection therefore allows one to estimate each of these signals, from which the demosaiced image can be retrieved. We notice that frequency-based approaches do not use Bayer's assumption that assimilates green levels to luminance, and blue and red levels to chrominance components.

A simplified derivation of Alleysson *et al.* 's approach has been proposed by [Dub05], whose formalism is retained here to present the general framework of frequency-domain representation of CFA images. Let us assume that, for each component of a color image, there exists an underlying signal f^k, $k \in \{R,G,B\}$. Demosaicing then consists in computing an estimation $\hat{f}^k$ coinciding with the estimated component at each pixel. Let us assume similarly that there exists a signal f^{CFA} which underlies the CFA image. This signal is referred to as *CFA signal* and coincides with CFA at each pixel. The CFA signal value at each pixel with coordinates (x,y) may be expressed as the sum of spatially

[1] Let us make here clear that frequency (i.e. *spatial* frequency), expressed in cycles per pixel, corresponds to the inverse number of adjacent pixels representing a given level series according to a particular direction in the image (classically, the horizontal or vertical direction).

sampled f^k signals :

$$f^{CFA}(x,y) = \sum_{k=R,G,B} f^k(x,y) m^k(x,y) \ , \tag{2.19}$$

where $m^k(x,y)$ is the sampling function for the color component k, $k \in \{R,G,B\}$. For the Bayer CFA of figure 2.4, this set of functions is defined as :

$$\begin{aligned} m^R(x,y) &= \frac{1}{4}\left(1-(-1)^x\right)\left(1+(-1)^y\right), && (2.20)\\ m^G(x,y) &= \frac{1}{2}\left(1+(-1)^{x+y}\right), && (2.21)\\ m^B(x,y) &= \frac{1}{4}\left(1+(-1)^x\right)\left(1-(-1)^y\right). && (2.22)\end{aligned}$$

With the definition $\begin{bmatrix} f^L \\ f^{C1} \\ f^{C2} \end{bmatrix} = \begin{bmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ -\frac{1}{4} & \frac{1}{2} & -\frac{1}{4} \\ -\frac{1}{4} & 0 & \frac{1}{4} \end{bmatrix} \begin{bmatrix} f^R \\ f^G \\ f^B \end{bmatrix}$, the expression of f^{CFA} becomes :

$$\begin{aligned} f^{CFA}(x,y) &= f^L(x,y) + f^{C1}(x,y)(-1)^{x+y} + f^{C2}(x,y)\left((-1)^x - (-1)^y\right) \\ &= f^L(x,y) + f^{C1}(x,y)e^{j2\pi(x+y)/2} + f^{C2}(x,y)\left(e^{j2\pi x/2} - e^{j2\pi y/2}\right). \quad (2.23)\end{aligned}$$

The CFA signal may therefore be interpreted as the sum of a luminance component f^L at baseband, a chrominance component f^{C1} modulated at spatial frequency (horizontal and vertical) (0.5,0.5), and of another chrominance component f^{C2} modulated at spatial frequencies (0.5,0) and (0,0.5). Such an interpretation can be easily checked on an achromatic image, in which $f^R = f^G = f^B$, the two chrominance components being equal to zero.

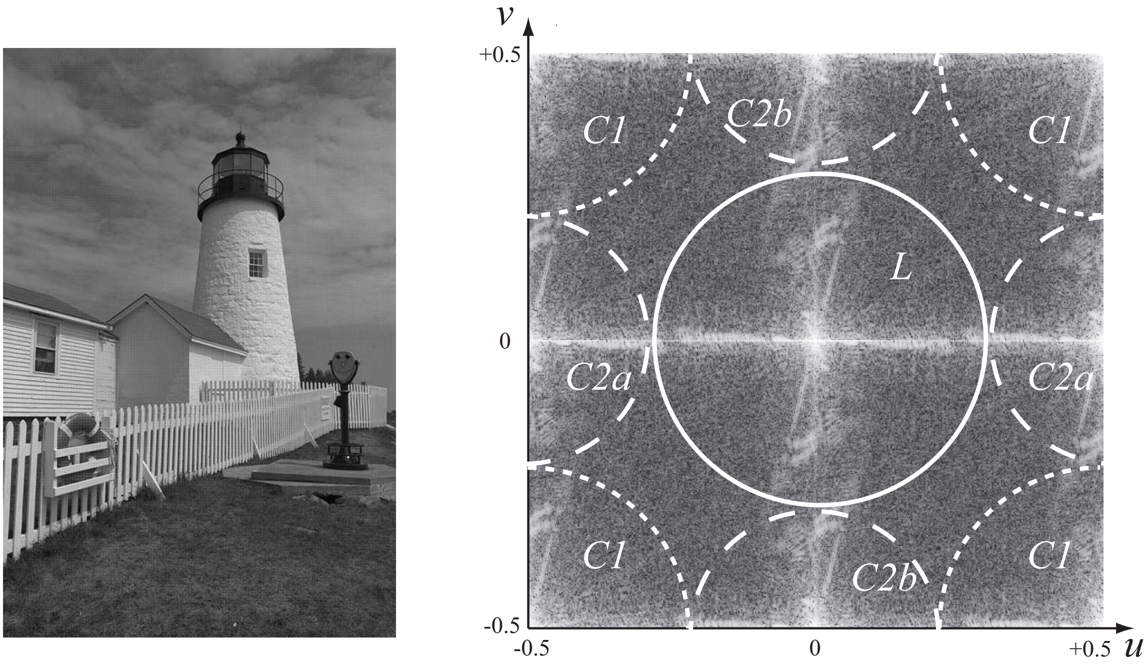
Provided that functions f^L, f^{C1} and f^{C2} can be estimated at each pixel from the CFA signal, estimated color levels $\hat{f}^R$, $\hat{f}^G$ and $\hat{f}^B$ are simply retrieved as :

$$\begin{bmatrix} \hat{f}^R \\ \hat{f}^G \\ \hat{f}^B \end{bmatrix} = \begin{bmatrix} 1 & -1 & -2 \\ 1 & 1 & 0 \\ 1 & -1 & 2 \end{bmatrix} \begin{bmatrix} \hat{f}^L \\ \hat{f}^{C1} \\ \hat{f}^{C2} \end{bmatrix}. \tag{2.24}$$

To achieve this, the authors take the Fourier transform of the CFA signal (equation (2.23)) :

$$F^{CFA}(u,v) = F^L(u,v) + F^{C1}(u-0.5,v-0.5) + F^{C2}(u-0.5,v) - F^{C2}(u,v-0.5), \tag{2.25}$$

expression in which terms are, respectively, the Fourier transforms of $f^L(x,y)$, of $f^{C1}(x,y)(-1)^{x+y}$, and of the two signals defined as $f^{C2a}(x,y) \triangleq f^{C2}(x,y)(-1)^x$ and $f^{C2b}(x,y) \triangleq -f^{C2}(x,y)(-1)^y$.



(a) "Lighthouse" CFA image.

(b) Normalized energy (frequencies in cycles/pixel).

Figure 2.8 : Localization of the energy (Fourier transform module) of a CFA signal in the frequency domain [ASH05].

It turns out that the energy of a CFA image is concentrated in nine areas of the frequency domain (see example of figure 2.8), centered on spatial frequencies according to equation (2.25) : energy of luminance $F^L(u,v)$ is mainly concentrated at the center of this domain (i.e. at low frequencies), whereas that of chrominance is located on its border (i.e. at high frequencies). More precisely, the energy of $F^{C1}(u-0.5,v-0.5)$ is located around diagonal areas ("corners" of the domain), that of $F^{C2}(u-0.5,v)$ along u axis of horizontal frequencies, and that of $F^{C2}(u,v-0.5)$ along v axis of vertical frequencies. These areas are quite distinct, so that isolating the corresponding frequency components is possible by means of appropriately designed filters. But their bandwidth should be carefully selected, since the spectra of the three functions overlap. In these frequency areas where luminance and chrominance cannot be properly separated, the *aliasing* phenomenon might occur and color artifacts be generated.

In order to design filter bandwidths which achieve the best possible separation of luminance (L) and chrominance ($C1$, $C2$), Dubois proposes an adaptive algorithm that mainly handles the spectral overlap between chrominance and high-frequency luminance components [Dub05]. The author observes that spectral overlap between luminance and chrominance chiefly occurs according to either the horizontal or the vertical axis. Hence he suggests to estimate f^{C2} by giving more weight to the sub-component of $C2$ ($C2a$ or $C2b$) that is least prone to spectral overlap with luminance. The implemented weight values are based on an estimation of the average directional energies, for which Gaussian filters (with standard deviation $\sigma = 3.5$ pixels and modulated at spatial frequencies (0,0.375) and (0.375,0) cycles per pixel) are applied to the CFA image.

2.4 Stereo matching of demosaiced color images

In the preceding sections, we detailed the main demosaicing methods used to estimate the color image from the CFA one. However, since the demosaicing methods intend to produce "visually pleasing" demosaiced color images, they attempt to reduce the presence of color artifacts, such as false colors or zipper effects, by filtering the images [YLD07]. So, the color texture information which is useful to match homologous pixels, is sometimes removed from the demosaiced color images.

Rather than displaying images, window-based stereo matching costs locally analyze texture information from estimated colors in order to search for homologous pixels. The efficiency of stereo matching costs computed on demosaiced color images may suffer from either color artifacts or absences of color texture caused by demosaicing schemes. In this section, we study the effect of demosaicing on the quality of pixel matching. We will show that applying standard pixel matching methods to demosaiced color images does not yield good results, and that one must consider developing dedicated techniques.

2.4.1 Demosaicing then matching

The straightforward method to match pixels in stereo image pairs acquired with single-CCD cameras is to first reconstruct a pair of color images by demosaicing. Once this pair of color images has been reconstructed, a standard color stereo matching algorithm can be applied. This procedure is called standard method and is summarized in figure 2.9.

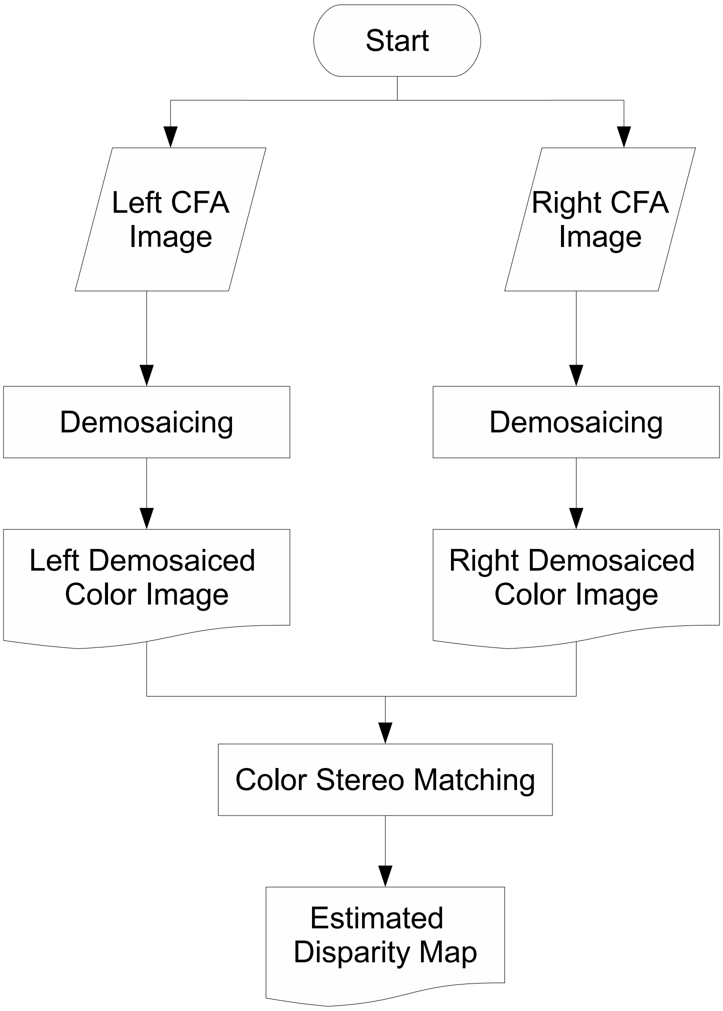


Figure 2.9 : Standard method flowchart.

Here, we use a local stereovision algorithm based on matching costs and winner-takes-all (WTA) method. A matching cost adapted to full color images, such as SSD_c^w (see equation (2.1)), SAD_c^w (see equation (2.2)) or NCC_c^w (see equation (2.3)), is evaluated between the left pixel with coordinates (x_l,y) and a candidate pixel in the right image, with the s-shifted coordinates $(x_l - s,y)$.

In order to avoid ambiguities, the matching costs computed on demosaiced color images will be denoted with a ST subscript (standing for *standard*) and therefore called standard costs since two of the three color components used in the cost expression are in fact estimations rather than measurements. Therefore, SSD_{ST}, SAD_{ST} and NCC_{ST} denote respectively the SSD, SAD and NCC matching costs computed on demosaiced color images, whereas SSD_c, SAD_c and NCC_c denote the same costs computed on full color images. When a generic notation of a matching cost will be required, we will use $COST_{ST}$ when the latter is computed on demosaiced color images and $COST_c$ when it is computed on full color images.

During the second step of the matching technique, costs computed for several right candidates, i.e. for several shifts s, are compared. With respect to the WTA method, the candidate pixel yielding the extremum cost is matched to the considered left pixel. The extremum is marked at the minimum of the costs in the case of SSD and SAD and at the maximum in the case of NCC.

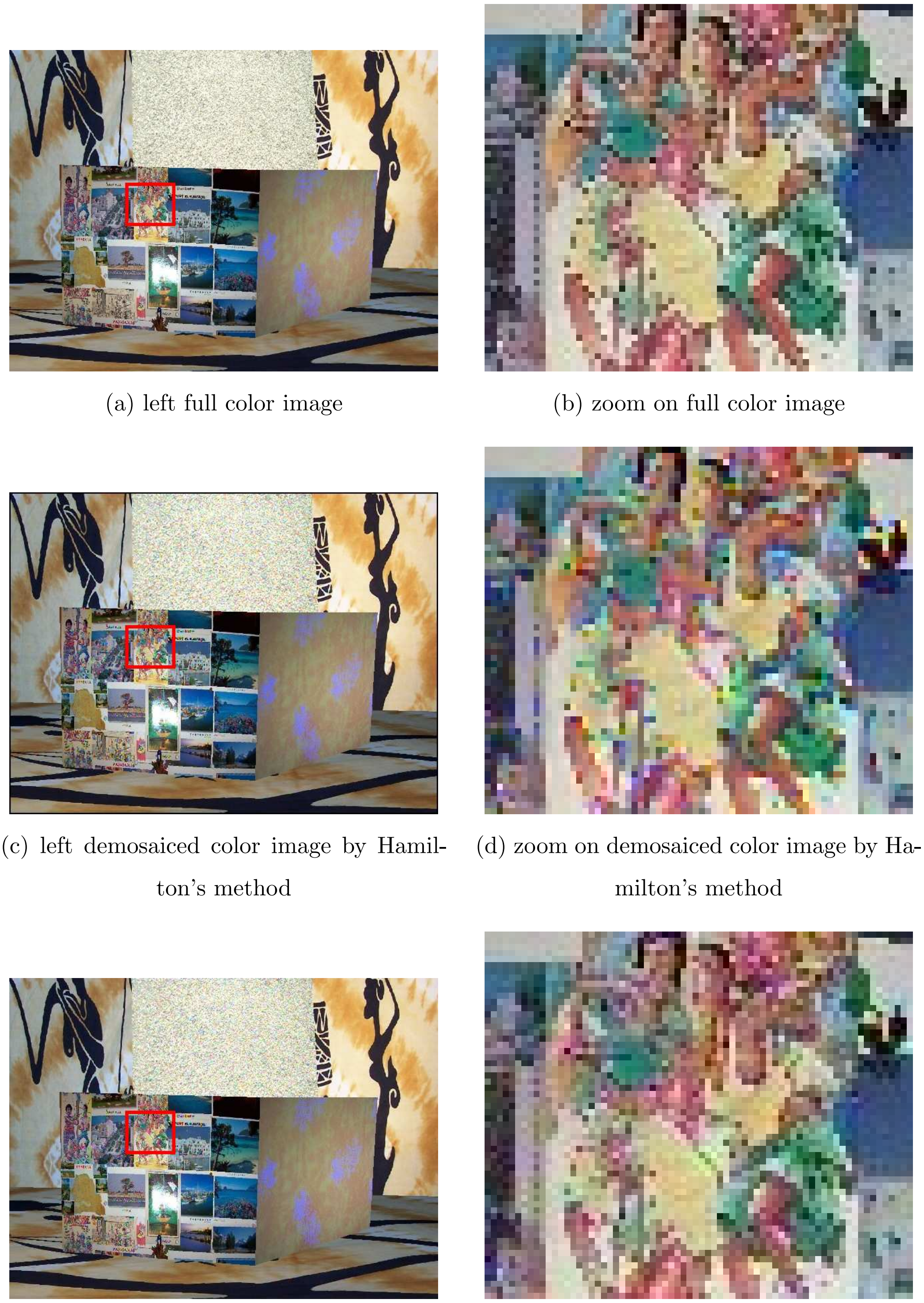
2.4.2 Experimental protocol

In order to experimentally show the limits of the matching results reached by applying different matching costs to a pair of demosaiced color images, we use 'Murs' full color stereo synthetic images designed by Bocquillon and available at http://www.irit.fr/~Benoit.Bocquillon/MYCVR/download.php, whose left disparity map is available. The disparity map contains the disparity value for each pixel, if it exists, or zero if it does not. Figure 2.10a shows the left image of the 'Murs' full color stereo image pair.

Our experimental procedure is decomposed into successive steps. First, in order to produce left and right CFA images from full color images, we simulate the CFA sampling process by keeping only one out of the three color components at every pixel. This is performed according to the spatial arrangement of the Bayer CFA (see figure 2.4). Then, the two missing color components at each pixel of the so-formed CFA images are estimated by either Hamilton's method [HA97] or Dubois' method [Dub05].

We have retained these two demosaicing schemes since Losson *et al.* have shown that Dubois' method provides the best performance with respect to different quality measurements and Hamilton's method reaches the best compromise between quality and computation time [YLD07].

The original left full color image (see figure 2.10a) and the left demosaiced color image demosaiced using Hamilton's method (see figure 2.10c) or Dubois' method (see figure 2.10e) look very similar. However, zooming on the square areas outlined in red in these three images (see figures 2.10b, 2.10d and 2.10f) shows that textured areas are



(a) left full color image

(b) zoom on full color image

(c) left demosaiced color image by Hamilton's method

(d) zoom on demosaiced color image by Hamilton's method

(e) left demosaiced color image by Dubois' method

(f) zoom on demosaiced color image by Dubois' method

Figure 2.10 : 'Murs' left image.

locally different : several artifacts have been generated by the demosaicing methods.

In a second step, we match all the pixels of the left image by computing the standard and color matching costs. By comparing the estimated disparity $\hat{d}(x_l,y)$ and the disparity $d(x_l,y)$, we estimate the rate of correctly matched pixels, denoted RCMP hereafter for simplicity. A matching is considered as valid when the absolute difference between the disparity and the estimated one is lower than or equal to δ, where δ is the disparity error tolerance. Since the disparity map is available with a sub-pixel precision whereas the estimations obtained with the used matching cost can reach only a pixel precision, δ is set to 0.5.

To capture the color by a single-sensor cameras, the bayer CFA is placed in front of the sensor. It is interesting to examine if putting a CFA inside the camera improves the RCMP reached by a stereovision scheme. When there is no CFA, pixels are characterized by gray-levels. From a full color image, the gray-level of a pixel with coordinates (x,y) is estimated as :

$$I(x,y) = \frac{R(x,y) + G(x,y) + B(x,y)}{3}. \tag{2.26}$$

Once the stereo gray-level images have been determined from the full color stereo images, we match all the pixels by gray-level matching costs denoted $COST_g$ (generic notation of equations (1.14), (1.16) and (1.17)). The RCMP obtained by analyzing gray-level images can be then compared with that obtained by analyzing demosaiced color images.

2.4.3 Experimental results

Figure 2.11 displays the RCMP obtained using three matching costs. In each of the six subfigures, the three curves represent the RCMP with respect to different aggregation window half-widths w. The black curve corresponds to the RCMP in the original full color image pair, using $COST_c$ for cost computation. The blue dotted curve corresponds to the RCMP in the gray-level image pair, using $COST_g$ for cost computation. Finally, the red dotted curve corresponds to the RCMP in the pair of demosaiced color images, using $COST_{ST}$ for cost computation. The left column shows results for Hamilton's demosaicing scheme and the right column for Dubois' scheme. The three lines correspond to the SSD, SAD and NCC matching costs, respectively.

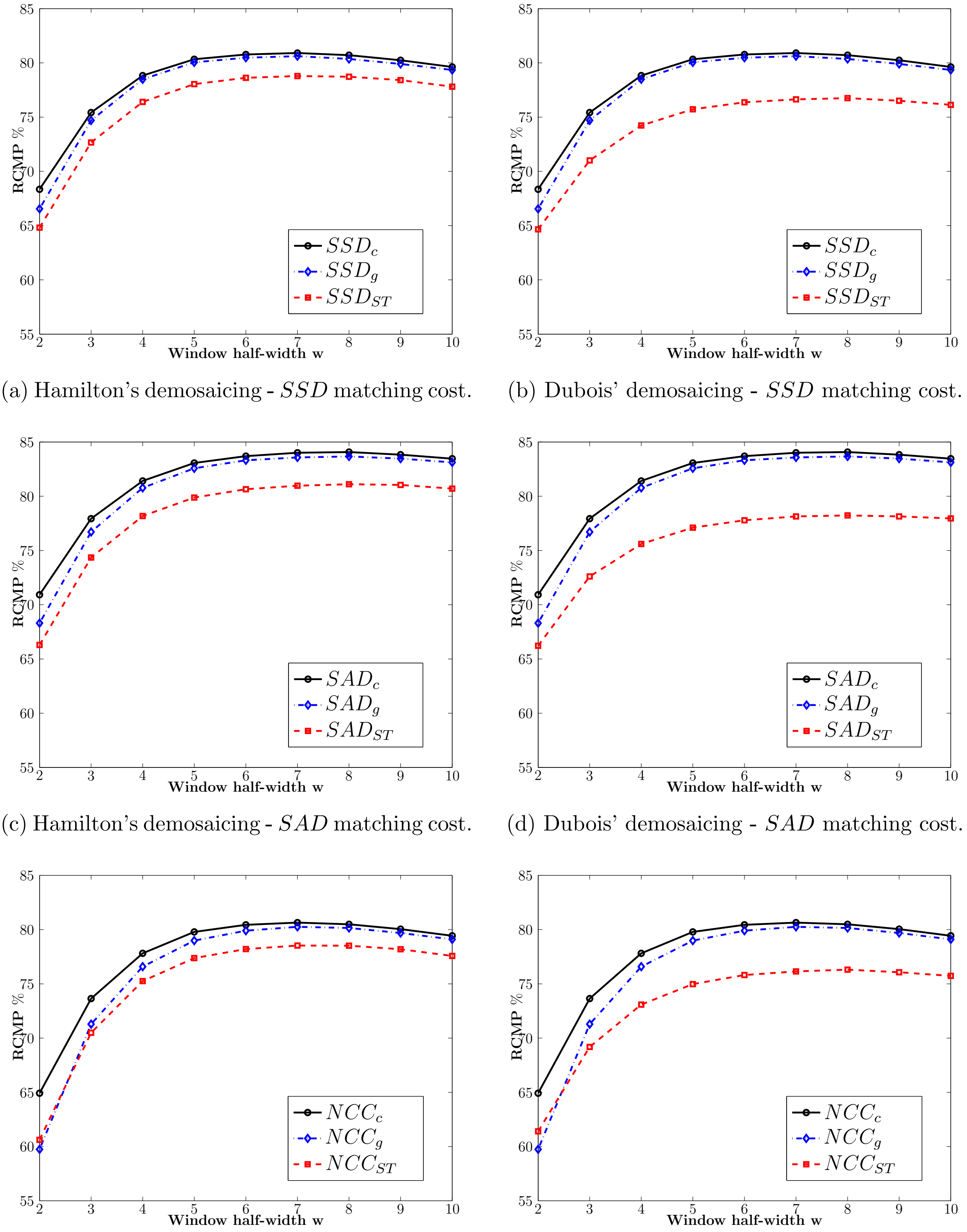
Whatever the image type and the matching cost, the matching performance increases with aggregation window half-width. Small windows do not contain enough information to allow a correct matching. At the opposite, large aggregation windows may cover image regions containing pixels with different disparities, which explains the decrease of matching performance. Here, for 'Murs' stereo image pair, the most adapted aggregation window half-width appears to range between 6 and 8 pixels.

Curves of figure 2.11 also show that, whatever the matching cost and the half-width w, the RCMP is lower for gray-level images than for full color images. The difference between RCMP ranges between 0.25% (SSD, large aggregation window) and 5.2% (NCC, small aggregation window). The increase of performance due to color is noticeably higher when the aggregation window is small. This tends to show that for small aggregation windows, color information is very useful, whereas for larger windows, image texture becomes a significant information.

Finally, the RCMP in demosaiced color images is lower that those of full color and gray-level images, except for the NCC cost when computed on small aggregation windows of gray-level images. For the Hamilton's demosaicing scheme, the difference between RCMP in demosaiced color images and full color images ranges between -4.6% (SAD, small aggregation window) and -1.8% (NCC, large aggregation window). Still for Hamilton's demosaicing scheme, the difference between RCMP in demoisaiced color images and gray-level images ranges between -2.7% (SAD, middle size aggregation window) and 0.8% (NCC, small aggregation window). This last situation is the only one for which the matching with Hamilton's demosaiced color images performs better than with gray-level images. We can see that the difference between RCMP in images demosaiced with Dubois' scheme and full color or gray-level images is even higher than for images demosaiced with Hamilton's scheme.

We can assume that the difference between the RCMP in full color and demosaiced color images is only caused by the errors introduced by the estimation of missing color components. On the other hand, the difference between the RCMP in gray-level and demosaiced color images is probably caused by the loss of texture.

These basic experimental results demonstrate, on a single image pair, that the CFA color sampling highly degrades the quality of stereo matching. This problem is not compensated by the demosaicing step, even when the most efficient demosaicing techniques are used. That leads us to propose a partial demosaicing scheme specifically designed for CFA dense stereovision.



(a) Hamilton's demosaicing - SSD matching cost.

(b) Dubois' demosaicing - SSD matching cost.

(c) Hamilton's demosaicing - SAD matching cost.

(d) Dubois' demosaicing - SAD matching cost.

(e) Hamilton's demosaicing - NCC matching cost.

(f) Dubois' demosaicing - NCC matching cost.

Figure 2.11 : Rate of correctly matched pixels (RCMP) obtained with three matching costs computed on 'Murs' stereo image pair for δ set to 0.5.

2.5 Conclusion

In this chapter, we have studied how color information can be used in stereovision. Since three-CCD color cameras which deliver full color images cannot be easily used by a stereovision setup due to their high cost, one generally use single-CCD color cameras. After having described why the demosaicing step is required to reconstruct the color images of scenes observed by single-CCD color cameras, the Bayer CFA solution has been presented. The simple bilinear interpolation has allowed us to introduce both artifact generation that demosaicing methods have to overcome, and two major rules widely used in the proposed approaches : spatial and spectral correlations. The vast majority of demosaicing methods strive to estimate the green component first, which bears the high-frequency information. The quality of this estimation strongly influences that of red and blue components.

Even if recent demosaicing methods yield very good images for human observers, none can actually recover the information that is lost during the image acquisition process. We have shown, thanks to a basic experiment, that the loss of information and/or the non-specific demosaicing step have a strong influence on the efficiency of color stereo matching techniques.

This basic experiment has been focused on the pixel matching quality obtained from the analysis of images acquired by single-CCD cameras. Two kinds of cameras are used, gray-level cameras that deliver gray-level images and color cameras which deliver color images that have been demosaiced by a scheme. Testing on ‘Murs’ stereo image pair, matching costs computed on the gray-level images outperforms matching costs computed on demosaiced color images. We can deduce that when the demosaicing scheme and /or the matching cost are not specifically designed for stereovision, then using gray-level cameras provides better results than using single-CCD color cameras.

However, this experiment shows that analyzing well represented full color information improves the performances reached by a pixel matching scheme. In the next chapter, we will present possible solutions to apply stereo matching to CFA image pairs rather than to the demosaiced color image pairs.

Chapitre 3

CFA stereovision

Résumé

Nous proposerons d'estimer la carte de disparité directement à partir des paires d'images stéréo CFA acquises par deux mono-CCD caméras couleur. Cette approche est nommée ci-après stéréovision CFA. Nous n'avons pas trouvé d'algorithme disponible dans la littérature qui estime la carte de disparité en analysant directement deux images stéréo CFA.

Le problème principal avec la stéréovision CFA est que les composants couleur disponible aux pixels homologués dans les images de gauche et de droite peuvent être différents. Par exemple, examinons la figure 3.1 qui montre un cas où le point P de la scène est projeté sur un pixel vert p_l dans l'image gauche CFA et sur un pixel rouge p_r dans l'image droite CFA. Un pixel vert (rouge, respectivement) sur l'image CFA est caractérisé par seulement le composant vert (rouge, respectivement). Ainsi, on ne peut supposer que le niveau vert du pixel de gauche est égal au niveau rouge de son pixel homologue dans l'image droite CFA.

Les fonctions de coûts de mise en correspondance qui sont calculées directement à partir des niveaux CFA ne donnant pas de résultats satisfaisants puisque l'hypothèse que les niveaux de pixels homologues sont similaires n'est pas vérifiée pour les disparités impaires. Plus précisément, la disparité entière pour chaque pixel de gauche peut être :

- Paire : Le pixel homologue de droite est caractérisé par le même composant couleur disponible dans l'image droite CFA. L'hypothèse des niveaux de similarité est vérifiée.
- Impaire : Le pixel homologue de droite est caractérisé par un autre composant

couleur disponible dans l'image droite CFA. L'hypothèse des niveaux de similarité n'est pas vérifiée.

Dans la partie suivante, nous présenterons deux solutions différentes pour calculer la carte de disparité paire à partir des images stéréo CFA. Ces deux solutions sont basées sur les fonctions des coûts de correspondance désignées pour comparer seulement les composantes couleur disponible entre les images CFA de gauche et de droite.

Comme l'estimation d'une carte de disparité paire ne peut fournir de résultats satisfaisants, nous proposerons une méthode 'coarse to fine' pour estimer les disparités dans la troisième partie. Tout d'abord, une disparité paire est 'coarsely' estimée pour chaque pixel en analysant les composants couleurs disponible des images CFA. Ensuite, une étape fine basée sur l'hypothèse de variations locales est calculée pour affiner la carte de disparité paire.

Des résultats expérimentaux avec la paire des images 'Murs' stéréo montrent les limites de l'estimation des disparités par l'analyse uniquement des niveaux CFA. Ceci nous mène à proposer une nouvelle approche basée sur une méthode de dematriçage spécialement désignée pour la mise en correspondance.

Dans la quatrième partie, nous présenterons un nouveau type d'images couleur appelé images couleur partiellement dematriçées. Ensuite, nous modifierons les coûts de correspondance standard pour qu'ils puissent être calculés avec ces images couleur partiellement dematriçées. Enfin, nous comparerons les résultats de mise en correspondance obtenus en utilisant les images couleurs dematriçées et les images couleurs partiellement dematriçées.

3.1 Introduction

We propose to estimate the disparity map directly from the pair of CFA stereo images acquired by two single-CCD color cameras. This approach is called hereafter CFA stereovision. We have found no algorithm available in the literature that estimates the disparity map by analyzing directly two CFA stereo images.

The main problem with CFA stereovision is that the color components available at homologous pixels in the left and right images may be different. For example, let us examine figure 3.1 that shows a case when a scene point P is projected onto a green pixel

p_l in the left CFA image and onto a red pixel p_r in the right CFA image. A green (red, respectively) pixel in a CFA image is characterized by only the green (red, respectively) color component. Therefore, one cannot assume that the green level of the left pixel is equal to the red level of its homologous pixel in the right CFA image.

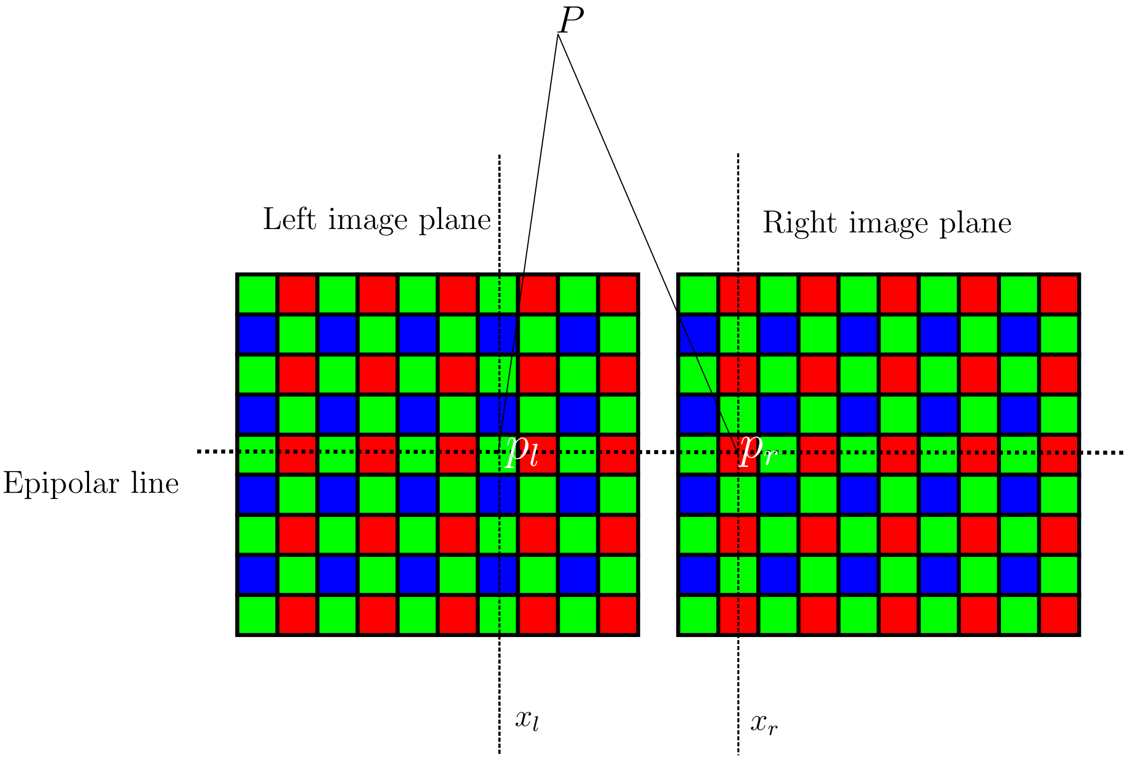


Figure 3.1 : Problem of matching CFA images. The homologous pixels are not characterized by the same color component. The left pixel is a green pixel while its homologous pixel is red.

Window-based matching costs computed directly from CFA levels do not provide satisfying results since the assumption that the CFA levels of homologous pixels are similar is not met for odd disparities. More precisely, the integer disparity for each left pixel can be :

– even : the right homologous pixel is characterized by the same available color component in the right CFA image. The assumption about similarity of levels is met ;

– odd : the right homologous pixel is characterized by another available color component in the right CFA image. The assumption about similarity of levels is not met.

In the next section, we present two different solutions to compute the even estimated disparity maps from the pair of stereo CFA images. These two solutions are based on matching costs designed for comparing only similar available color components between the left and right CFA images.

As estimating an even disparity map cannot provide satisfying results, we propose a coarse to fine disparity estimation scheme in the third section. First, an even disparity is coarsely estimated for each pixel by analyzing similar available color components of CFA images. Then, a fine step based on an assumption about the local variations of levels is computed in order to refine the even estimated disparity map.

Experimental results with 'Murs' stereo image pair show the limits of disparity estimation by analyzing only the CFA levels. This leads us to propose a new approach based on a partial demosaicing scheme specially designed for stereo matching.

In the fourth section, we present a new kind of color images called partially demosaiced color images. Then, we modify the standard matching costs so that it can be computed with these partially demosaiced color images. Finally, we compare the matching results obtained using demosaiced color images and partially demosaiced color images.

3.2 Even estimated disparity

In this section, we aim to calculate the best "even" estimated disparity at each pixel. Even if the disparity of a pixel is odd, its estimated disparity will be even. For example, if a pixel p_l in the left image has an odd disparity equal to $2k+1$ where k is an integer number, then the best even estimated disparity will be either $2k$ or $2k+2$, yielding a disparity estimation error of one pixel.

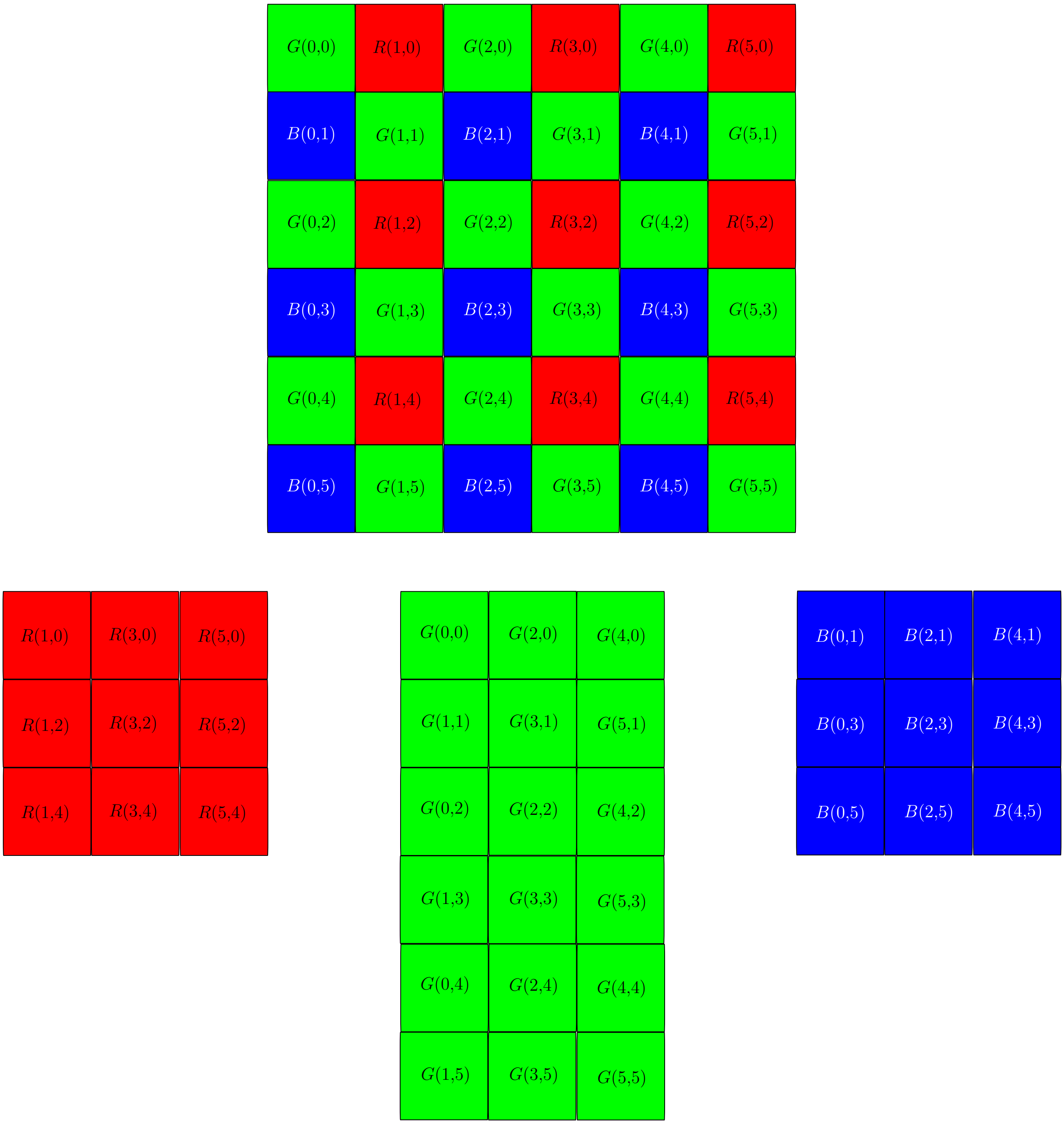
In order to provide the even estimated disparity map, we can follow two strategies.

3.2.1 Even disparity estimation method 1 (EDE1)

The first one, called hereafter even disparity estimation method 1 (EDE1), splits the left (right, respectively) CFA image whose size is $M \times N$ pixels into three images as illustrated in figure 3.2 :

- one composed of only green pixels : the left (right, respectively) green image whose size is $\frac{M}{2} \times N$;

- one composed of only red pixels : the left (right, respectively) red image whose size is $\frac{M}{2} \times \frac{N}{2}$;
- one composed of only blue pixels : the left (right, respectively) blue image whose size is $\frac{M}{2} \times \frac{N}{2}$.



$G(0,0)$	$R(1,0)$	$G(2,0)$	$R(3,0)$	$G(4,0)$	$R(5,0)$
$B(0,1)$	$G(1,1)$	$B(2,1)$	$G(3,1)$	$B(4,1)$	$G(5,1)$
$G(0,2)$	$R(1,2)$	$G(2,2)$	$R(3,2)$	$G(4,2)$	$R(5,2)$
$B(0,3)$	$G(1,3)$	$B(2,3)$	$G(3,3)$	$B(4,3)$	$G(5,3)$
$G(0,4)$	$R(1,4)$	$G(2,4)$	$R(3,4)$	$G(4,4)$	$R(5,4)$
$B(0,5)$	$G(1,5)$	$B(2,5)$	$G(3,5)$	$B(4,5)$	$G(5,5)$

$R(1,0)$	$R(3,0)$	$R(5,0)$
$R(1,2)$	$R(3,2)$	$R(5,2)$
$R(1,4)$	$R(3,4)$	$R(5,4)$

$G(0,0)$	$G(2,0)$	$G(4,0)$
$G(1,1)$	$G(3,1)$	$G(5,1)$
$G(0,2)$	$G(2,2)$	$G(4,2)$
$G(1,3)$	$G(3,3)$	$G(5,3)$
$G(0,4)$	$G(2,4)$	$G(4,4)$
$G(1,5)$	$G(3,5)$	$G(5,5)$

$B(0,1)$	$B(2,1)$	$B(4,1)$
$B(0,3)$	$B(2,3)$	$B(4,3)$
$B(0,5)$	$B(2,5)$	$B(4,5)$

Figure 3.2 : CFA image splitting into three images : red, blue and green images.

The stereo matching process is simplified by matching pixels in the left red, green and blue images with pixels in the right red, green and blue images, respectively. The flowchart of this method is illustrated in figure 3.3.

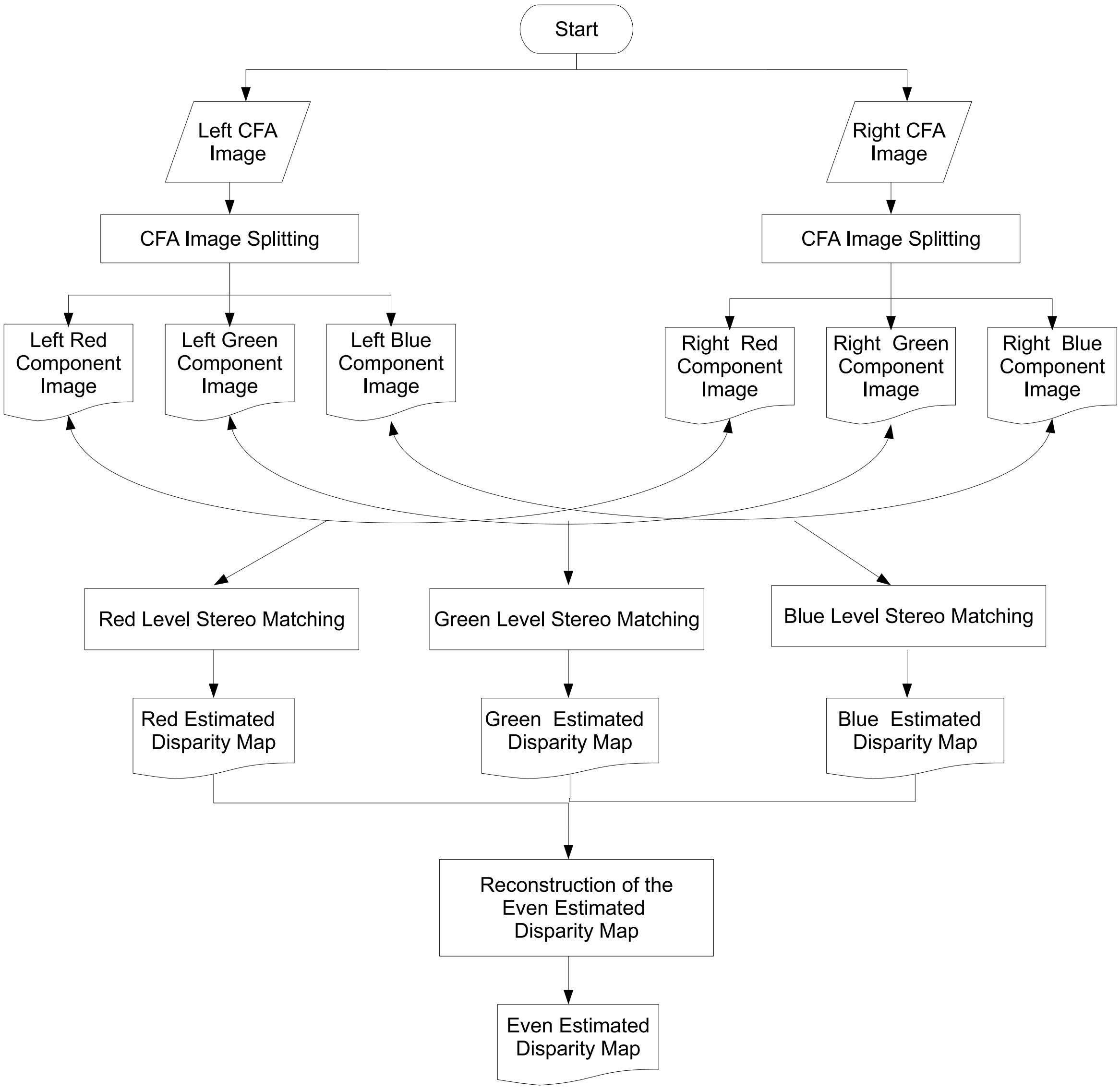

Start
Left CFA Image
Right CFA Image
CFA Image Splitting
CFA Image Splitting
Left Red Component Image
Left Green Component Image
Left Blue Component Image
Right Red Component Image
Right Green Component Image
Right Blue Component Image
Red Level Stereo Matching
Green Level Stereo Matching
Blue Level Stereo Matching
Red Estimated Disparity Map
Green Estimated Disparity Map
Blue Estimated Disparity Map
Reconstruction of the Even Estimated Disparity Map
Even Estimated Disparity Map

Figure 3.3 : Even disparity estimation method 1 (EDE1) flowchart.

Gray-level versions of SSD, SAD and NCC matching costs ($COST_g$) can be computed on the pairs of red, green and blue images. However, in order to follow the same notation as that used in chapter 2, we will replace the subscript g in the gray-level versions of these costs with the subscript R (G and B, respectively) when it is applied to red (green and blue, respectively) images. So, the generic notation of these costs will be $COST_R$, $COST_G$

and $COST_B$.

At the end of the matching process, the red, green, and blue estimated disparity maps have to be combined so that the even estimated disparity map is retrieved. In the combined even estimated disparity map, at each red (green and blue, respectively) pixel, we double its corresponding estimated disparity value calculated using the red (green and blue, respectively) images.

3.2.2 Even disparity estimation method 2 (EDE2)

The second method, called hereafter even disparity estimation 2 (EDE2), consists in analyzing the CFA images as gray-level images. Then, we apply gray-level stereo matching methods by modifying the matching costs to take into consideration only pixels at even shifts.

Thus, SSD_g^w (see equation (1.14)), SAD_g^w (see equation (1.16)) and NCC_g^w (see equation (1.17)) matching costs are modified in order to take only even shifts.

For example, the SSD_g^w cost defined for gray-level images in equation (1.14) is modified to SSD_{EDE2}^w and expressed as :

$$\begin{aligned} SSD_{EDE2}^w(x_l,y,2k) = & \\ & \sum_{i=-w}^{w}\sum_{j=-w}^{w} (CFA_l(x_l+i,y+j) - CFA_r(x_l+i-2k,y+j))^2, \end{aligned} \tag{3.1}$$

where k is a positive integer, and $CFA_l(x_l,y)$ is the available color component of pixel with coordinates (x_l,y) in the left CFA image. Similarly, we can define SAD_{EDE2}^w and NCC_{EDE2}^w. The generic notation of these costs is $COST_{EDE2}$ and the flowchart of this method is illustrated in figure 3.4.

The difference between the two even disparity estimation methods is that the aggregation window used by the second method contains all kinds of pixels (red, blue, green) whereas the aggregation window used by the first method contains only pixels with the same available color component as the central pixel (specifically red, green or blue pixels).

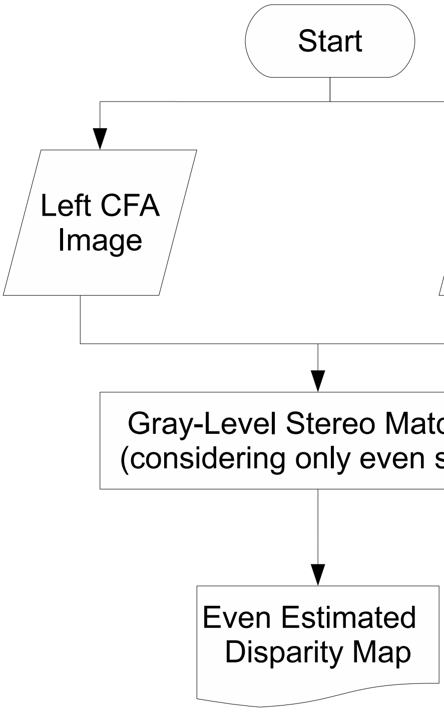

Start
Left CFA Image
Right CFA Image
Gray-Level Stereo Matching (considering only even shifts)
Even Estimated Disparity Map

Figure 3.4 : Even disparity estimation method 2 (EDE2) flowchart.

3.2.3 Experimental results with even disparity estimation methods

To assess the efficiency of even disparity estimation methods, we compare RCMP reached by using the standard method (see flowchart in figure 2.9) and both even methods (see flowcharts in figures 3.3 and 3.4) applied to 'Murs' stereo image pair.

Our experimental procedure is decomposed into successive steps. First, in order to produce left and right CFA images from full color images, we simulate the CFA sampling process by keeping only one out of the three color components at every pixel. This is performed according to the spatial arrangement of the Bayer CFA (see figure 2.4). Then, the two missing color components at each pixel of the so-formed CFA images are estimated by either Hamilton's or Dubois' method.

At pixels where the disparity is odd, the even estimated disparity error is at least one pixel. So, when the disparity error tolerance δ is set to 0.5, all pixels with odd disparities will not be correctly matched. Moreover, when the disparity map is available with subpixel precision, all pixels whose disparity values do not range between $2k-0.5$ and $2k+0.5$ cannot be correctly matched by even disparity estimation methods.

In 'Murs' stereo image pair, 51.9% of the left pixels cannot be correctly matched by even disparity estimation methods when δ is set to 0.5. However, when δ is set to 1, all pixels can be correctly matched. Hence, the RCMP will be calculated when δ is set to 1.

Figure 3.5 displays the RCMP obtained using three matching costs. In each of the of the six subfigures, the three curves represent the RCMP with respect to different aggregation window half-widths w.

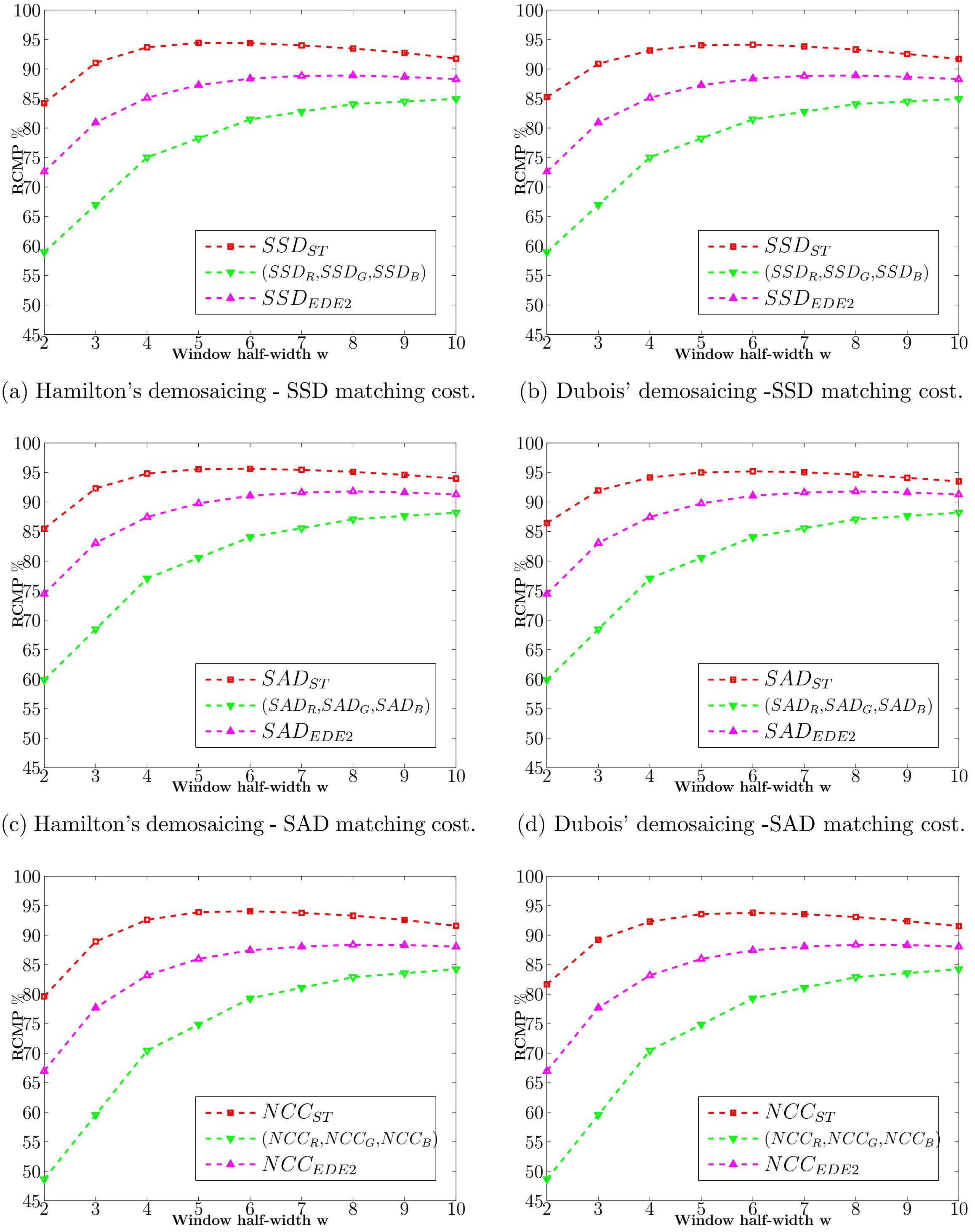
The red dotted curve corresponds to the RCMP in the pair of demosaiced color images, using $COST_{ST}$ for cost computation. The green dotted curve corresponds to the RCMP in CFA image pair, using $COST_R$, $COST_G$ and $COST_B$ for cost computation. Finally, the magenta dotted curve corresponds to the RCMP in CFA image pair, using $COST_{EDE2}$ for cost computation. The left column shows results for Hamilton's demosaicing scheme and the right column for Dubois' scheme. The three lines correspond respectively to the SAD, SSD and NCC matching costs.

Curves of figure 3.5 show that, whatever the matching cost and the half-width w, the RCMP is higher for EDE2 method than for EDE1 method. The difference between RCMP ranges between 3% (SSD, large aggregation window) and 17% (NCC, small aggregation window). This proves that taking into account all the pixels in the aggregation window provides better results than taking into account only pixels for which the same color components available in the aggregation window.

We notice also that the RCMP reached by using the standard method is higher than that of even disparity estimation methods. For the Hamilton's demosaicing scheme, the difference between RCMP using the standard method and EDE2 (which is better than EDE1 method) ranges between 3% and 12%. For Dubois' demosaicing scheme, the difference ranges between 2% and 15%.

As a conclusion, even if the acceptable disparity error tolerance is set to one, the standard method outperforms the even disparity estimation ones. However, a more accurate disparity map is needed in most cases. So, the RCMP is calculated for δ set to 0.5. In this case, the RCMP reached by using an even disparity estimation method on 'Murs' CFA image pair cannot be higher than 48.06% since 51.94% of left pixels cannot be correctly matched.

For this reason, we propose another method which refines a posteriori the even estimated disparity map calculated by EDE2 method.



(a) Hamilton's demosaicing - SSD matching cost.

(b) Dubois' demosaicing -SSD matching cost.

(c) Hamilton's demosaicing - SAD matching cost.

(d) Dubois' demosaicing -SAD matching cost.

(e) Hamilton's demosaicing - NCC matching cost.

(f) Dubois' demosaicing -NCC matching cost.

Figure 3.5 : Rate of correctly matched pixels (RCMP) obtained with three matching costs computed on 'Murs' stereo image pair for δ set to 1 using standard, EDE1 and EDE2 methods.

3.3 Coarse to fine disparity estimation

3.3.1 RGGBD cost

Let us examine the case of odd disparities, when the available color components of homologous pixels are different. This occurs, for example, when a red pixel p_l with coordinates (x_l,y) in the left CFA image is homologous to a green pixel p_r in the right CFA image. The homologous pixel p_r is at a horizontal shift from p_l in the right image equal to $d(x_l,y)$, so its coordinates are $(x_l - d(x_l,y),y)$. We assume that when p_l is homologous to p_r, the variation between the red level of p_l and the green level of its right neighbor is close to the variation between the green level of p_r and the red level of its right neighbor. In this case, the following approximation can be accepted as illustrated in figure 3.6 :

$$|R_l(x_l,y) - G_l(x_l+1,y)| \approx |G_r(x_l - d(x_l,y),y) - R_r(x_l - d(x_l,y)+1,y)|\,. \tag{3.2}$$

Using this approximation, a new difference cost is used by the matching process. The new cost, denoted as $RGGBD^w$ (which stands for R-G G-B difference), is defined as :

$$\begin{aligned} RGGBD^w(x_l,y,s) &= \textstyle\sum_{j=-w}^{w}\sum_{i=-w}^{w}\left(diff_l(x_l+i,y+j)\right.\\ &- \left.diff_r(x_l-w+s+i,y+j)\right)^2, \end{aligned} \tag{3.3}$$

where $diff_l(x_l,y)$ is the absolute difference between the available color component levels of two consecutive pixels in the left image defined as :

$$diff_l(x_l,y) = |CFA_l(x_l,y) - CFA_l(x_l+1,y)|. \tag{3.4}$$

The left absolute difference between two consecutive pixels in the aggregation window can have several forms depending on the position of (x_l,y) as summarized in the following :

$$diff_l(x_l,y) = \begin{cases} |G_l(x_l,y) - R_l(x_l+1,y)| & \text{if } x_l \text{ is even and } y \text{ is even,}\\ |R_l(x_l,y) - G_l(x_l+1,y)| & \text{if } x_l \text{ is odd and } y \text{ is even,}\\ |B_l(x_l,y) - G_l(x_l+1,y)| & \text{if } x_l \text{ is even and } y \text{ is odd,}\\ |G_l(x_l,y) - B_l(x_l+1,y)| & \text{if } x_l \text{ is odd and } y \text{ is odd.} \end{cases} \tag{3.5}$$

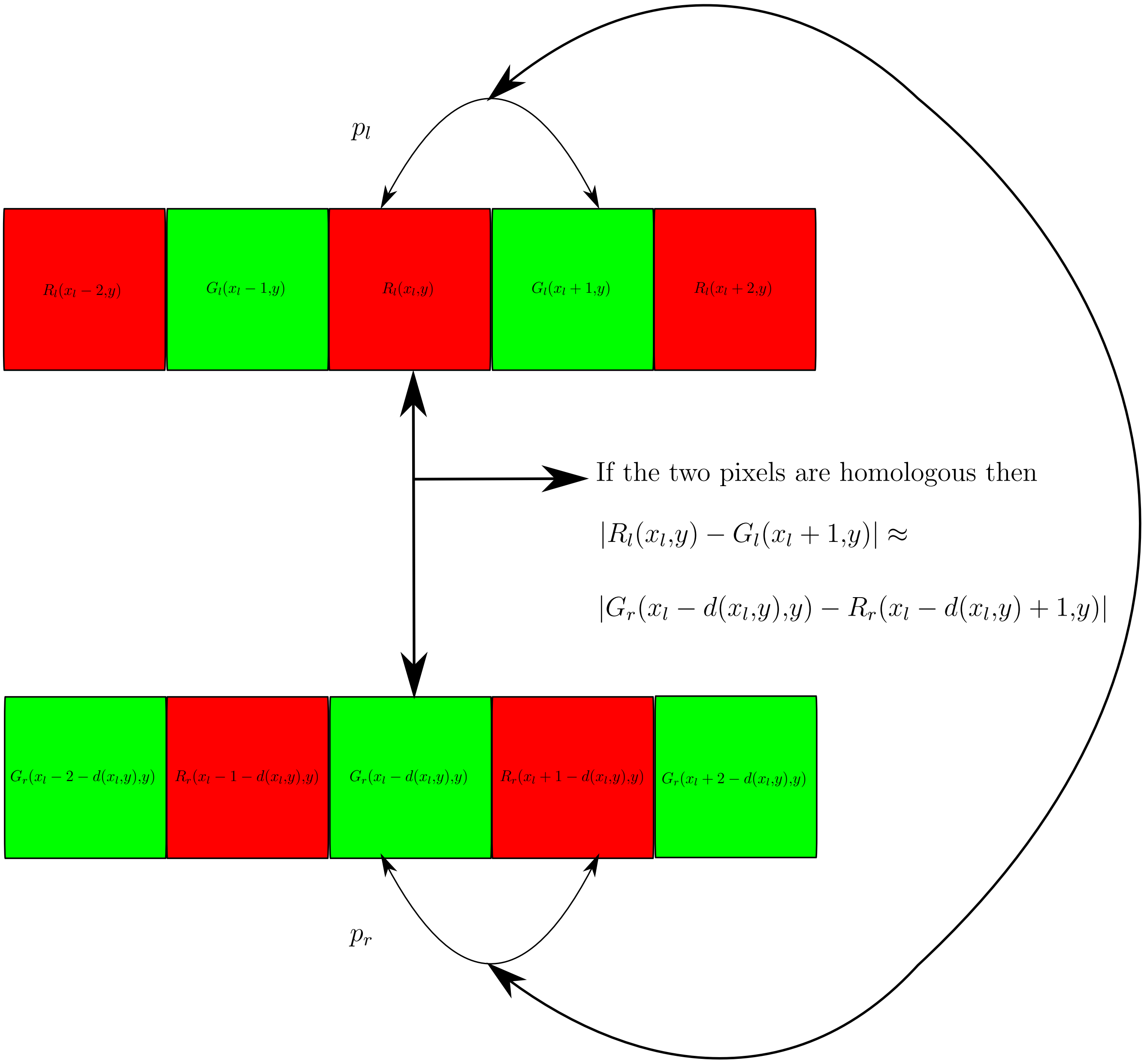


Figure 3.6 : RG-GB difference approximation.

3.3.2 Coarse to fine disparity estimation method (CTFDE)

We apply it as a refinement step after having computed the even estimated disparity map using EDE2 method explained in section 3.2.

The idea is to use the even estimated disparity map as a coarse guess of the estimated disparity map. For each pixel p_l with coordinates (x_l,y) in the left CFA image, we will use the even estimated disparity value to identify the possible candidate pixels in the right image. Let us denote the even estimated disparity of p_l by $\hat{d}^{w}_{EDE2}(x_l,y)$ where we added the subscript $EDE2$ to the symbol of estimated disparity. So, the fine estimated disparity value $\hat{d}^w(x_l,y)$ is defined as :

$$\hat{d}_l^w(x_l,y) = \arg\min_s(RGGBD_g^w(x_l,y,s)), \tag{3.6}$$

where $s \in [\hat{d}^w_{EDE2}(x_l,y) - 1, \hat{d}^w_{EDE2}(x_l,y) + 1]$. In other words, the fine homologous pixel will be the homologous pixel found by EDE2 or its left neighbor or its right one. The flowchart of this method is shown in figure 3.7.

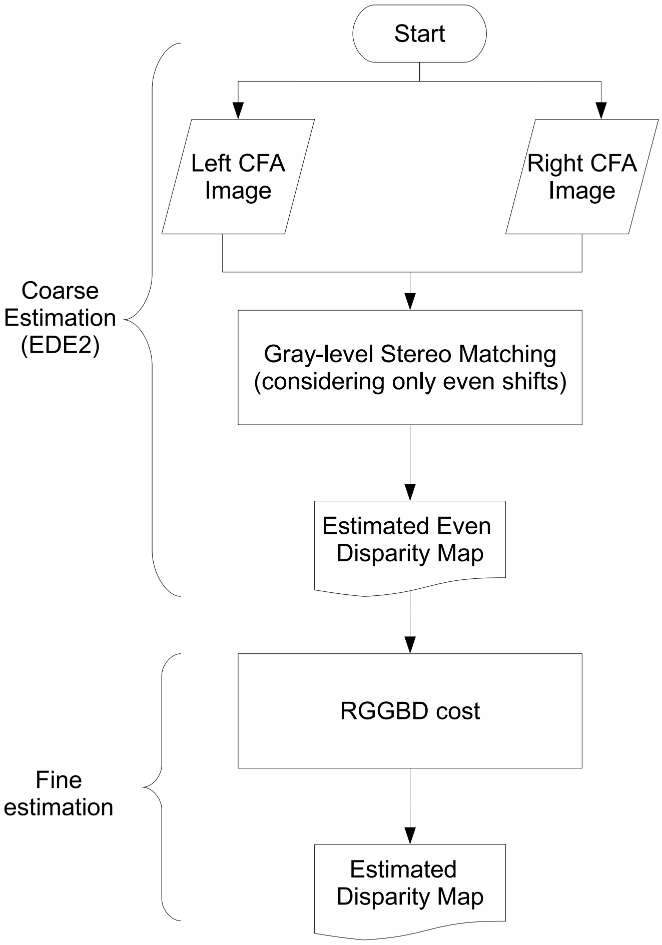


Figure 3.7 : Coarse to fine disparity estimation method flowchart.

3.3.3 Experimental results

In this section, we compare the matching results reached by standard method (see flowchart in figure 2.9) and CTFDE method (see flowchart in figure 3.6) on 'Murs' stereo image pair. We also include the results of the EDE2 method since it is the coarse estimation of the CTFDE method.

Our experimental procedure is decomposed into successive steps. First, in order to produce left and right CFA images from full color images, we simulate the CFA sampling process by keeping only one out of the three color components at every pixel. This is performed according to the spatial arrangement of the Bayer CFA (see figure 2.4). Then, the two missing color components at each pixel of the so-formed CFA images are estimated by either Hamilton's method or Dubois' method.

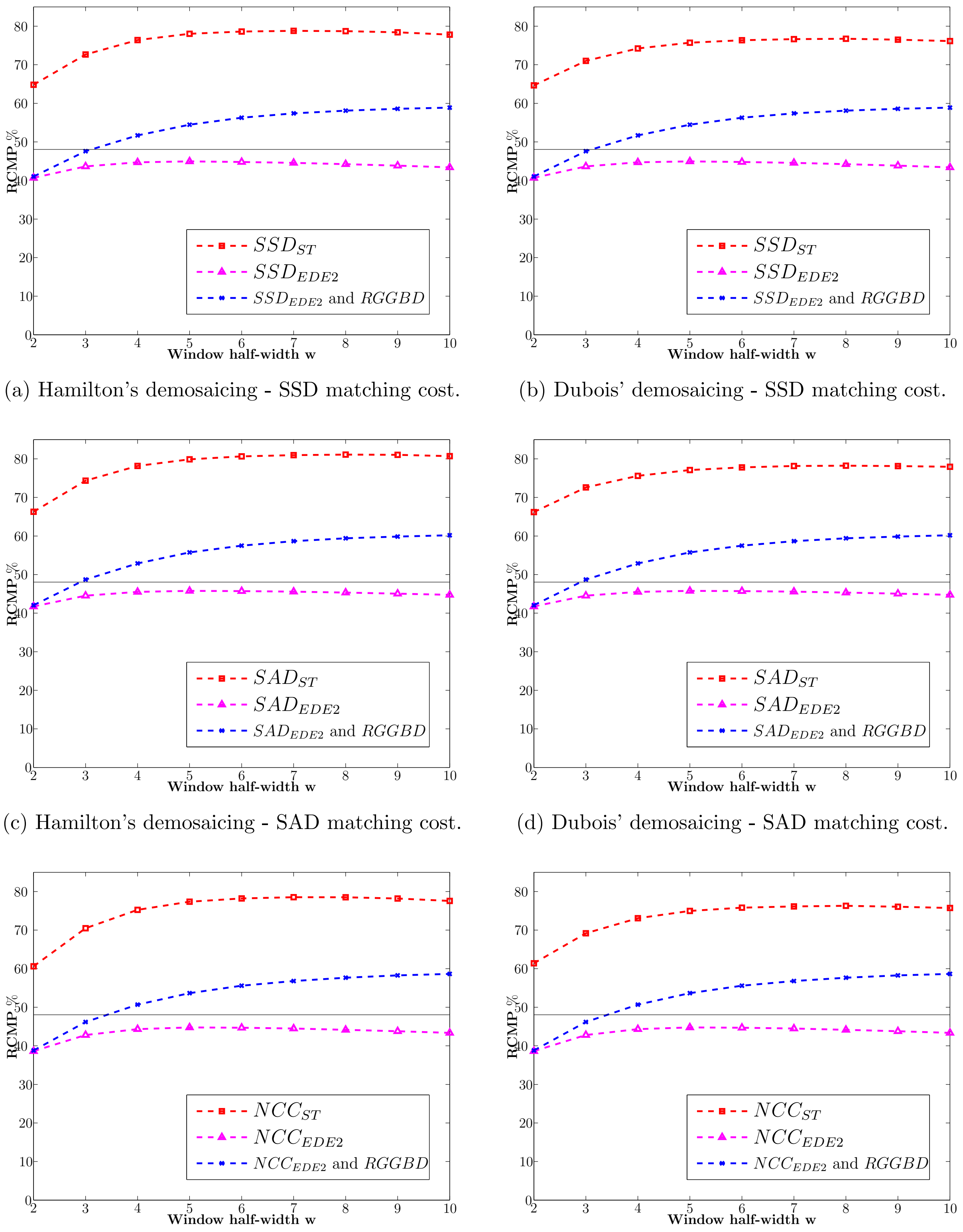
Figure 3.8 displays the RCMP obtained using three matching costs. In each of the of the six subfigures, the three curves represent the RCMP with respect to different aggregation window half-widths w.

The red dotted curve corresponds to the RCMP in the pair of demosaiced color images using the standard method. The magenta dotted curve corresponds to the RCMP in the pair of CFA images using EDE2 method. Finally, the blue dotted curve corresponds to the RCMP in the pair of CFA images, using the new CTFDE method. The left column shows results for Hamilton's demosaicing scheme and the right column for Dubois' scheme. The three lines correspond respectively to the SAD, SSD and NCC matching costs.

The disparity estimation tolerance δ is set to 0.5 in this experiment. In this case, as explained before, 51.94% of the left pixels cannot be correctly matched using $EDE2$ method. For this reason we add to each graph a horizontal black line at 48.06% to show the maximum limit of RCMP using EDE2 method.

Curves of figure 3.8 show that, whatever the matching cost and the half-width w, the RCMP is lower for EDE2 method than for CTFDE method. The difference between RCMP ranges between 0% (small aggregation window) and 15% (large aggregation window). We also notice that RCMP of CTFDE is higher than the 48.06% when w is larger than 3. This proves that coarse to fine method improves the even disparity estimation.

However, we still notice that the RCMP reached by using the standard method is higher than that of CTFDE method. That leads us to propose a new approach which consists in partial demosaicing of CFA images.



(a) Hamilton's demosaicing - SSD matching cost.

(b) Dubois' demosaicing - SSD matching cost.

(c) Hamilton's demosaicing - SAD matching cost.

(d) Dubois' demosaicing - SAD matching cost.

(e) Hamilton's demosaicing - NCC matching cost.

(f) Dubois' demosaicing - NCC matching cost.

Figure 3.8 : Rate of correctly matched pixels (RCMP) obtained with three matching costs computed on 'Murs' stereo image pair for δ set to 0.5 using standard, EDE2 and CTFDE methods.

3.4 Partial demosaicing for CFA stereovision

3.4.1 Second and third color components

We assume that matching errors generated by the different matching costs applied to demosaiced color images are mainly caused by estimation errors on the two missing color components at each pixel.

The Bayer pattern which is the most widely used, is designed so that the pixels in one line of the CFA image are characterized by one among two possible color components. Figure 2.4 shows that pixels of even (odd, respectively) lines are characterized by red or green levels (blue or green levels, respectively).

Using a demosaicing scheme, the missing components for each pixel can be estimated. The estimated components are denoted second color component (SCC) and third color component (TCC). SCC is the estimated component that is available at the same line where TCC is the one that is not available. This means that SCC is the green color component for all red and blue pixels. However, for green pixels the SCC is the red color component for even lines and the blue one for odd lines. This is summarized by the following :

$$SCC(x,y) = \begin{cases} \hat{G}(x,y) & \text{for red and blue pixels,} \quad (3.7a) \\ \hat{R}(x,y) & \text{for green pixels in even lines,} \quad (3.7b) \\ \hat{B}(x,y) & \text{for green pixels in odd lines.} \quad (3.7c) \end{cases}$$

$$TCC(x,y) = \begin{cases} \hat{B}(x,y) & \text{in even lines,} \quad (3.8a) \\ \hat{R}(x,y) & \text{in odd lines.} \quad (3.8b) \end{cases}$$

To evaluate the estimated color image quality in comparison with the original full color image, an objective criterion is used, namely the peak signal-to-noise ratio ($PSNR$) derived from the mean square error (MSE) between the two images. On the red plane for instance, these quantities are defined as :

$$MSE^R = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \Big(R(x,y) - \hat{R}(x,y) \Big)^2, \tag{3.9}$$

$$PSNR^R = 10 \cdot \log_{10} \left(\frac{255^2}{MSE^R} \right). \tag{3.10}$$

Considering the left demosaiced color image of the 'Murs' stereo image pair, we calculate the $PSNR$ for SCC, TCC, red, green and blue color components when using either Hamilton's method or Dubois' method. The results are grouped in tables 3.1 and 3.2.

Table 3.1 : PSNR for SCC, TCC, red, green and blue color components for 'Murs' stereo image pair where missing components are estimated using Hamilton's method.

Images	$PSNR^R$	$PSNR^G$	$PSNR^B$	$PSNR^{SCC}$	$PSNR^{TCC}$
Left image	28.596	29.348	28.441	31.523	26.758
right image	28.563	29.356	28.456	31.528	26.745

Table 3.2 : PSNR for SCC, TCC, red, green and blue color components for 'Murs' stereo image pair where missing components are estimated using Dubois' method.

Images	$PSNR^R$	$PSNR^G$	$PSNR^B$	$PSNR^{SCC}$	$PSNR^{TCC}$
Left image	31.487	33.866	31.262	33.946	29.768
right image	31.382	33.811	31.227	33.889	29.694

We found that for both demosaicing methods, $PSNR^{SCC}$ is higher than $PSNR^{TCC}$. We can deduce that the estimation quality of the third color component is lower than that of the second color component. For this reason, we propose to reduce the matching error by estimating only SCC.

3.4.2 Partial demosaicing then matching

A partial demosaicing scheme characterizes each pixel of an even (odd, respectively) line only by its red (blue, respectively) and green levels in the so-called *partially* demosaiced color image (as illustrated in figure 3.9).

Each pixel with coordinates (x,y) in the partially demosaiced color image is characterized by a two-dimensional partial color point denoted $\hat{\mathbf{I}}_{PA}(x,y)$. This partial color point is composed of the available color component and the estimated second color component and is defined as :

$$\hat{\mathbf{I}}_{PA}(x,y) = \begin{cases} (R(x,y),\hat{G}(x,y))^T & \text{if } x \text{ is odd and } y \text{ is even,} & (3.11a) \\ (\hat{R}(x,y), G(x,y))^T & \text{if } x \text{ is even and } y \text{ is even,} & (3.11b) \\ (\hat{G}(x,y), B(x,y))^T & \text{if } x \text{ is even and } y \text{ is odd,} & (3.11c) \\ (G(x,y),\hat{B}(x,y))^T & \text{if } x \text{ is odd and } y \text{ is odd,} & (3.11d) \end{cases}$$

where the *SCC* color component is estimated by Hamilton's or Dubois' approach.

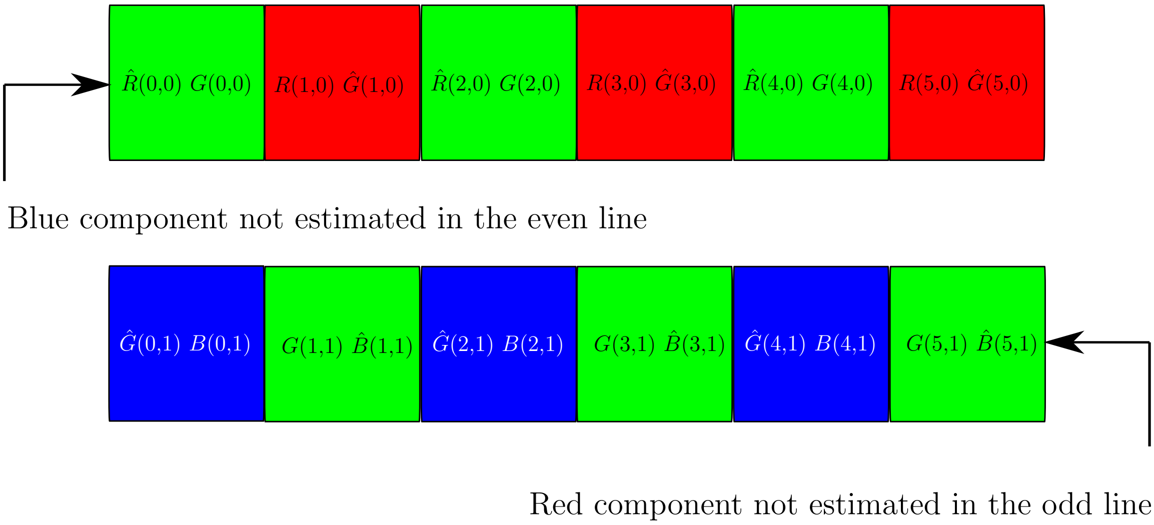


Figure 3.9 : Partially demosaiced color image.

Once the pair of partially demosaiced color images has been reconstructed, a matching algorithm can be applied. The flowchart of this method called partial method is shown in figure 3.10.

Here, we use a local stereo matching algorithm based on matching costs and WTA method. Thus, SSD_c^w (see equation (2.1)), SAD_c^w (see equation (2.2)) and NCC_c^w (see equation (2.3)) matching costs are modified in order to take into account partial color points rather than color points.

The generalization to partially demosaiced color images of the sum of squared differences cost computed between the left pixel with coordinates (x_l,y) and a candidate pixel in the right partially demosaiced color image, with s-shifted coordinates $(x_l - s,y)$, is expressed as :

$$SSD^w_{PA}(x_l,y,s) \;=\; \sum_{i=-w}^{w}\sum_{j=-w}^{w}\left\|\hat{\mathbf{I}}_{lPA}(x_l+i,y+j)-\hat{\mathbf{I}}_{rPA}(x_l+i-s,y+j)\right\|^2 ,\tag{3.12}$$

where $\|\cdot\|$ is the euclidean norm. In equation (3.12), $\hat{\mathbf{I}}_{lPA}$ and $\hat{\mathbf{I}}_{rPA}$ are the partial color points associated respectively with the left and right pixels, s is the spatial shift along the horizontal epipolar line, and w the half-width of a $(2w+1)\times(2w+1)$ aggregation window.

Similarly, SAD^w_c can be generalized to deal with partially demosaiced color images as :

$$SAD^w_{PA}(x_l,y,s) \;=\; \sum_{i=-w}^{w}\sum_{j=-w}^{w} AD(x_l+i,y+j,s),\tag{3.13}$$

where $AD(x_l,y,s)$ is the sum of the absolute difference between the corresponding components of the partial color points $\hat{\mathbf{I}}_{lPA}(x_l,y)$ and $\hat{\mathbf{I}}_{rPA}(x_l-s,y)$ defined as :

$$= \begin{cases} |CFA_l(x_l,y)-CFA_r(x_l-s,y)|+|SCC_l(x_l,y)-SCC_r(x_l-s,y)| & \text{if } s \text{ is even,}\\ |CFA_l(x_l,y)-SCC_r(x_l-s,y)|+|SCC_l(x_l,y)-CFA_r(x_l-s,y)| & \text{if } s \text{ is odd.}\end{cases}\tag{3.14}$$

The normalized cross-correlation cost (NCC^w_c, see equation (2.3)) can be generalized to partially demosaiced color images using the expression :

$$NCC^w_{PA}(x_l,y,s) \;=\; \frac{\sum_{i=-w}^{w}\sum_{j=-w}^{w}|\hat{\mathbf{I}}_{lPA}(x_l+i,y+j)\cdot\hat{\mathbf{I}}_{rPA}(x_l+i-s,y+j)|}{NormC^w_{lPA}(x_l,y)\times NormC^w_{rPA}(x_l-s,y)},\tag{3.15}$$

where $NormC^w_{lPA}(x_l,y)$ and $NormC^w_{rPA}(x_l-s,y)$ are normalization coefficients of partial color points corresponding to pixels of a $(2w+1)\times(2w+1)$ window, centered respectively on (x_l,y) in the left image and on (x_l-s,y) in the right image :

$$NormC^w_{lPA}(x_l,y)=\sqrt{\sum_{i=-w}^{w}\sum_{j=-w}^{w}\left\|\hat{\mathbf{I}}_{lPA}(x_l+i,y+j)\right\|^2},\tag{3.16}$$

and

$$NormC^w_{rPA}(x_l-s,y)=\sqrt{\sum_{i=-w}^{w}\sum_{j=-w}^{w}\left\|\hat{\mathbf{I}}_{rPA}(x_l+i-s,y+j)\right\|^2}.\tag{3.17}$$

The generic notation of SSD, SAD and NCC matching costs when computed on partially demosaiced color images will be denoted as $COST_{PA}$.

Since pixels of horizontal lines with the same parity in the left and right partially demosaiced color images are characterized by the same two color components, we can reasonably assume that the partial color points of two homologous pixels are similar. Since our partial costs compare the partial color points of left and right pixels located on the same horizontal lines, we assume that they reach an extremum when the shift s is equal to the disparity.

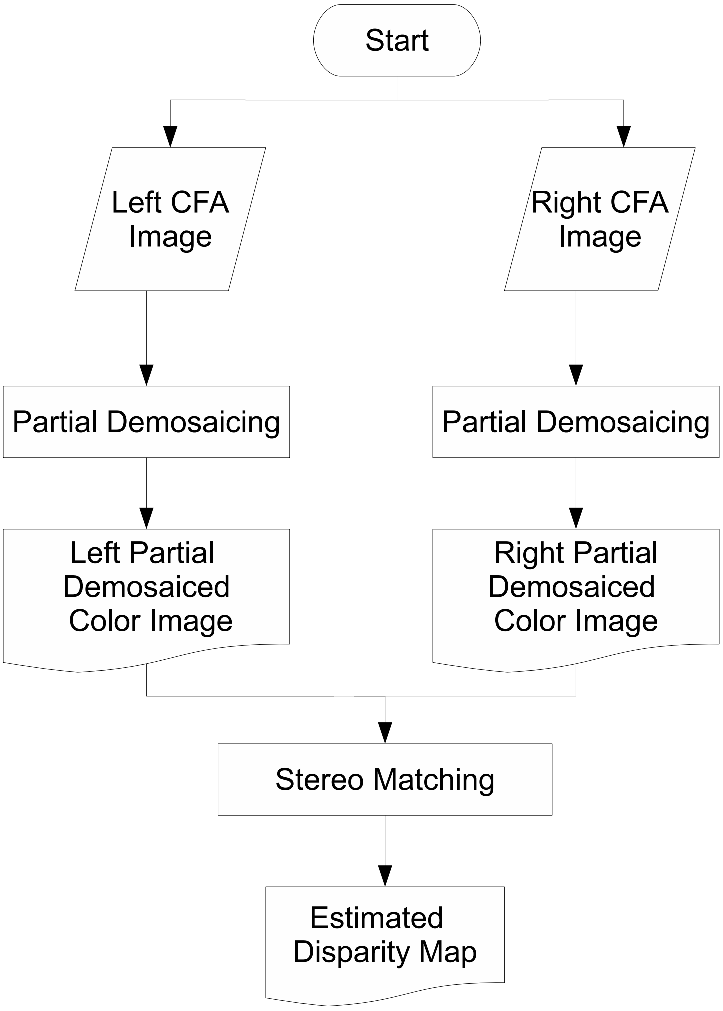


Figure 3.10 : Partial method flowchart.

3.4.3 Experiments on 'Murs' stereo image pair

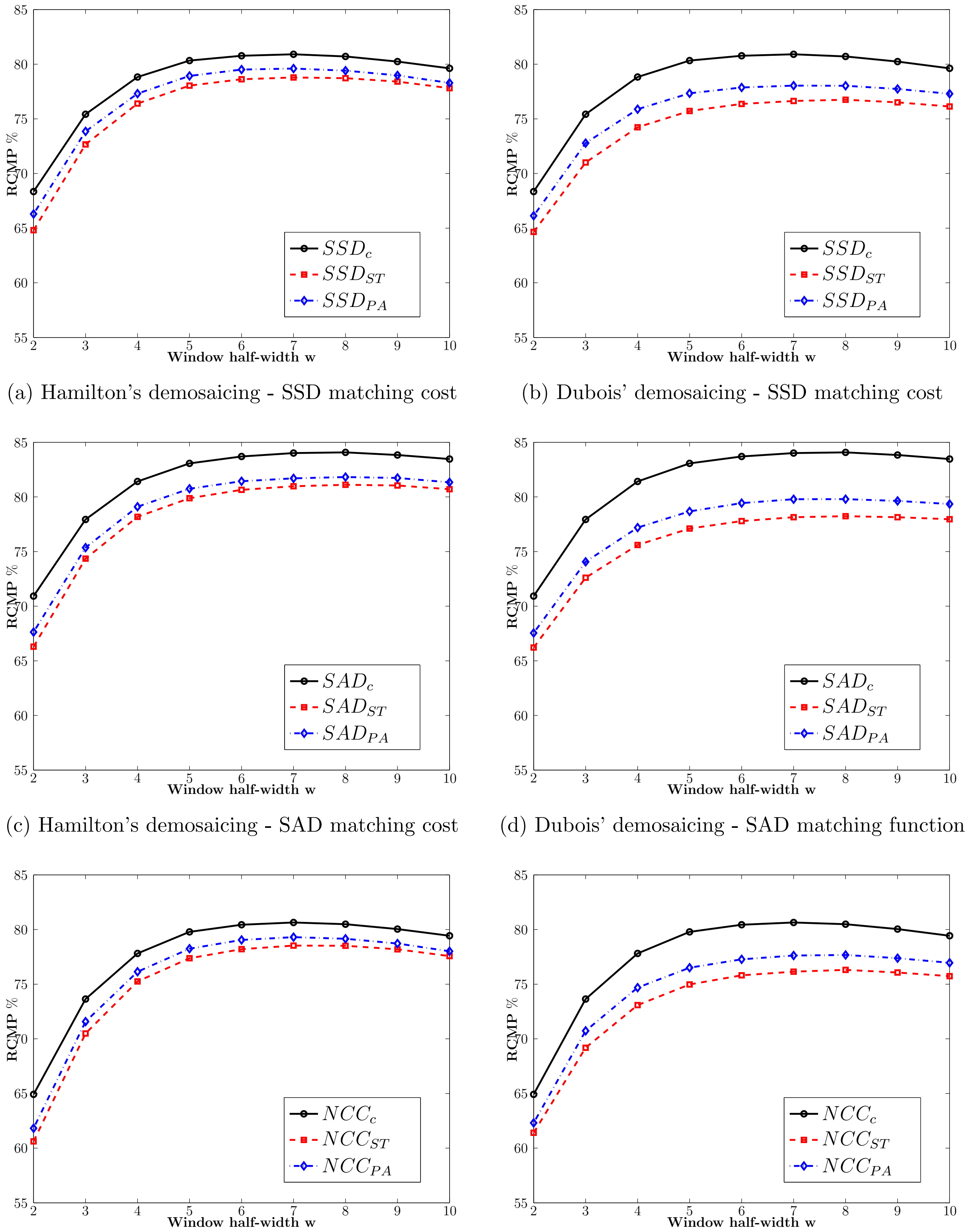
First of all, we simulate the CFA color sampling and then calculate the demosaiced and partially demosaiced color images using both Hamilton's and Dubois' demosaicing methods. After that, we apply a local stereo matching method to full, demosaiced and partially demosaiced color images using SSD, SAD and NCC matching costs.

Figure 3.11 displays the RCMP obtained using these three matching costs. The disparity error tolerance δ is set to 0.5. In each of the six subfigures, the three curves represent RCMP with respect to different aggregation window half-widths w. The black curve corresponds to the RCMP in the original full color image pair, using $COST_c$ for cost computation. The blue dotted curve corresponds to the RCMP in the partially demosaiced color image pair, using $COST_{PA}$ for cost computation. Finally, the red dotted curve corresponds to the RCMP in the pair of demosaiced color images, using $COST_{ST}$ for cost computation. The left column shows results for Hamilton's demosaicing scheme and the right column for Dubois' scheme. The three lines correspond respectively to the SAD, SSD and NCC matching costs.

Curves of figure 3.11 show that, whatever the matching cost and the half-width w, the RCMP is higher for partially demosaiced color images than for demosaiced color images. The difference between RCMP ranges between 1% and 1.5% when using Hamilton's demosaicing scheme. For Dubois' demosaicing scheme, this difference ranges between 1% and 2%.

Obviously, the matching quality reached on partially demosaiced color images is globally lower than that obtained on full color images, even if the difference between these two rates decreases when the size of the aggregation window increases.

This basic experimental results demonstrate, on a single stereo image pair, that the partial demosaicing scheme is more adapted to stereo matching than the classical full demosaicing scheme. Furthermore, the processing time needed for partial demosaicing and for computing the partial matching costs is lower than the time required by full demosaicing and standard matching costs computation. Partial demosaicing estimates one color component for each pixel while the full demosaicing estimates two color components for each pixel. Moreover, the partial matching costs use only two color components for each pixel while the standard matching costs use three ones. Finally, the number of operations needed to compute the partial matching costs is lower than that needed to compute the standard matching costs.



(a) Hamilton's demosaicing - SSD matching cost

(b) Dubois' demosaicing - SSD matching cost

(c) Hamilton's demosaicing - SAD matching cost

(d) Dubois' demosaicing - SAD matching function

(e) Hamilton's demosaicing - NCC matching cost

(f) Dubois' demosaicing - NCC matching cost

Figure 3.11 : Rate of correctly matched pixels (RCMP) obtained with three matching costs computed on 'Murs' stereo image pair for δ set to 0.5.

3.5 Conclusion

In this chapter, we have introduced the problem of CFA stereovision which consists in pixel matching by taking into account CFA levels. Different solutions have been proposed. First, we have proposed a coarse to fine solution that consists in calculating an even estimated disparity at each pixel and then in refining the estimation.

A simple experiment with one pair of stereo images has shown that matching results using the standard method based on the analysis of the fully demosaiced color images are better than using the coarse to fine one.

Furthermore, we propose to divide the two color components that are determined by the demosaicing scheme into second and third color components. By measuring the fidelity between the demosaiced color images and the original full color images, we found that the third color component estimation fidelity is worse than that of the second color component.

For this reason, we propose to reconstruct partially demosaiced color images from the CFA images to be used by the matching process. Pixels in the partially demosaiced color images are characterized by both the available color component and the second color component. Then, we adapted color matching costs to partially demosaiced color images.

We have shown, thanks to a basic example, that our partial matching costs computed on the partially demosaiced color images can enhance the matching results obtained by the standard matching costs computed on demosaiced color images. In the next chapter, we will present more experiments on benchmark images.

Chapitre 4

Experimental evaluations and discussion

Résumé

Dans le chapitre précédent, nous avons présenté une méthode partielle pour la mise en correspondance des paires d'images stéréo CFA. Pour évaluer l'efficacité de notre méthode, nous comparerons ses performances à celles qui ont été obtenues par la méthode standard expliquée dans la partie 2.4. Pour ce faire, nous utiliserons la base de données bien connue Middlebury (http://vision.middlebury.edu/stereo/). Chaque ensemble de données de cette base de données est constituée d'une paire d'images stéréo et de la carte de disparité correspondante.

Parmi les 24 ensembles de données de cette base de données, nous en avons sélectionné quatre ('Cones', 'Poster', 'Tsukuba' et 'Venus') pour l'étude expérimentale de ce chapitre. Nous utiliserons également deux ensembles de données stéréo conçus par Bocquillon, intitulé 'Journaux' et "Plante" et disponible sur ~http://www.irit.fr/Benoit.Bocquillon/MYCVR/download.php. Ces images stéréo résultent de la simulation d'acquisition des images par les caméras couleur à trois capteurs.

Grâce à la carte de disparité et compte tenu d'une tolérance sur l'erreur de disparité, nous pouvons déterminer les pixels qui sont correctement appariés par les méthodes partielles et standard appliquées à un ensemble de pair des images stéréo. Toutefois, une comparaison des performances basée sur le taux global de pixels qui sont correctement

appariés par les méthodes partielle et standard n'est pas suffisante. En effet, cela ne signifie pas que tous les pixels correctement appariés par la première méthode, sont également bien appariés par la seconde. Donc, il est intéressant d'examiner plus précisément les positions spatiales des pixels qui sont correctement appariés par les deux méthodes.

Pour cela, nous diviserons les pixels en différents sous-ensembles en fonction des résultats de mise en correspondance obtenus par les méthodes partielle et standard. Pour comparer les performances obtenues par les deux méthodes, on compare la taille de ces sous-ensembles. Cette procédure expérimentale est détaillée dans la deuxième partie de ce chapitre.

Dans la troisième partie de ce chapitre, nous comparerons les taux de pixels qui sont correctement appariés en suivant cette procédure expérimentale. Les expériences sont réalisées avec les six ensembles de données choisis.

Dans la quatrième partie, nous essayerons d'expliquer les raisons pour lesquelles les taux de pixels qui sont correctement appariés par notre méthode partielle, sont plus élevés que ceux qui sont correctement appariés par la méthode standard. À cette fin, nous devons nous concentrer sur la méthode de dématriçage qui estime les composantes de couleur manquant à chaque pixel.

Les deux méthodes analysent la seconde composante couleur (SCC) estimée à chaque pixel alors que la troisième composante couleur (TCC) n'est prise en compte que par la méthode standard. Analyser ou non cette troisième composante couleur estimée est l'une des principales différences entre les méthodes partielle et standard. Ainsi, la qualité de l'estimation de la TCC devrait influer sur la qualité de l'appariement de pixel.

Dans les troisième et quatrième parties, nous mesurerons les performances de mise en correspondance stéréo en calculant le taux de pixels correctement appariés. Ces expériences nécessitent un ajustement de la tolérance sur l'erreur de la disparité afin de déterminer les pixels correctement appariés. En outre, les systèmes de stéréovision dense cherchent à estimer de manière la plus précise possible les cartes de disparité. Dans la cinquième partie, nous comparerons la précision des cartes de disparité estimées grâce aux deux méthodes, en calculant la racine d'erreur quadratique moyenne.

Les deux méthodes sont conçues pour la mise en correspondance de paires d'images stéréo CFA. Cependant, ces images peuvent être corrompues par le bruit d'acquisition.

Dans la sixième partie, nous comparerons la robustesse des performances obtenues par les méthodes standard et partielle contre le bruit d'acquisition.

Enfin, le dispositif stéréo utilisé est composé de caméras couleur équipées d'un capteur unique. Pour être spectralement sensible aux couleurs, une CFA est placée en face de chaque capteur de telle sorte que les caméras fournissent des images CFA. En l'absence de CFA, les caméras délivrent des images en niveaux de gris.

Donc, nous devrons examiner si la présence de CFA à l'intérieur des caméras pour capturer les informations de couleur, améliore ou non la qualité de la mise en correspondance. Pour ce faire, nous comparerons la qualité des pixels correspondant fournie par nos coûts partiels calculés sur les paires des images CFA et celle fournie par les coûts des niveaux de gris calculés sur des images en niveaux de gris dans la dernière partie.

4.1 Introduction

In the previous chapter, we have presented a partial method to match CFA stereo image pairs. To assess the efficiency of our method, we compare its performance with that reached by the standard method explained in section 2.4. For this purpose, we use the well known Middlebury database (http://vision.middlebury.edu/stereo/). Each dataset of this database is made up of a pair of stereo images and the corresponding disparity map.

In these datasets, the color stereo images are acquired by high resolution cameras (Canon G1) equipped with one single-sensor [SS02]. So, the full color images are in fact color images which have been demosaiced by a specific chip inside the camera. They could yet contain artifacts caused by the demosaicing step.

Moreover, applying a demosaicing step on CFA images which have been generated by sampling color components from these previously demosaiced color images, involves applying two successive demosaicing steps on the CFA images acquired by the camera. However, as Middlebury is the most used database for stereovision, we must compare the performances of the tested methods applied to these datasets.

Among the 24 datasets of this database, we have selected four ones ('Cones', 'Poster', 'Tsukuba' and 'Venus') for the experimental study in this chapter. The complete results with all datasets are presented in appendix A.

We also use two stereo datasets designed by Bocquillon, entitled 'Journaux' and 'Plante' and available at http://www.irit.fr/~Benoit.Bocquillon/MYCVR/download.php. The stereo images result from the simulation of image acquisition by three-sensor color cameras which deliver full color images. The left and right color images and disparity map for each stereo dataset examined in this chapter are shown in figure 4.1.

Thanks to the disparity map and given a tolerance about the disparity error, we can determine the pixels that are correctly matched by partial and standard methods applied to a dataset. However, a comparison of the performances based on the global rates of pixels that are correctly matched by the partial and standard methods is not sufficient. Indeed, this does not mean that all pixels which are correctly matched by the first method, are also correctly matched by the second one. So, it is interesting to examine more accurately the spatial locations of pixels that are correctly matched by the two methods.

For this purpose, we divide pixels into different subsets according to the matching results obtained by the partial and standard methods. For comparing performances reached by the two methods, we compare the sizes and the intersection of these subsets. This experimental procedure is detailed in the second section of this chapter.

In the third section of this chapter, we compare the rates of pixels that are correctly matched by following this experimental procedure. Experiments are achieved with the six chosen datasets in order to measure the robustness of performances against the parameter adjustment.

In the fourth section, we attempt to explain the reasons why the rates of pixels that are correctly matched by our partial method, are higher than those that are correctly matched by the standard one. For this purpose, we have to focus on the demosaicing scheme which estimates the missing color components at each pixel.

The two methods analyze the second color component (SCC) estimated at each pixel whereas the third color component (TCC) is taken into account by only the standard one. Analyzing or not this estimated third color component is one of the main differences between the partial and standard methods. So, the quality of TCC estimation should influence the quality of pixel matching.

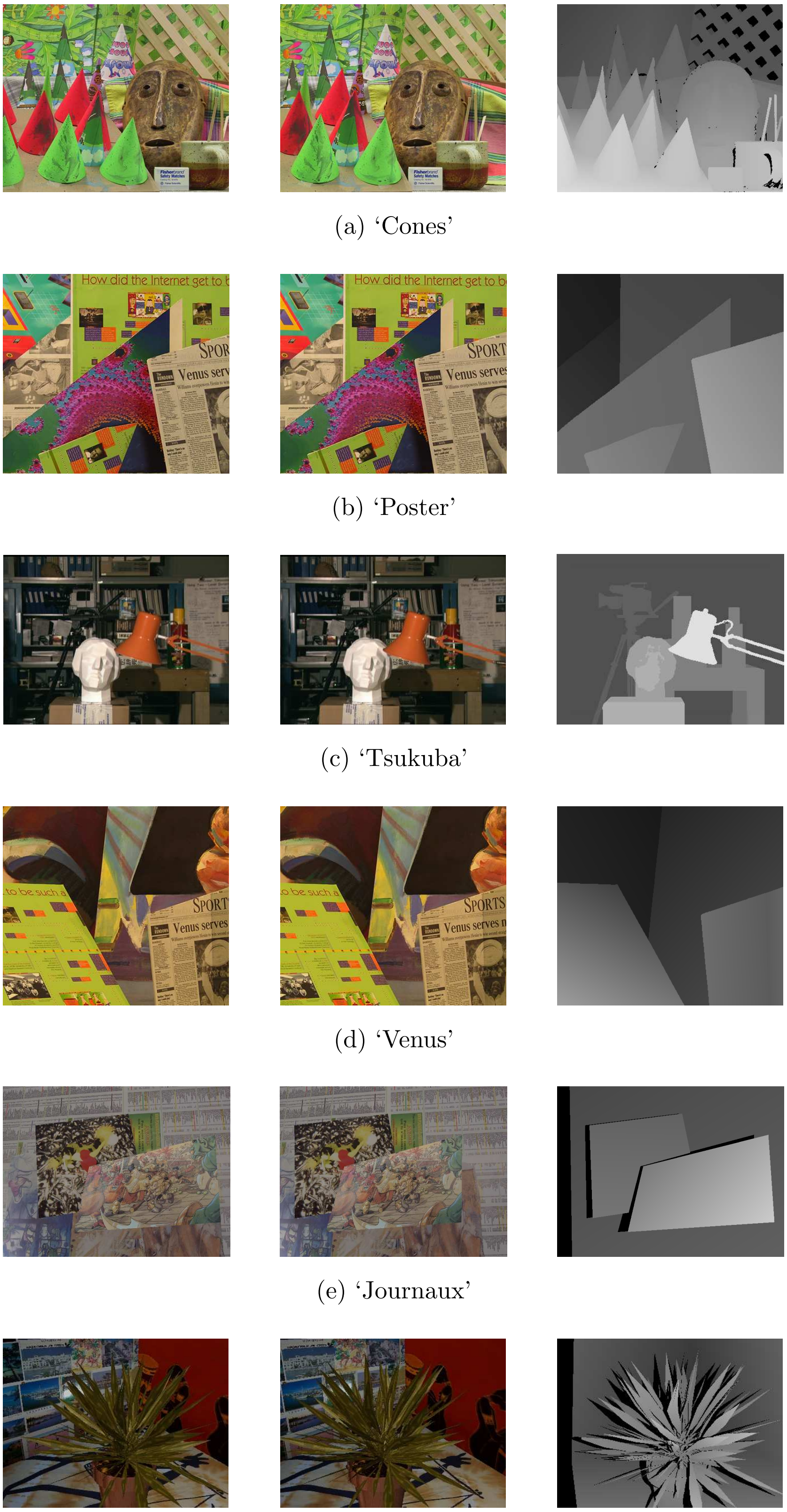
One generally compares the quality of estimation of a color component thanks to its fidelity to its true one in the full color image. For this purpose, we study the correlation between the fidelity of TCC estimation and the performance of pixel matching reached by the two methods.

In the third and fourth sections, we measure the stereo matching performances by computing the rates of correctly matched pixels. These experiments require to adjust the tolerance about disparity error in order to determine the correctly matched pixels. Furthermore, dense stereovision schemes estimate as accurate disparity maps as possible. In the fifth section, we compare the accuracies of the disparity maps estimated thanks to the two methods, by computing the root mean square error.

The two methods are designed to match CFA stereo image pairs. However, these images can be corrupted by acquisition noise. In the sixth section, we compare the robustness of performances reached by partial and standard methods against the acquisition noise.

Finally, the examined stereo device is composed of color cameras equipped with one single sensor. To be spectrally sensitive to the colors, a color filter array is placed in front of each sensor so that the cameras deliver CFA images. When no CFA is added in the cameras, they deliver gray-level images.

So, we should examine if putting a color filter array inside the cameras to capture color information, leads the stereovision schemes to improve their quality of pixel matching. For this purpose, we compare the quality of pixel matching provided by our partial costs computed on CFA stereo image pairs and that provided by gray-level costs computed on gray-level images in the last section.



(a) 'Cones'

(b) 'Poster'

(c) 'Tsukuba'

(d) 'Venus'

(e) 'Journaux'

(f) 'Plante'

Figure 4.1 : (a), (b), (c) and (d) are stereo images taken from Middlebury, (e) and (f) are stereo images taken from Bocquillon. The first and middle columns represent the left and right images, respectively. The last column is the corresponding disparity map.

4.2 Experimental procedure

Since the datasets contain left and right full color images, we have to follow a specific experimental procedure. First we simulate the CFA stereo images, by sampling the available color components according to Bayer's CFA.

Then, we estimate the fully demosaiced and partially demosaiced color images by applying Hamilton's and Dubois' demosaicing methods to the CFA stereo image pairs. Two local stereo algorithms are used to estimate the disparity at each left pixel. The first one is based on the standard matching costs computed on the fully demosaiced color images (standard method) and the second is based on the partial matching costs computed on the partially demosaiced color images (partial method).

The single parameter required by matching costs is the half-width w of the aggregation square window. First, we consider a special case when w is set to 3 and compare the results provided by the two methods. Then, we extend the results for different window sizes.

The two local stereo algorithms provide two estimated disparity maps. Given w, let us denote $\hat{d}^w_{ST}(x_l,y)$ and $\widehat{d}^w_{PA}(x_l,y)$ the disparities estimated by the standard method and our partial method at the left pixel with coordinates (x_l,y) in the left image $\mathbf{I}_l$ of a stereo dataset, respectively.

To compare the performances reached by the standard and partial methods, we divide the left pixels into different disjoint subsets with respect to the quality of matching. Given the disparity error tolerance δ, we can build two different subsets of pixels in the left CFA image :

- subset $\mathcal{ST}^{\delta,w}$: $\Big\{(x_l,y)$ so that $|\hat{d}^w_{ST}(x_l,y) - d(x_l,y)| <= \delta\Big\}$ of left pixels that are correctly matched by the standard method ;

- subset $\mathcal{PA}^{\delta,w}$: $\Big\{(x_l,y)$ so that $|\hat{d}^w_{PA}(x_l,y) - d(x_l,y)| <= \delta\Big\}$ of left pixels that are correctly matched by our partial method.

Since left pixels can be correctly matched by both methods, these two subsets can overlap. Comparing the performances reached by the two methods requires to examine the union and intersection of the subsets.

So, left pixels of the image $\mathbf{I}_l$ can be divided into 4 disjoint subsets :

- subset $\mathcal{PS}^{\delta,w} = \mathcal{PA}^{\delta,w} \cap \mathcal{ST}^{\delta,w}$ of pixels that are correctly matched by both methods

($\mathcal{PS}^{\delta,w}$= Partial Standard) ;

- subset $\mathcal{PO}^{\delta,w} = \mathcal{PA}^{\delta,w} - \mathcal{PA}^{\delta,w} \cap \mathcal{ST}^{\delta,w}$ of pixels that are correctly matched only by our partial method ($\mathcal{PO}^{\delta,w}$= Partial Only) ;

- subset $\mathcal{SO}^{\delta,w} = \mathcal{ST}^{\delta,w} - \mathcal{PA}^{\delta,w} \cap \mathcal{ST}^{\delta,w}$ of pixels that are correctly matched only by the standard method ($\mathcal{SO}^{\delta,w}$ = Standard Only) ;

- subset $\mathcal{NO}^{\delta,w} = \mathbf{I}_l - \mathcal{PA}^{\delta,w} \cup \mathcal{ST}^{\delta,w}$ of pixels that are not correctly matched by both methods ($\mathcal{NO}^{\delta,w}$ = no cost provides good pixel matching).

The union of these 4 subsets is the left image. We propose to compare the sizes of these subsets for the discussion purposes. We generally express the size of a subset as the rate of pixels belonging to it, expressed as its cardinal divided by the number $M \times N$ of pixels in the image multiplied by 100. For example, the rate $\mathcal{SO}^{\delta,w}\%$ of pixels that are correctly matched only by the standard method is expressed as :

$$\mathcal{SO}^{\delta,w}\% = 100.\frac{card(\mathcal{SO}^{\delta,w})}{M.N}. \tag{4.1}$$

The higher $\mathcal{PO}^{\delta,w}\%$ and the lower $\mathcal{SO}^{\delta,w}\%$, the higher the improvement of results brought by our partial method is.

4.3 Rates of correctly matched pixels (RCMP)

4.3.1 Experiments with 'Journaux' dataset

First, we apply the two methods to 'Journaux' dataset (see figure 4.1e). The disparity has been estimated by using SSD matching cost when the window half-width w is set to 3. Table 4.1 shows the sizes of the subsets obtained when the disparity error tolerance δ is set to 0.5.

In this table, we focus on the size of the subset of pixels matched only by our partial method ($\mathcal{PO}^{0.5,3}$) and that of pixels matched only by standard one ($\mathcal{SO}^{0.5,3}$). The highest size between these two subsets is marked as bold. The size of these two subsets

Table 4.1 : Size of left pixel subsets of 'Journaux' dataset using SSD cost.

Demosaicing	pixels correctly matched by both methods $\mathcal{PS}^{0.5,3}$ %	pixels correctly matched by partial method only $\mathcal{PO}^{0.5,3}$ %	pixels correctly matched by standard method only $\mathcal{SO}^{0.5,3}$ %	pixels not matched correctly by both methods $\mathcal{NO}^{0.5,3}$ %
Hamilton	84.40	**2.53**	.70	12.20
Dubois	86.50	**1.04**	.63	11.70

is low compared with that of pixels which are correctly matched by both methods. As we apply a simple stereovision algorithm that does not analyze half-occlusions and depth discontinuities, at least 11% of left pixels are never correctly matched (subset $\mathcal{NO}^{0.5,3}$).

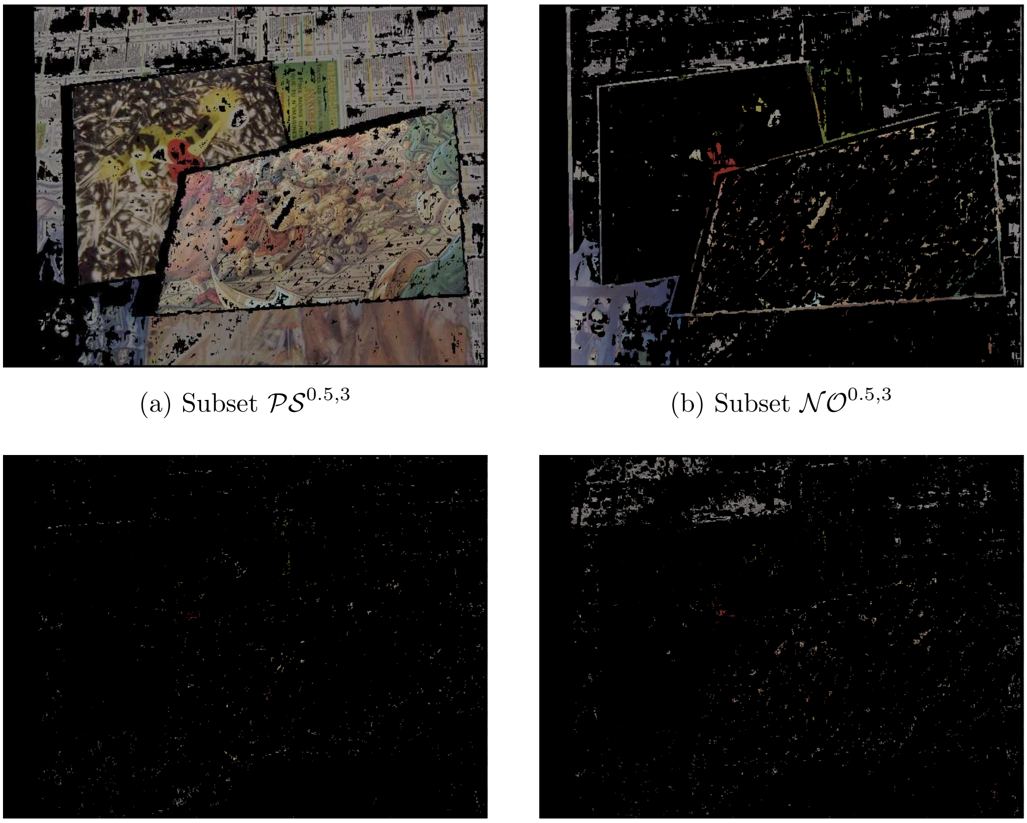
For the two used demosaicing schemes, the population of left pixels which are correctly matched only by the partial method ($\mathcal{PO}^{0.5,3}$) is higher than that of pixels which are correctly matched by only the standard one ($\mathcal{SO}^{0.5,3}$).

By examining figure 4.2 which displays the different subsets, we see that left pixels which are never correctly matched subset ($\mathcal{NO}^{0.5,3}$), are mainly located at transition areas between objects. Moreover, left pixels that are correctly matched only by one of the two matching methods, are scattered through the image. So, there is no specific location of pixels belonging to $\mathcal{SO}^{0.5,3}$ or to $\mathcal{PO}^{0.5,3}$.

4.3.2 Comparison between RCMP with fixed aggregation window size

We compare the size of subset $\mathcal{SO}^{0.5,3}$ of pixels that are correctly matched only by the standard method and the size of subset $\mathcal{PO}^{0.5,3}$ of pixels that are correctly matched only by our partial method. For this purpose, we use SSD, SAD and NCC matching costs. Let us examine the results shown in tables 4.2 and 4.3 with the six datasets of figure 4.1 for Hamilton's and Dubois' demosaicing schemes, respectively. At each comparison, the value printed in bold typeface highlights the best result.

Tables 4.2 and 4.3 show that except for 'Cones' dataset, $\mathcal{PO}^{0.5,3}$% is higher than



(a) Subset $\mathcal{PS}^{0.5,3}$

(b) Subset $\mathcal{NO}^{0.5,3}$

(c) Subset $\mathcal{SO}^{0.5,3}$

(d) Subset $\mathcal{PO}^{0.5,3}$

Figure 4.2 : Subsets of left pixels of 'Journaux' dataset using SSD cost (Hamilton demosaicing scheme).

Table 4.2 : RCMP by partial method only ($\mathcal{PO}^{0.5,3}\%$) and by standard method only ($\mathcal{SO}^{0.5,3}\%$) (Hamilton's demosaicing method).

Matching cost	SSD		SAD		NCC	
Dataset	$\mathcal{PO}^{0.5,3}\%$	$\mathcal{SO}^{0.5,3}\%$	$\mathcal{PO}^{0.5,3}\%$	$\mathcal{SO}^{0.5,3}\%$	$\mathcal{PO}^{0.5,3}\%$	$\mathcal{SO}^{0.5,3}\%$
Plante	**3.23**	2.30	**3.54**	2.66	**3.22**	2.28
Journaux	**2.53**	0.70	**2.11**	0.76	**2.74**	0.71
Cones	1.24	**1.29**	1.31	**1.77**	**1.25**	0.98
Poster	**2.33**	1.71	**2.30**	1.91	**2.39**	1.89
Tsukuba	**2.07**	0.93	**2.11**	1.03	**2.36**	1.01
Venus	**2.32**	1.91	**2.53**	2.15	**2.20**	2.01

Table 4.3 : RCMP by partial method only ($\mathcal{PO}^{0.5,3}\%$) and by standard method only ($\mathcal{SO}^{0.5,3}\%$) (Dubois' demosaicing method).

Matching cost / Dataset	SSD		SAD		NCC	
	$\mathcal{PO}^{0.5,3}\%$	$\mathcal{SO}^{0.5,3}\%$	$\mathcal{PO}^{0.5,3}\%$	$\mathcal{SO}^{0.5,3}\%$	$\mathcal{PO}^{0.5,3}\%$	$\mathcal{SO}^{0.5,3}\%$
Plante	**2.04**	1.80	**2.74**	2.07	**1.91**	1.83
Journaux	**1.04**	0.63	**1**	0.79	**1.15**	0.69
Cones	0.95	**1**	1.02	**1.73**	**0.95**	0.62
Poster	**1.40**	0.96	**1.43**	1.20	**1.26**	0.99
Tsukuba	**1.23**	0.66	**1.20**	0.82	**1.36**	0.73
Venus	**1.31**	1.20	1.45	**1.50**	**1.23**	1.20

$\mathcal{SO}^{0.5,3}\%$, whatever the used demosaicing scheme and matching cost. So, the partial approach outperforms the standard one except for 'Cones' dataset with SSD, SAD matching cots. So, matching performance using NCC_{PA} partial matching cost is always better than the standard matching costs in our experiment.

The complete results for the 26 datasets are included in appendix A in tables A.1 and A.2. For 24 datasets, the partial method provides better results than the standard one. So, these extensive experiments show that when the window half-width size w is set to 3 and the error tolerance δ is set to 0.5, our partial method provides better results than those provided by the standard one.

4.3.3 Comparison between rates of correctly matched pixels

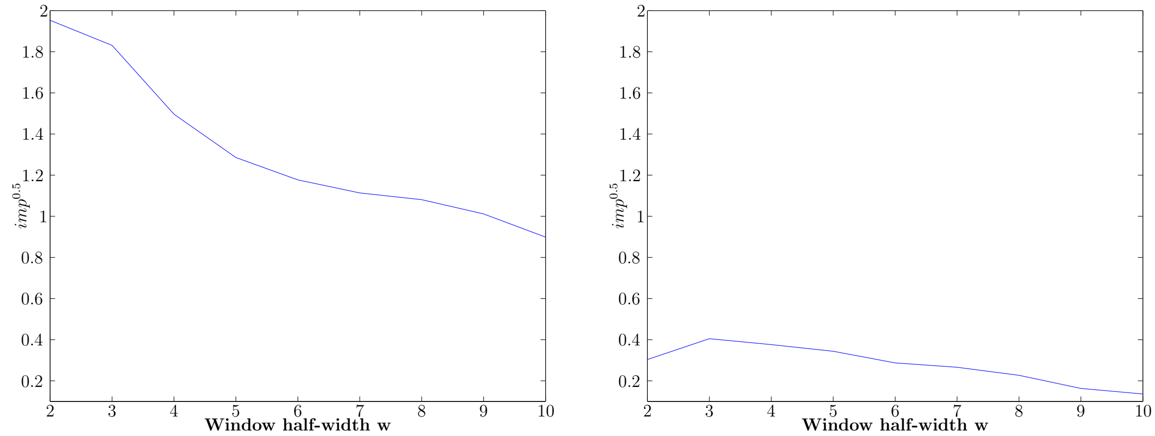
Now, let us examine the behavior of both approaches with respect to w. For this purpose, we propose to compute statistics for comparing the results provided by both tested methods. Let us define the improvement $Imp^{\delta,w}$ as the difference between the rate $\mathcal{PO}^{\delta,w}\%$ of correctly matched pixels using partial method and the rate $\mathcal{PO}^{\delta,w}\%$ of correctly matched pixels using standard one for a given window half-width w and a disparity error tolerance δ. It is expressed as :

$$Imp^{\delta,w} = (\mathcal{PO}^{\delta,w}\% - \mathcal{SO}^{\delta,w}\%). \qquad (4.2)$$

A positive value of $Imp^{\delta,w}$ means that the partial method outperforms the standard one.

Taking 'Journaux' dataset, $Imp^{0.5,w}$ with respect to w and when using SSD matching cost is shown in figure 4.3. We see that the improvement varies but is always positive. Since it is fastidious to examine such a figure for the other datasets, we propose to compute the mean improvement brought by our method instead of showing figures. The mean improvement measure for a given δ is formulated as :

$$\overline{Imp}^{\delta} = \frac{1}{9}\sum_{w=2}^{10} Imp^{\delta,w}. \tag{4.3}$$



(a) Hamilton's demosaicing- SSD matching cost. (b) Dubois' demosaicing- SSD matching cost.

Figure 4.3 : $imp^{0.5}$,w for 'Journaux' dataset.

Tables 4.4 and 4.5 display $\overline{Imp}^{\delta}$ when using Hamilton's and Dubois' demosaicing methods, respectively. We compute the improvement when δ is set to 0.5 and 1. Since most of the examined datasets have a disparity map with subpixel accuracy and since the retained stereovision scheme cannot provide subpixel disparity estimation, we do not examine the case when δ is set to 0. The positive values are marked in bold.

$\overline{Imp}^{\delta}$ is positive with most of the test datasets using three matching costs and the two demosaicing methods, except for 'Cones' dataset. As for the overall results presented in tables A.3 and A.4 in appendix A, the mean improvement is positive for 23 datasets when the Hamilton's demosaicing method is used and for 22 datasets when Dubois' demosaicing method is used. These experimental results prove the improvement brought by our partial

Table 4.4 : Mean improvement $\overline{Imp}^{\delta}$ (Hamilton's demosaicing method).

	$\delta = 0.5$			$\delta = 1$		
Matching cost Dataset	SSD	SAD	NCC	SSD	SAD	NCC
Plante	**0.24**	**0.24**	**0.24**	**0.59**	**0.58**	**0.53**
Journaux	**0.66**	**0.48**	**0.75**	**1.31**	**1**	**1.44**
Cones	-0.11	-0.33	**0.04**	-0.14	-0.36	**0.01**
Poster	**0.34**	**0.37**	**0.34**	**0.46**	**0.35**	**0.39**
Tsukuba	**0.77**	**0.67**	**0.82**	**0.77**	**0.68**	**0.83**
Venus	**0.30**	**0.19**	**0.16**	**0.47**	**0.31**	**0.30**

Table 4.5 : Mean improvement $\overline{Imp}^{\delta}$ (Dubois' demosaicing method).

	$\delta = 0.5$			$\delta = 1$		
Matching cost Dataset	SSD	SAD	NCC	SSD	SAD	NCC
Plante	**0.22**	**0.23**	**0.21**	**0.55**	**0.55**	**0.48**
Journaux	**0.67**	**0.49**	**0.76**	**1.34**	**1.02**	**1.47**
Cones	-0.12	-0.33	**0.04**	-0.15	-0.35	**0.01**
Poster	**0.34**	**0.36**	**0.34**	**0.45**	**0.34**	**0.38**
Tsukuba	**0.77**	**0.66**	**0.83**	**0.77**	**0.68**	**0.84**
Venus	**0.28**	**0.18**	**0.15**	**0.46**	**0.29**	**0.30**

demosaicing approach. Although the mean improvement rates are low (less than 1%), we should not forget that our partial approach is less time consuming than the standard one.

4.4 Fidelity of estimated TCC and matching performance

The main difference between the standard and our partial method is the analysis of the third color component. The standard method takes into account TCC whereas the partial one neglects it. As the preceding results show that the partial method outperforms the standard one, we propose to compare the estimation quality of SCC and TCC thanks to the $PSNR$ criterion. In this section, we study the relationship between the quality estimation of the third color component TCC by a demosaicing scheme and the rate of correctly matched pixels. For this purpose, we first achieve a global analysis of the estimation quality of TCC. Then, we combine the study of estimation quality and matching performance by measuring $PSNR$ inside the subsets $\mathcal{PO}^{\delta,w}$ and $\mathcal{SO}^{\delta,w}$.

4.4.1 Global analysis

Table 4.6 displays the $PSNR$ of SCC and TCC, estimated by Hamilton's and Dubois' demosaicing schemes applied to the left CFA images of the six examined stereo datasets. At each comparison, the value printed in bold typeface highlights the best result. We see that $PSNR$ of SCC is always higher than that obtained by the estimation of the third color component TCC. Hence, the quality of estimation of the third color component is lower than that of the second component. Moreover, table 4.7 shows the $PSNR$ of the estimated color components R, G, and B computed with the same images. As shown by Yang *et al.* [LMY10], the fidelity of color component G estimation provides the highest PSNR.

Moreover, by comparing tables 4.6 and 4.7, we see that the $PSNR$ of the third color component TCC is always the lowest one. So, the estimation fidelity of the third color component is vey bad compared with that of the other components. The same conclusions arise from the analysis of the complete tables presented in tables A.5 and A.6 of appendix A.

So, it is interesting to study in depth the relationship between the bad estimation fidelity of TCC and the pixel matching performance.

4.4.2 Subset analysis

A global study across the left image is not sufficient to measure the relationship between the fidelity of color estimation and matching performance. So, we propose to measure the fidelity of color estimation across the four subsets $\mathcal{PS}^{\delta,w}$, $\mathcal{PO}^{\delta,w}$, $\mathcal{SO}^{\delta,w}$ and $\mathcal{NO}^{\delta,w}$ that can be built from the comparison of matching performances reached by the standard

Table 4.6 : $PSNR$ of SCC, TCC estimated by Hamilton's and Dubois' demosaicing methods applied to left images.

Demosaicing Dataset	Hamilton		Dubois	
	$PSNR^{SCC}$	$PSNR^{TCC}$	$PSNR^{SCC}$	$PSNR^{TCC}$
Plante	**34.76**	30.07	**36.81**	32.87
Journaux	**35.39**	30.36	**38.86**	34.6
Cones	**32.27**	28.24	**31.34**	27.23
Poster	**30.55**	25.79	**29.35**	24.97
Tsukuba	**37.48**	32.28	**37.23**	32.15
Venus	**34.44**	29.41	**33.49**	29.18

Table 4.7 : $PSNR$ of R, G, and B estimated by Hamilton's and Dubois' demosaicing methods applied to left images.

Demosaicing dataset	Hamilton			Dubois		
	$PSNR^{R}$	$PSNR^{G}$	$PSNR^{B}$	$PSNR^{R}$	$PSNR^{G}$	$PSNR^{B}$
Plante	31.65	**32.49**	31.98	34.29	**36.48**	34.53
Journaux	32.20	**33.32**	32.17	36.24	**38.86**	36.22
Cones	29.74	**33.87**	29.86	28.71	**31.45**	28.93
Poster	26.98	**31.55**	28.22	26.21	**29.36**	27.09
Tsukuba	33.30	**38.24**	35.19	33.18	**38.31**	34.98
Venus	31.29	**34.26**	31.17	30.93	**33.02**	30.72

and partial methods.

The subsets $\mathcal{PS}^{\delta,w}$ and $\mathcal{NO}^{\delta,w}$ regroup left pixels that have been correctly matched by both methods and by no method, respectively. At these pixels, the TCC estimation quality does not impact the quality of matching. So, we do not examine these subsets.

At pixels belonging to $\mathcal{SO}^{\delta,w}$, taking into account TCC improves the quality of matching whereas at pixels belonging to $\mathcal{PO}^{\delta,w}$, taking into account TCC decreases the quality of matching. So, we propose to examine $PSNR$ of TCC at pixels belonging to one of these two subsets.

4.4.2.1 $PSNR$ of TCC

Tables 4.8 and 4.9 display $PSNR$ of TCC at pixels belonging to $\mathcal{PO}^{0.5,3}$ and $\mathcal{SO}^{0.5,3}$, when Hamilton's and Dubois' demosaicing schemes are applied, respectively. At each comparison, the value printed in bold typeface highlights the highest PSNR. We see that $PSNR$ of TCC at pixels belonging to $\mathcal{SO}^{0.5,3}$ is always higher than that of pixels belonging to $\mathcal{PO}^{0.5,3}$, except for the SSD matching cost used to match pixels of 'Tsukuba' dataset. These tables show that PSNR value strongly changes according to the considered subsets of pixels. So, the quality of TCC estimation is not constant across the image.

Moreover, by comparing $PSNR$ of TCC across the image (see table 4.6) and at pixels belonging to $\mathcal{SO}^{0.5,3}$ (see tables 4.8 and 4.9), we see that $PSNR$ of TCC is always higher at pixels that are well matched by the standard matching than across the image. So, the estimation quality of pixels belonging to $\mathcal{SO}^{0.5,3}$ is higher than for the other pixels.

We can conclude that the estimation of TCC is good enough in subset $\mathcal{SO}^{0.5,3}$ to improve matching performance. In other subsets, the estimation of TCC is so bad that it does not impact or decreases the matching quality. We obtain the same conclusions with the images of the other datasets (see tables A.7 and A.8 in appendix A).

Table 4.8 : PSNR of TCC estimated by Hamilton's demosaicing method at left pixels belonging to different subsets.

Matching cost	SSD		SAD		NCC	
Subset Dataset	$\mathcal{PO}^{0.5,3}$	$\mathcal{SO}^{0.5,3}$	$\mathcal{PO}^{0.5,3}$	$\mathcal{SO}^{0.5,3}$	$\mathcal{PO}^{0.5,3}$	$\mathcal{SO}^{0.5,3}$
Plante	29.5	**31.9**	30.3	**32.7**	29.1	**31.5**
Journaux	25.3	**31.6**	25.8	**32.6**	25.4	**31.5**
Cones	28	**30**	28	**30.3**	27.4	**30.1**
Poster	25.2	**26.8**	25.5	**27**	25	**27.5**
Tsukuba	**35.1**	33.9	33.6	**35.8**	34.4	**34.9**
Venus	29.8	**32.3**	30.4	**32.3**	29.8	**32.1**

Table 4.9 : PSNR of TCC estimated by Dubois' demosaicing method at left pixels belonging to different subsets.

Matching cost	SSD		SAD		NCC	
Subset	$\mathcal{PO}^{0.5,3}$	$\mathcal{SO}^{0.5,3}$	$\mathcal{PO}^{0.5,3}$	$\mathcal{SO}^{0.5,3}$	$\mathcal{PO}^{0.5,3}$	$\mathcal{SO}^{0.5,3}$
Dataset						
Plante	34	**35.1**	33.8	**34.5**	33.7	**34.5**
Journaux	30.2	**36.5**	31.3	**36.9**	30.5	**35.6**
Cones	27.3	**28.6**	27.4	**28.8**	27.4	**27.6**
Poster	24.6	**27**	24.7	**26.2**	24.5	**27.4**
Tsukuba	**34.4**	33.2	33.8	**34.4**	34.4	**35.6**
Venus	30.7	**33**	32.3	**32.5**	31	**33.4**

4.4.2.2 Pixel matching based on *TCC* Only

In order to prove that the third estimated color component TCC mainly causes the difference between the two subsets $\mathcal{PO}$ and $\mathcal{SO}$, we propose to match the pixels by taking into account only it.

SSD, SAD and NCC matching cost can be computed when using only the TCC component of the image pairs. In this case, the gray-level version of these costs is computed. However, in order to follow the same notation as that used in chapter 2, we will replace the subscript g in the gray-level versions of these costs with the subscript TCC. So, the generic notation of these costs will be $COST_{TCC}$.

We focus on the subsets $\mathcal{PO}^{0.5,3}$ and $\mathcal{SO}^{0.5,3}$ of left pixels. Their sizes have been displayed in tables 4.2 and 4.3. Among the pixels of a subset, we propose to count those which are correctly matched by the matching cost $COST_{TCC}$. Tables 4.10 and 4.11 (and tables A.9 and A.10 in appendix A) show the rate of pixels belonging to the subset $\mathcal{PO}^{0.5,3}$ or $\mathcal{SO}^{0.5,3}$, that are correctly matched by the matching cost $COST_{TCC}$ (w set to 3 and δ set to 0.5). The highest rate between these two subsets is marked as bold.

First, we notice that whatever the used matching cost, the performances are very close. Then, we notice that among the pixels which are correctly matched only by partial method ($\mathcal{PO}^{0.5,3}$), a very few pixels (often less than 3) are also correctly matched by taking into account only the third color component TCC.

On the opposite, most of pixels of the subset $\mathcal{SO}^{0.5,3}$, i.e. pixels which are correctly matched only by the standard method, are also correctly matched by taking into account only the third color component TCC. This phenomenon can be explained by the quality of TCC estimation inside these two subsets. Indeed, high $PSNR$ of TCC inside subset $\mathcal{SO}^{0.5,3}$ (see tables 4.8 and 4.9) indicates that the third color component TCC is well estimated at these pixels. So, taking it into account provides good quality of pixel matching inside the subset $\mathcal{SO}^{0.5,3}$. On the opposite, as the $PSNR$ of TCC is low inside subset $\mathcal{PO}^{0.5,3}$, the fidelity of estimation of TCC is not good enough to correctly match these pixels.

Experiments in section 4.3.3 show that the number of pixels which are correctly matched by our partial method is higher than that of pixels which are correctly matched by the standard one. The partial method does not take into account the estimated third color component TCC whereas the standard one does. This result can be explained by the low quality of TCC estimation provided by the demosaicing schemes which are yet the most efficient ones. The TCC is so badly estimated that it often decreases the performance of pixel matching.

To provide a more detailed analysis of the disparity maps calculated using the partial and standard methods, a statistical study will be done in the next section.

Table 4.10 : Rate of pixels belonging to different subsets that are correctly matched by $COST_{TCC}$ (TCC estimated by Hamilton's demosaicing method) (δ set to 0.5 and w set to 3).

Matching cost	SSD		SAD		NCC	
Subset Dataset	$\mathcal{PO}^{0.5,3}$	$\mathcal{SO}^{0.5,3}$	$\mathcal{PO}^{0.5,3}$	$\mathcal{SO}^{0.5,3}$	$\mathcal{PO}^{0.5,3}$	$\mathcal{SO}^{0.5,3}$
Plante	0	**77.4**	0	**77.6**	.04	**74.3**
Journaux	.08	**83.9**	.23	**83.7**	.25	**84.7**
Cones	.11	**90.2**	.26	**90.4**	1.68	**85.3**
Poster	.09	**83.8**	.06	**82.3**	.62	**84.8**
Tsukuba	0	**81.2**	0	**82.1**	.22	**81.7**
Venus	0	**82.4**	.24	**80.4**	.37	**80.9**

Table 4.11 : Rate of pixels belonging to different subsets that are correctly matched by $COST_{TCC}$ (TCC estimated by Dubois' demosaicing method) (δ set to 0.5 and w set to 3).

Matching cost	SSD		SAD		NCC	
Subset	$\mathcal{PO}^{0.5,3}$	$\mathcal{SO}^{0.5,3}$	$\mathcal{PO}^{0.5,3}$	$\mathcal{SO}^{0.5,3}$	$\mathcal{PO}^{0.5,3}$	$\mathcal{SO}^{0.5,3}$
Dataset						
Plante	0	**78.5**	0	**80.2**	.12	**77.5**
Journaux	.33	**85.9**	.57	**86**	.96	**86.7**
Cones	.07	**91.8**	.47	**92.5**	1.07	**86.2**
Poster	.1	**86.8**	0	**83.3**	1.58	**87.3**
Tsukuba	0	**91.3**	0	**90.1**	0	**88.4**
Venus	.05	**89.4**	.09	**87.3**	.65	**86.3**

4.5 Root mean square error

Since all datasets used in our experiments contain the disparity maps, we can calculate the error on the disparity estimation at each left pixel. So, for each stereo image pair, we compute statistics on disparity error for all the pixels. Thus, we compare the performances reached by the standard and partial methods with respect to the accuracy of the estimated disparity.

For this purpose, we process the root mean square error $RMSE^w$ [SS02] (measured in disparity units) between the estimated disparity $\hat{d}^w(x_l,y)$ provided by a matching method using an aggregation window whose size is $(2w+1) \times (2w+1)$ and the disparity $d(x_l,y)$:

$$RMSE^w = \left(\frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left(\hat{d}^w(x_l,y) - d(x_l,y) \right)^2 \right)^{\frac{1}{2}} . \tag{4.4}$$

A low level of $RMSE$ indicates that the accuracy of the estimated disparity map is high.

We propose to compare the partial and standard methods thanks to $RMSE$. Tables 4.12 and 4.13 show $RMSE^3$ computed with the 6 retained datasets using Hamilton's and Dubois' demosaicing schemes, respectively. The lowest error between these two methods is marked as bold, for each of three matching costs SSD, SAD and NCC. By examining these tables, we see that $RMSE^3$ is very high (close to 10) for 'Plante', 'Journaux' and 'Cones' datasets. This is caused by the retained simple stereovision scheme which does not notably analyze the half-occlusion cases. Moreover, the aggregation window size depending on w is too low to provide satisfying results for complex scenes. However, that does not matter since we want only to compare the performances of the two tested approaches.

When SSD matching cost is used, $RMSE^3$ measured with partial method is lower than that measured with standard one for all datasets when Hamilton's demosaicing scheme is used and 5 among 6 when Dubois' demosaicing scheme is used.

For SAD matching cost, $RMSE^3$ measured with partial method is lower than that measured with standard one for all datasets when Hamilton's demosaicing scheme is used and 4 among 6 when Dubois' demosaicing scheme is used.

Finally, when colors (or partial colors) of pixels are compared thanks to NCC matching cost, standard and partial methods provide similar results.

Results with all the datasets (see tables A.11 and A.12) show that for 95 over 156 studied cases (since we use 26 datasets, three matching costs and two demosaicing schemes, we have 156 cases), $RMSE^3$ measured on the disparity computed thanks to our partial method is lower than that measured thanks to the standard one.

Table 4.12 : $RMSE^3$ (Hamilton's demosaicing method).

Matching cost Images	SSD_{PA}	SSD_{ST}	SAD_{PA}	SAD_{ST}	NCC_{PA}	NCC_{ST}
Plante	**10.4**	10.6	**9.47**	9.64	**12.4**	12.5
Journaux	**8.46**	9.18	**8.58**	9.08	**8.44**	9.49
Cones	**8.39**	8.4	**8.32**	8.34	8.88	**8.84**
Poster	**3.25**	3.31	**3.26**	3.32	**3.5**	3.5
Tsukuba	**1.95**	2.02	**1.9**	1.95	**2.14**	2.21
Venus	**3.91**	3.93	**3.88**	3.88	4.26	**4.2**

Table 4.13 : $RMSE^3$ (Dubois' demosaicing method).

Matching cost Dataset	SSD_{PA}	SSD_{ST}	SAD_{PA}	SAD_{ST}	NCC_{PA}	NCC_{ST}
Plante	**10.4**	10.4	**9.41**	9.52	12.4	**12.3**
Journaux	**8**	8.08	8.15	**8.14**	**7.62**	7.69
Cones	8.24	**8.23**	8.23	**8.19**	8.78	**8.72**
Poster	**3.15**	3.2	**3.16**	3.22	**3.41**	3.41
Tsukuba	**1.99**	2.02	**1.94**	1.97	**2.19**	2.22
Venus	**3.92**	3.92	**3.82**	3.82	4.32	**4.25**

4.6 Robustness against noise

As the partial and standard methods are designed to match CFA stereo image pairs, we have to measure its robustness against the acquisition noise. In order to test the behavior of our partial approach in presence of noise within CFA images, we use a simplified version of the noise model developed by Irie *et al.* [IMUW08] and dedicated to CCD sensors. First, we simulate the CFA sampling in order to produce the left and right 'Journaux' CFA images. Then, we separately corrupt the CFA images by two different kinds of noise, i.e. additive and multiplicative non-correlated gaussian noise.

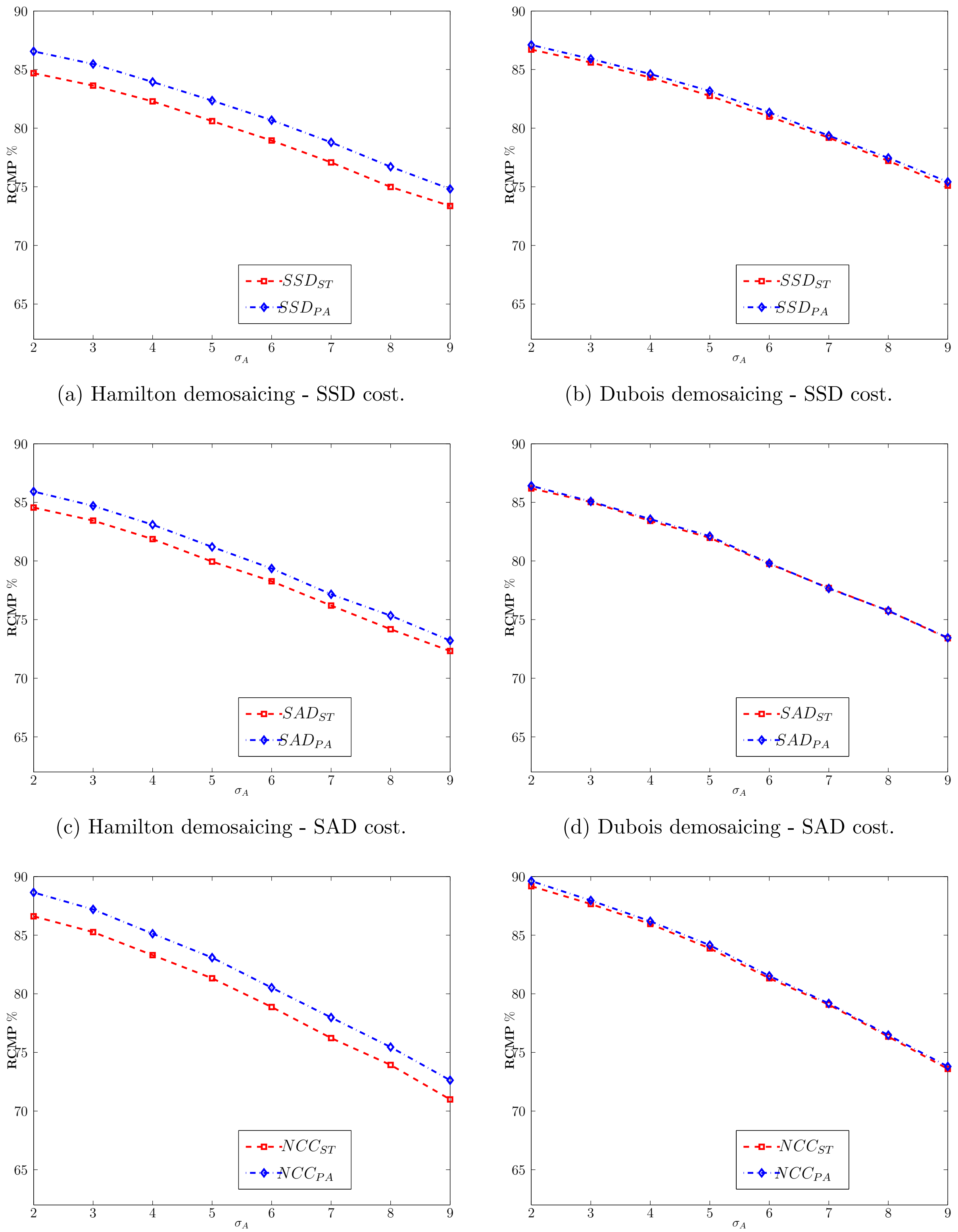
In the first case, we add non-correlated gaussian noise with a standard deviation σ_A. In the second case, we add the level of each pixel with its product with a white multiplicative gaussian noise whose standard deviation is σ_M.

First, the noisy demosaiced and partially demosaiced color images are determined by Hamilton's or Dubois' demosaicing schemes. Then, partial matching costs are computed on the noisy partially demosaiced color images and the standard ones are computed on the demosaiced color images.

The rates of correctly matched pixels (δ set to 0.5, w set to 3) with respect to the standard deviations σ_A and σ_M are shown by figures 4.4 and 4.5, respectively.

In each subfigure of these figures, the red dotted curve corresponds to the RCMP in the noisy demosaiced color image pair, using $COST_{ST}$ for cost computation while the blue dotted curve corresponds to the RCMP in the noisy partially demosaiced color image pair, using $COST_{PA}$ for cost computation. The left column shows results for Hamilton's demosaicing scheme and the right column for Dubois' scheme. The three lines correspond respectively to the SAD, SSD and NCC matching costs.

For the Hamilton's demosaicing scheme, the difference between RCMP in noisy partially demosaiced color images and noisy fully demosaiced color images ranges between 0.5% and 2%. For the Dubois' demosaicing scheme, this difference ranges between 0% and 0.5%. We can clearly see that the partial method outperforms standard one for all the tested σ_A and σ_M values.



(a) Hamilton demosaicing - SSD cost.

(b) Dubois demosaicing - SSD cost.

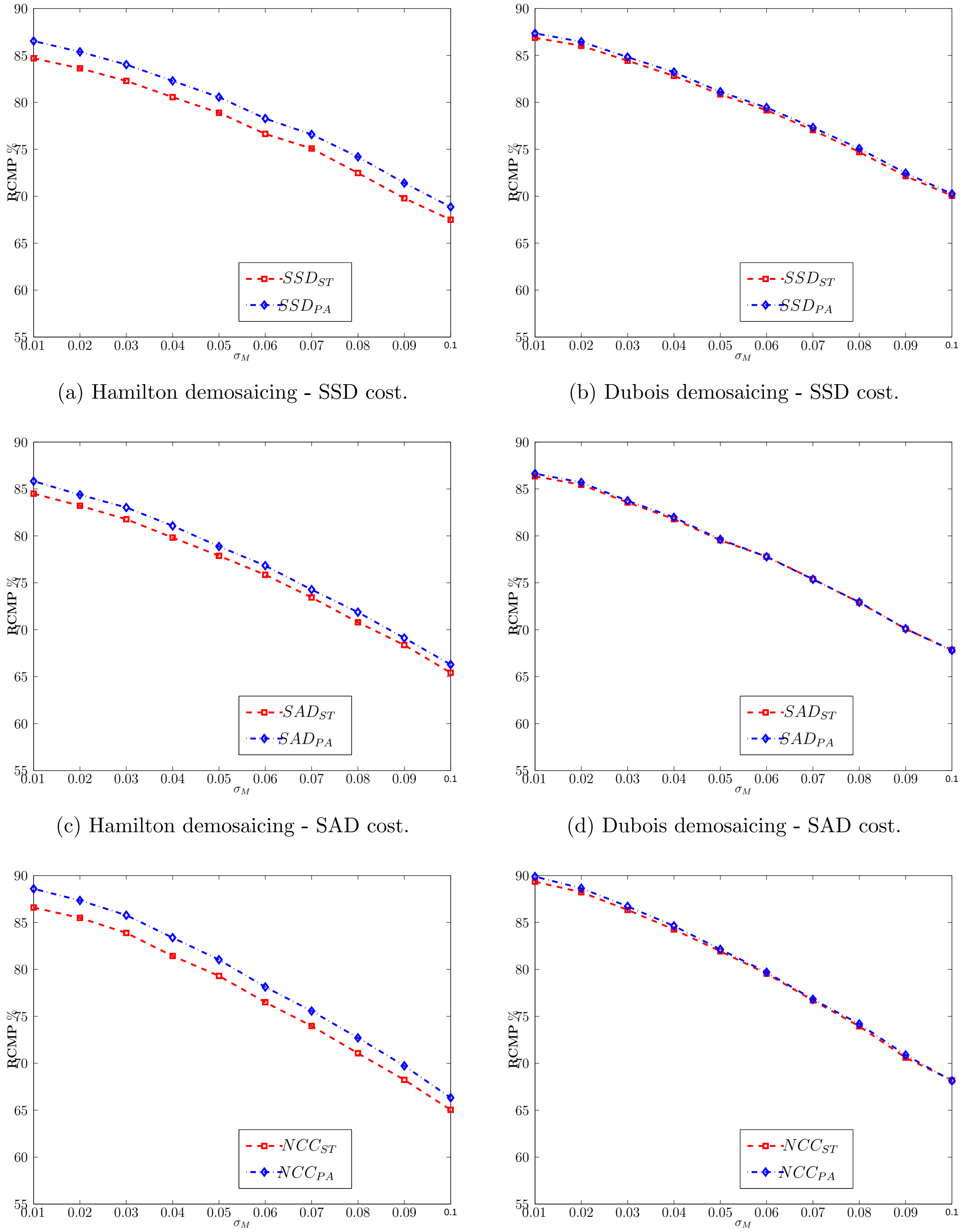
(c) Hamilton demosaicing - SAD cost.

(d) Dubois demosaicing - SAD cost.

(e) Hamilton demosaicing - NCC cost.

(f) Dubois demosaicing - NCC cost.

Figure 4.4 : Rate of correctly matched pixels (RCMP) by three matching costs computed on 'Journaux' stereo image pair corrupted with additive noise for δ set to 0.5 and w set to 3.



(a) Hamilton demosaicing - SSD cost.

(b) Dubois demosaicing - SSD cost.

(c) Hamilton demosaicing - SAD cost.

(d) Dubois demosaicing - SAD cost.

(e) Hamilton demosaicing - NCC cost.

(f) Dubois demosaicing - NCC cost.

Figure 4.5 : Rate of correctly matched pixels (RCMP) by three matching costs computed on 'Journaux' stereo image pair corrupted with multiplicative noise for δ set to 0.5 and w set to 3.

4.7 Single-CCD or Gray-level cameras for stereo matching

In this work, we consider that the stereo device is composed of cameras equipped with single-CCD sensors. To be spectrally sensitive to color primaries, a color filter array is placed in front of the sensor. Our partial method is designed to match pixels from images generated by this kind of cameras. However, we should examine if putting a color filter array inside the cameras helps the stereovision schemes to improve their quality of pixel matching.

When the camera is not equipped with a color filter array, pixels whose coordinates are (x,y), are characterized by gray-levels $I(x,y)$. From a full color image, we can estimate the gray-level at each pixel using equation (2.26).

We propose to compare the performances reached by a dense stereovision scheme which analyses CFA images delivered by color cameras equipped with single-CCD and those reached by a dense stereovision which analyses gray-level images delivered by gray-level cameras.

Two local stereo algorithms are used to estimate the disparity at each left pixel. The first one is based on the partial matching costs computed on the partially demosaiced color images (Partial method) and the second one is based on gray-level matching costs computed on the gray-level images (Gray-level method). For this purpose, we examine RCMP when w is set to 3 and δ is set to 0.5 and $RMSE^3$.

Tables 4.14 and 4.15 show the rates of pixels that are correctly matched by the two compared methods. For these experiments, w is set to 3 and δ is set to 0.5, as for experiments detailed in section 4.3. At each comparison, the value printed in bold typeface highlights the best result.

For the three used matching costs, RCMP in partially demosaiced color images is higher than those of gray-level images for 2 among 6 datasets when Hamilton's or Dubois' schemes are used.

Results with all datasets (see tables A.13 and A.14) show that for 127 cases (we studied 156 cases), RCMP in partially demosaiced color images is lower than that in gray-level images. So, when the aggregation window half-width is small, the gray-level approach outperforms the partial one.

In order to confirm these results, we also compare the accuracy of the disparity estimation provided by the partial and gray-level methods. Tables 4.16 and 4.17 show $RMSE^3$ when using both methods.

When SSD matching cost is used, $RMSE^3$ with partially demosaiced color images is lower than those with gray-level images in 3 among 6 datasets for Hamilton's scheme and 4 among 6 for Dubois' schemes. When SAD matching cost is used, the two methods provide similar results. Finally, when NCC matching cost is used, $RMSE^3$ obtained with partially demosaiced color images is lower than those of gray-level images for all the retained datasets for Hamilton's scheme and for 5 among 6 for Dubois' scheme.

Results with all datasets (see tables A.15 and A.16) show that for 107 over 156 studied cases, $RMSE^3$ measured on the disparity computed using partially demosaiced images is lower than that measured using gray-level images.

We can conclude from these results that matching using gray-level images generally provides higher RCMP than matching using partially demosaiced color images. However, the accuracy of the estimated disparity map using gray-level images is lower than that calculated using partially demosaiced color images.

Table 4.14 : RCMP (Hamilton's demosaicing method used by $COST_{PA}$, w is set to 3 and δ is set to 0.5).

Matching cost Images	SSD_{PA}	SSD_g	SAD_{PA}	SAD_g	NCC_{PA}	NCC_g
Plante	45.51	**45.53**	49.17	**50.31**	**42.78**	41.88
Journaux	**87**	86.45	**86.4**	85.89	**89.25**	88.52
Cones	67.19	**72.26**	65.99	**70.71**	70.92	**76.11**
Poster	61.89	**70.83**	61.53	**69.19**	62.66	**74.35**
Tsukuba	**69.77**	67.46	**69.82**	65.81	**69.75**	67.15
Venus	67.68	**75.73**	67.54	**74.66**	66.96	**76.28**

Table 4.15 : RCMP (Dubois' demosaicing method used by $COST_{PA}$) (w set to 3 and δ set to 0.5).

Matching cost Images	SSD_{PA}	SSD_g	SAD_{PA}	SAD_g	NCC_{PA}	NCC_g
Plante	**45.88**	45.53	50.17	**50.31**	**43.27**	41.88
Journaux	**87.58**	86.45	**86.88**	85.89	**90.19**	88.52
Cones	65.37	**72.26**	63.91	**70.71**	68.35	**76.11**
Poster	55.4	**70.83**	53.69	**69.19**	55.78	**74.35**
Tsukuba	65.1	**67.46**	64.38	**65.81**	63.75	**67.15**
Venus	65.22	**75.73**	64.39	**74.66**	64.34	**76.28**

Table 4.16 : $RMSE^3$ (Hamilton's demosaicing method used by $COST_{PA}$).

Matching cost Images	SSD_{PA}	SSD_g	SAD_{PA}	SAD_g	NCC_{PA}	NCC_g
Plante	**10.4**	11.4	**9.47**	10.1	**12.4**	15.3
Journaux	**8.46**	9.01	**8.58**	9.03	**8.44**	10.2
Cones	8.39	**8.3**	8.32	**8.21**	**8.88**	9
Poster	**3.25**	3.38	**3.26**	3.5	**3.5**	3.5
Tsukuba	1.95	**1.93**	1.9	**1.87**	**2.14**	2.16
Venus	3.91	**3.68**	3.88	**3.74**	**4.26**	4.32

Table 4.17 : $RMSE^3$ (Dubois' demosaicing method used by $COST_{PA}$).

Matching cost Images	SSD_{PA}	SSD_g	SAD_{PA}	SAD_g	NCC_{PA}	NCC_g
Plante	**10.4**	11.4	**9.41**	10.1	**12.4**	15.3
Journaux	**8**	9.01	**8.15**	9.03	**7.62**	10.2
Cones	**8.24**	8.3	8.23	**8.21**	**8.78**	9
Poster	**3.15**	3.38	**3.16**	3.5	**3.41**	3.5
Tsukuba	1.99	**1.93**	1.94	**1.87**	2.19	**2.16**
Venus	3.92	**3.68**	3.82	**3.74**	**4.32**	4.32

4.8 Conclusion

In this chapter, experiments have been achieved to assess the efficiency of our partial method that is applied to partially demosaiced color images. For this purpose, we have compared its performances with those reached by the standard matching cost applied to fully demosaiced color images. The performances have been measured thanks to the rates of correctly matched pixels and the root mean square estimated disparity error. Experiments show that the performance improvement is robust against the aggregation window size adjustment, the used demosaicing scheme, the disparity error tolerance and the acquisition noise.

Even if the improvement brought by our partial method is often slight, our scheme is less time consuming than the standard one. First, partial demosaicing is less time consuming than the full one since it estimates only one single color component. Moreover, our partial cost takes into account two color components at each pixel whereas that used by the standard cost takes into account three color components.

To explain the reasons why our partial cost outperforms the standard one, we have focused on the demosaicing scheme which estimates the two color components at each pixel. The two methods analyze the second color component (SCC) estimated at each pixel whereas the third color component (TCC) is taken into account by only the standard one. We have experimentally shown that the quality estimation of the third color component is so bad that it decreases the quality of pixel matching in many cases.

Finally, we have examined the quality of matching improvement brought when using single-CCD color cameras instead of gray-level ones. For this purpose, we have compared the quality of pixel matching of CFA stereo image pairs using partial costs and that provided by the gray-level costs computed on the gray-level stereo image pairs.

We conclude from this comparison that matching using gray-level images generally provides higher RCMP than matching using partially demosaiced color images when the aggregation window is low. However, the accuracy of the estimated disparity map using partially demosaiced color images is higher than that calculated using gray-level images.

General conclusion

This work achieved during the preparation of this PhD thesis concerns the development of a partial demosaicing scheme specially designed for stereo matching of CFA images.

We divided our PhD thesis manuscript into four chapters. In the first chapter, we have introduced the fundamentals of computational binocular stereovision. This includes the various properties of a stereovision setup, the stereo correspondence problem and matching methods.

The stereo correspondence problem consists in identifying homologous pixels in the two images. In our work, the cameras included in the retained stereovision setup respect the canonical configuration, so that the homologous pixels lie in the same line in the two images but at different locations. Methods that aim at solving the correspondence problem are called stereo matching methods and are divided into local and global ones.

Local methods implicitly apply a smoothness assumption while the global ones explicitly model it. Local methods compare the local neighborhoods of pixels to identify the homologous pixels whereas global ones take into account the whole image. Even if global methods provide better matching results according to recent taxonomies, they are not well adapted to real-time applications since they are time consuming. On the opposite, local methods based on WTA can easily be implemented in embedded systems using parallel architectures. The study in this chapter is limited to gray-level stereo matching methods.

In the second chapter, we have extended gray-level stereo matching methods to color stereo matching methods since many authors have reported that the use of color can highly improve the accuracy of stereo matching results.

We have studied how color information can be used by each one of the four steps that constitute generally a stereovision scheme : preprocessing, cost computation, cost aggregation, and optimization. However,the quality of color representation depends on the type of color cameras. Color images can be acquired by three-CCD cameras or single-CCD ones. Three-CCD color cameras deliver full color images where pixels are characterized by R, G, and B color components. Since Three-CCD cameras cannot be easily used by a stereovision setup which respects the canonical configuration, one generally uses single-CCD color cameras. Single-CCD cameras provide CFA images where each pixel is characterized by one single color component (R, G or B). A demosaicing step estimates the two missing color components to reconstruct the color images. We have introduced different demosaicing methods that exploit spatial and/or frequency domains to estimate the missing components.

For our study, we have retained Dubois's method since it provides the best performance with respect to different quality measurements and Hamilton's method since it reaches the best compromise between quality and computation time.

In order to match homologous pixels in stereo image pair acquired by single-CCD color cameras, the standard method consists in first reconstructing the demosaiced color images, and then appling a color stereo matching method to the demosaiced color images.

Since demosaicing methods intend to produce “perceptually satisfying” demosaiced color images, they attempt to reduce the presence of color artifacts by filtering the images. So, some useful color textures information may be altered in the demosaiced color images. However, to match homologous pixels, window-based stereo matching costs need as much local texture information as possible.

Testing on the 'Murs' stereo image pair, window-based matching costs computed on the gray-level images outperforms matching costs computed on demosaiced color images. So, when the demosaicing scheme and /or the matching cost are not specifically designed for stereovision, using gray-level cameras provides better results than using single-CCD color cameras.

In the third chapter, we have proposed a specific demosaicing scheme adapted to stereo matching applied to CFA stereo image pairs. For this purpose, we have studied the behavior of the full demosaicing scheme which estimates two color components at each every pixel, i.e. the second and the third color components with respect to CFA arrangement. By measuring the fidelity between the demosaiced color images and the original full color images, we found that the third color component estimation fidelity is worse than that of the second color component.

That leads us to propose a partial demosaicing scheme instead of a full one. Partial demosacing scheme only estimates the second color component at each pixel in the CFA images. So, pixels in the partially demosaiced color images are characterized by both the available color component and the second color component. Then, we adapted color matching costs so that they can be computed on partially demosaiced color images.

Hence, to identify the homologous pixels of CFA stereo image pairs, we propose to follow a partial approach instead of the standard one. Experiments have shown using 'Murs' stereo image pair that our partial matching costs computed on the partially demosaiced color images can enhance the matching results obtained by the standard matching costs computed on fully demosaiced color images.

In the last chapter, we have compared the matching results of our partial method and those of standard one based on 26 stereo datasets. The performances have been measured thanks to the rates of correctly matched pixels and the root mean square error of the estimated disparity. In addition to reducing the processing time of matching process, partial method improves the matching results obtained using the standard one. A detailed study of the estimation quality has been achieved in order to explain the reasons of this improvement.

As a conclusion, it is recommended to use the partial approach instead of the standard one when using single-CCD cameras for stereo matching. However, an important question rises here : should we choose single-CCD cameras for stereo matching or gray-level ones ?

To answer this question, we have examined the matching quality improvement when using single-CCD cameras instead of gray-level ones. The matching results obtained with gray-level images are better than those obtained with partially demosaiced color images when compared with respect to RCMP.

This leads us to conclude that for applications which make use of cameras only for the purpose stereo matching, it seems that the gray-level cameras are more adapted than the single-CCD cameras. However, if color is needed for other purpose and for stereo matching, then the partial method applied on the CFA stereo images has proven to give better matching results than the standard method.

In future research, there are some topics that we are planning to address :

- We will try to study in more details the difference in matching results when using single-CCD and gray-level cameras.
- Another point to study is the partial demosaicing method. Although Dubois's method is classified as better than Hamilton's method in the taxonomies about demosaicing, the stereo matching results using Hamilton's scheme was better than that using Dubois's one. For this reason, we will try to find, in future work, the best partial demosacing method which is adapted to stereo matching of CFA stereo images.
- It would be also interesting to test a stereo matching method of CFA stereo images which is between the partial and standard one (we will call it adapted method). In this method, for certain pixels we use the partial method to identify their homologous and for others we use the standard one. In other words, the third color component will be used for the matching process of certain pixels and ignored for others. For example, the generalization to this adapted approach of the sum of squared differences cost computed between the left pixel with coordinates (x_l,y) and a candidate pixel in the right demosaiced color image, with s-shifted coordinates $(x_l - s,y)$, is expressed as :

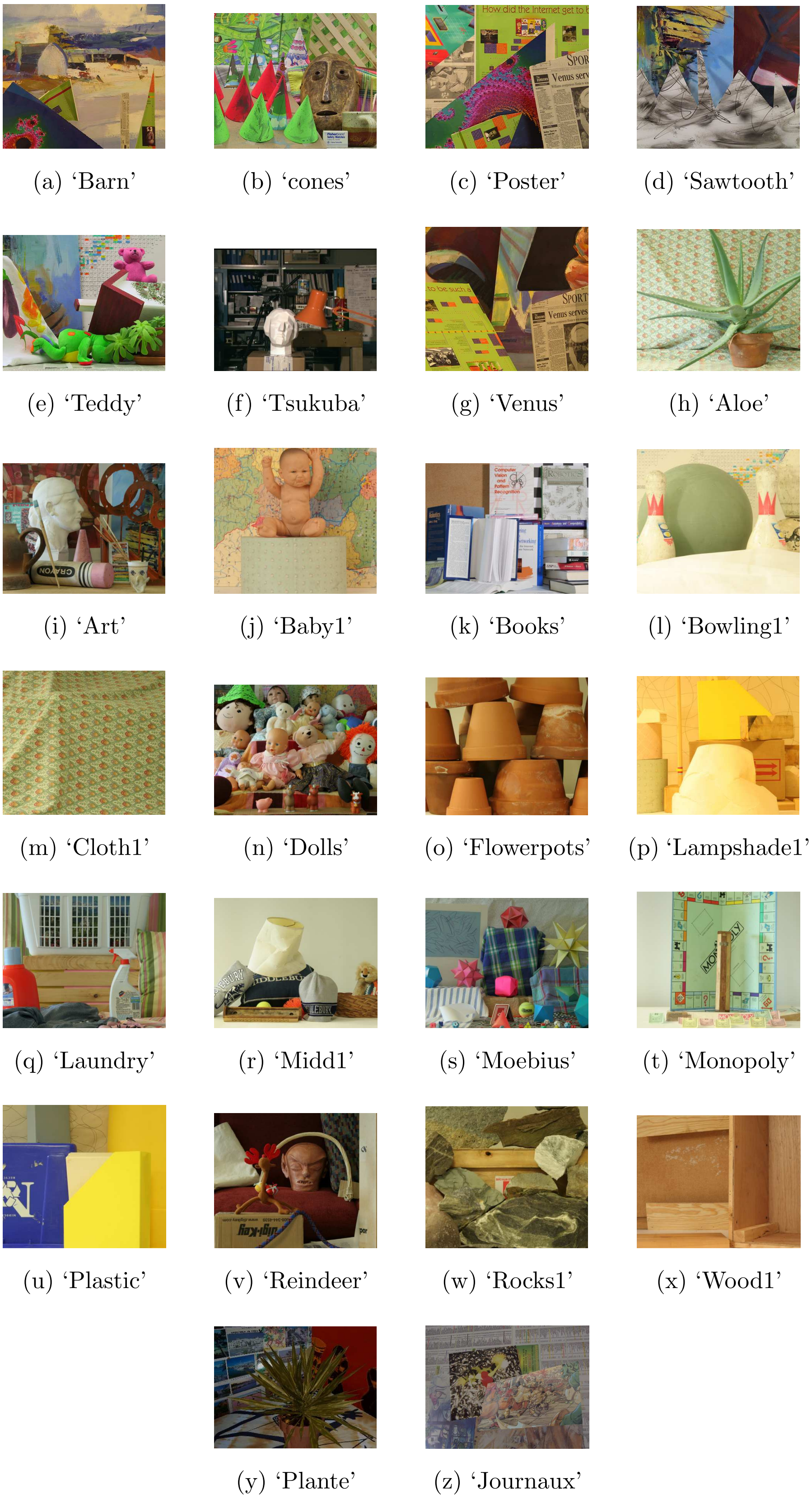
$$\begin{aligned} SSD^w_{AD}(x_l,y,s) \quad = \quad & \sum_{i=-w}^{w} \sum_{j=-w}^{w} \left(\left\| \hat{\mathbf{I}}_{lPA}(x_l+i,y+j) - \hat{\mathbf{I}}_{rPA}(x_l+i-s,y+j) \right\|^2 \right. \qquad (4.5) \\ & \left. + \beta(x_l,y) \cdot \| TCC(x_l+i,y+j) - TCC(x_l+i-s,y+j) \|^2 \right) , \end{aligned}$$

where $\beta(x,y)$ is equal to one for pixels where TCC is used and zero otherwise. The problem of attributing a β values for each pixel should be studied. The idea of this approach is to use the third color component when necessary to enhance the results of the partial method.

Annexe A

Tables of experimental results over all datasets

The left image of each stereo image pair used in our experiments is shown in figure A.1 and its corresponding left disparity map are shown in figure A.1.



(a) 'Barn' (b) 'cones' (c) 'Poster' (d) 'Sawtooth'

(e) 'Teddy' (f) 'Tsukuba' (g) 'Venus' (h) 'Aloe'

(i) 'Art' (j) 'Baby1' (k) 'Books' (l) 'Bowling1'

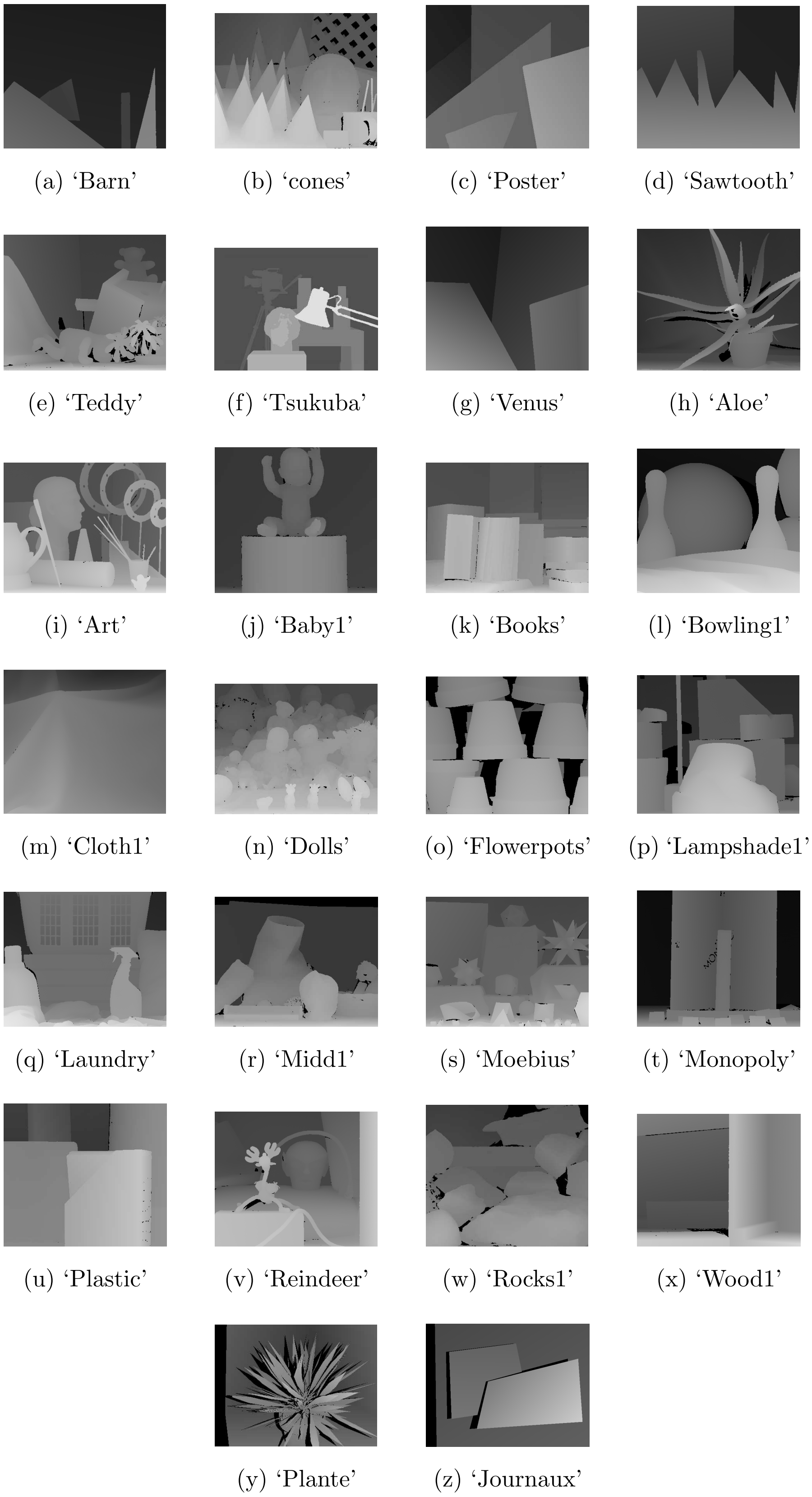
(m) 'Cloth1' (n) 'Dolls' (o) 'Flowerpots' (p) 'Lampshade1'

(q) 'Laundry' (r) 'Midd1' (s) 'Moebius' (t) 'Monopoly'

(u) 'Plastic' (v) 'Reindeer' (w) 'Rocks1' (x) 'Wood1'

(y) 'Plante' (z) 'Journaux'

Figure A.1 : Left image of each tested stereo image pair.



(a) 'Barn' (b) 'cones' (c) 'Poster' (d) 'Sawtooth'

(e) 'Teddy' (f) 'Tsukuba' (g) 'Venus' (h) 'Aloe'

(i) 'Art' (j) 'Baby1' (k) 'Books' (l) 'Bowling1'

(m) 'Cloth1' (n) 'Dolls' (o) 'Flowerpots' (p) 'Lampshade1'

(q) 'Laundry' (r) 'Midd1' (s) 'Moebius' (t) 'Monopoly'

(u) 'Plastic' (v) 'Reindeer' (w) 'Rocks1' (x) 'Wood1'

(y) 'Plante' (z) 'Journaux'

Figure A.2 : Disparity maps of tested stereo image pairs shown in figure A.1.

Table A.1 : RCMP by partial method only ($\mathcal{PO}^{0.5,3}\%$) and by standard method only ($\mathcal{SO}^{0.5,3}\%$) (Hamilton's demosaicing method).

Matching cost	SSD		SAD		NCC	
Dataset	$\mathcal{PO}^{0.5,3}\%$	$\mathcal{SO}^{0.5,3}\%$	$\mathcal{PO}^{0.5,3}\%$	$\mathcal{SO}^{0.5,3}\%$	$\mathcal{PO}^{0.5,3}\%$	$\mathcal{SO}^{0.5,3}\%$
Plante	**3.23**	2.3	**3.54**	2.66	**3.22**	2.28
Journaux	**2.53**	0.7	**2.11**	0.76	**2.74**	0.71
Barn	1.73	**2.64**	2.01	**3.14**	1.62	**2.03**
Cones	1.24	**1.29**	1.31	**1.77**	**1.25**	0.98
Poster	**2.33**	1.71	**2.3**	1.91	**2.39**	1.89
Sawtooth	1.52	**1.6**	**1.78**	1.51	1.6	**1.65**
Teddy	**2.35**	1.72	**2.7**	1.93	2	**2.01**
Tsukuba	**2.07**	0.93	**2.11**	1.03	**2.36**	1.01
Venus	**2.32**	1.91	**2.53**	2.15	**2.2**	2.01
Aloe	**1.34**	0.88	**1.53**	0.9	**1.43**	0.9
Art	**1.5**	1.2	**1.52**	1.49	**1.66**	1.19
Baby1	**2.19**	1.32	**2.12**	1.55	**2.71**	1.31
Books	**2.41**	1.41	**2.26**	1.62	**2.42**	1.47
Bowling2	**1.51**	1.19	**1.35**	1.13	**3.36**	1.41
Cloth1	**0.91**	0.72	**1.11**	0.7	**0.96**	0.67
Dolls	**1.27**	1.11	**1.45**	1.28	**1.66**	1.09
Flowerpots	**1.91**	1.49	**1.85**	1.7	**3.77**	1.76
Lampshade1	**2.53**	1.84	**2.47**	2.24	**3.4**	1.46
Laundry	**2.32**	1.06	**2.12**	1.19	**3.01**	1.21
Midd1	**1.19**	1.08	1.11	**1.44**	**1.78**	0.84
Moebius	**1.43**	1	**1.5**	1.26	**1.78**	0.99
Monopoly	**1.37**	1.17	**1.53**	1.51	**2.31**	1.25
Plastic	**1.38**	1.12	**1.4**	1.06	**2.84**	1.71
Reindeer	**2.84**	0.96	**2.61**	1.23	**2.99**	1.01
Rocks1	**1.11**	0.82	**1.22**	1.02	**1.6**	0.95
Wood1	**2.17**	2.01	2.15	**2.66**	**2.45**	1.25

Table A.2 : RCMP by partial method only ($\mathcal{PO}^{0.5,3}\%$) and by standard method only ($\mathcal{SO}^{0.5,3}\%$) (Dubois' demosaicing method).

Matching cost	SSD		SAD		NCC	
Dataset	$\mathcal{PO}^{0.5,3}\%$	$\mathcal{SO}^{0.5,3}\%$	$\mathcal{PO}^{0.5,3}\%$	$\mathcal{SO}^{0.5,3}\%$	$\mathcal{PO}^{0.5,3}\%$	$\mathcal{SO}^{0.5,3}\%$
Plante	**2.04**	1.8	**2.74**	2.07	**1.91**	1.83
Journaux	**1.04**	0.63	**1**	0.79	**1.15**	0.69
Barn	1.07	**1.97**	1.25	**2.34**	0.98	**1.52**
Cones	0.95	**1**	1.02	**1.73**	**0.95**	0.62
Poster	**1.4**	0.96	**1.43**	1.2	**1.26**	0.99
Sawtooth	**1.13**	0.83	**1.36**	0.97	**1.16**	0.88
Teddy	**1.73**	1.31	**1.99**	1.63	**1.54**	1.29
Tsukuba	**1.23**	0.66	**1.2**	0.82	**1.36**	0.73
Venus	**1.31**	1.2	1.45	**1.5**	**1.23**	1.2
Aloe	**1.09**	0.62	**1.17**	0.71	**1.23**	0.64
Art	**1.27**	1.13	1.27	**1.31**	**1.68**	0.97
Baby1	**3.37**	1.25	**3.72**	1.33	**4.65**	0.95
Books	**2.02**	1.24	**2.04**	1.44	**2.33**	1.18
Bowling2	**1.62**	1.44	1.34	**1.5**	**5.11**	1.44
Cloth1	**0.64**	0.3	**0.7**	0.38	**0.6**	0.26
Dolls	**1.08**	0.89	**1.26**	1.11	**1.75**	0.88
Flowerpots	**2.42**	1.31	**2.35**	1.65	**4.14**	1.27
Lampshade1	**2.29**	1.15	**1.92**	1.64	**3.24**	0.75
Laundry	**1.82**	0.87	**1.69**	1	**2.61**	1.02
Midd1	**1.04**	0.85	1.12	**1.35**	**1.99**	0.61
Moebius	**1.1**	0.9	**1.22**	1.1	**1.45**	0.82
Monopoly	**1.33**	1.06	**1.54**	1.54	**2.79**	1.04
Plastic	**1.25**	0.96	**1.3**	1.08	**2.04**	1.1
Reindeer	**1.57**	0.94	**1.66**	1.06	**2.13**	0.98
Rocks1	**0.67**	0.55	**0.76**	0.71	**1.12**	0.52
Wood1	**1.82**	1.58	1.86	**2.23**	**2.57**	0.86

Table A.3 : Mean improvement $\overline{Imp}^{\delta}$ (Hamilton's demosaicing method).

	$\delta = 0.5$			$\delta = 1$		
Matching cost Dataset	SSD	SAD	NCC	SSD	SAD	NCC
Plante	**0.24**	**0.24**	**0.24**	**0.59**	**0.58**	**0.53**
Journaux	**0.66**	**0.48**	**0.75**	**1.31**	**1**	**1.44**
Barn	-0.57	-0.84	-0.17	-1.02	-1.36	**-0.36**
Cones	-0.11	-0.33	**0.04**	-0.14	-0.36	**0.01**
Poster	**0.34**	**0.37**	**0.34**	**0.46**	**0.35**	**0.39**
Sawtooth	**0.08**	**0.15**	**0.09**	**0.16**	**0.24**	**0.17**
Teddy	**0.37**	**0.33**	**0.17**	**0.36**	**0.31**	**0.12**
Tsukuba	**0.77**	**0.67**	**0.82**	**0.77**	**0.68**	**0.83**
Venus	**0.3**	**0.19**	**0.16**	**0.47**	**0.31**	**0.3**
Aloe	**0.16**	**0.24**	**0.15**	**0.61**	**0.77**	**0.59**
Art	**0.06**	**0.01**	**0.07**	**0.19**	**0.02**	**0.29**
Baby1	**0.15**	**0.11**	**0.16**	**0.63**	**0.53**	**0.77**
Books	**0.22**	**0.18**	**0.21**	**0.61**	**0.53**	**0.67**
Bowling2	**0.02**	-0.01	**0.74**	**0.13**	**0.05**	**1.87**
Cloth1	**0.02**	**0.05**	**0**	**0.24**	**0.4**	**0.17**
Dolls	**0.05**	**0.05**	**0.11**	**0.15**	**0.2**	**0.32**
Flowerpots	**0.2**	**0.06**	**0.85**	**0.54**	**0.27**	**1.9**
Lampshade1	**0.3**	**0.19**	**0.7**	**0.51**	**0.22**	**1.53**
Laundry	**0.31**	**0.2**	**0.35**	**1.44**	**1.08**	**1.81**
Midd1	-0.02	-0.17	**0.22**	**0.05**	-0.27	**0.63**
Moebius	**0.09**	**0.09**	**0.12**	**0.32**	**0.26**	**0.53**
Monopoly	**0.13**	**0.13**	**0.42**	**0.2**	**0.18**	**0.99**
Plastic	**0.11**	**0.17**	**0.55**	**0.2**	**0.33**	**1.25**
Reindeer	**0.41**	**0.31**	**0.47**	**1.1**	**0.84**	**1.21**
Rocks1	**0.1**	**0.06**	**0.27**	**0.19**	**0.1**	**0.54**
Wood1	**0**	-0.15	**0.25**	**0.21**	-0.13	**0.85**
average	**0.17**	0.11	**0.31**	**0.40**	0.27	**0.74**

Table A.4 : Mean improvement $\overline{Imp}^{\delta}$ (Dubois' demosaicing method).

	$\delta = 0.5$			$\delta = 1$		
Matching cost Dataset	SSD	SAD	NCC	SSD	SAD	NCC
Plante	**0.22**	**0.23**	**0.21**	**0.55**	**0.55**	**0.48**
Journaux	**0.67**	**0.49**	**0.76**	**1.34**	**1.02**	**1.47**
Barn	-0.56	-0.84	-0.18	-1.02	-1.36	**-0.37**
Cones	-0.12	-0.33	**0.04**	-0.15	-0.35	**0.01**
Poster	**0.34**	**0.36**	**0.34**	**0.45**	**0.34**	**0.38**
Sawtooth	**0.06**	**0.14**	**0.08**	**0.16**	**0.23**	**0.16**
Teddy	**0.34**	**0.32**	**0.17**	**0.32**	**0.29**	**0.11**
Tsukuba	**0.77**	**0.66**	**0.83**	**0.77**	**0.68**	**0.84**
Venus	**0.28**	**0.18**	**0.15**	**0.46**	**0.29**	**0.3**
Aloe	**0.15**	**0.24**	**0.14**	**0.61**	**0.77**	**0.6**
Art	**0.05**	**0.01**	**0.08**	**0.17**	**0**	**0.27**
Baby1	**0.14**	**0.11**	**0.16**	**0.62**	**0.52**	**0.76**
Books	**0.22**	**0.17**	**0.2**	**0.61**	**0.54**	**0.69**
Bowling2	**0.04**	-0.02	**0.72**	**0.17**	**0.04**	**1.88**
Cloth1	**0**	**0.05**	**0**	**0.2**	**0.39**	**0.17**
Dolls	**0.04**	**0.04**	**0.09**	**0.15**	**0.19**	**0.31**
Flowerpots	**0.2**	**0.07**	**0.84**	**0.59**	**0.3**	**1.94**
Lampshade1	**0.32**	**0.19**	**0.71**	**0.53**	**0.25**	**1.53**
Laundry	**0.29**	**0.2**	**0.35**	**1.43**	**1.09**	**1.81**
Midd1	-0.03	-0.18	**0.2**	**0.04**	-0.27	**0.63**
Moebius	**0.1**	**0.08**	**0.13**	**0.32**	**0.26**	**0.53**
Monopoly	**0.13**	**0.13**	**0.42**	**0.2**	**0.18**	**0.98**
Plastic	**0.1**	**0.16**	**0.54**	**0.2**	**0.32**	**1.27**
Reindeer	**0.41**	**0.31**	**0.48**	**1.08**	**0.88**	**1.22**
Rocks1	**0.09**	**0.05**	**0.27**	**0.2**	**0.1**	**0.57**
Wood1	-0.01	-0.16	**0.25**	**0.14**	-0.2	**0.75**
average	**0.16**	0.10	**0.31**	**0.39**	0.27	**0.74**

Table A.5 : $PSNR$ of SCC, TCC estimated by Hamilton's and Dubois' demosaicing methods applied to left images.

Demosaicing / Dataset	Hamilton		Dubois	
	$PSNR^{SCC}$	$PSNR^{TCC}$	$PSNR^{SCC}$	$PSNR^{TCC}$
Plante	**34.76**	30.07	**36.81**	32.87
Journaux	**35.39**	30.36	**38.86**	34.6
Barn	**35.16**	30.07	**33.97**	29.24
Cones	**32.27**	28.24	**31.34**	27.23
Poster	**30.55**	25.79	**29.35**	24.97
Sawtooth	**34.86**	29.96	**34.8**	29.81
Teddy	**33.84**	29.15	**32.4**	28.16
Tsukuba	**37.48**	32.28	**37.23**	32.15
Venus	**34.44**	29.41	**33.49**	29.18
Aloe	**35.29**	29.26	**34.35**	28.66
Art	**39.22**	33.69	**37.3**	31.33
Baby1	**35.98**	29.93	**33.92**	27.81
Books	**36.93**	31.26	**36.14**	30.03
Bowling2	**42.48**	36.67	**39.93**	33.67
Cloth1	**35.33**	29.29	**34.11**	28.36
Dolls	**37.79**	32.14	**36.06**	30.59
Flowerpots	**46.73**	40.31	**43.35**	37.79
Lampshade1	**44.7**	38.33	**41.17**	36.51
Laundry	**34.97**	30.07	**36.88**	32.04
Midd1	**41.87**	35.99	**40.72**	35.05
Moebius	**38.98**	34.07	**37.74**	32.6
Monopoly	**38.97**	33.59	**36.8**	31.76
Plastic	**44.02**	39.19	**41.34**	35.71
Reindeer	**38.85**	33.1	**38.58**	32.7
Rocks1	**40.88**	35.25	**41**	35.33
Wood1	**45.95**	39.88	**43.77**	37.99

Table A.6 : $PSNR$ of R, G, and B estimated by Hamilton's and Dubois' demosaicing methods applied to left images.

Demosaicing	Hamilton			Dubois		
dataset	$PSNR^R$	$PSNR^G$	$PSNR^B$	$PSNR^R$	$PSNR^G$	$PSNR^B$
Plante	31.65	**32.49**	31.98	34.29	**36.48**	34.53
Journaux	32.2	**33.32**	32.17	36.24	**38.86**	36.22
Barn	31.79	**35.41**	32.02	30.87	**34.06**	31.11
Cones	29.74	**33.87**	29.86	28.71	**31.45**	28.93
Poster	26.98	**31.55**	28.22	26.21	**29.36**	27.09
Sawtooth	32.55	**34.17**	31.08	32.42	**34.91**	30.95
Teddy	31.04	**35.02**	30.74	29.87	**32.42**	29.7
Tsukuba	33.3	**38.24**	35.19	33.18	**38.31**	34.98
Venus	31.29	**34.26**	31.17	30.93	**33.02**	30.72
Aloe	30.24	**34.34**	32.71	29.72	**34.28**	31.78
Art	34.67	**40.36**	36.88	32.71	**37.35**	34.13
Baby1	30.98	**37.34**	33.28	29.34	**33.89**	30.48
Books	32.72	**37.32**	33.8	31.73	**36.36**	32.48
Bowling2	38.6	**43.7**	38.74	35.77	**39.4**	35.75
Cloth1	30.06	**34.69**	33.14	29.26	**34.08**	31.8
Dolls	33.67	**37.93**	34.59	32.19	**35.98**	32.87
Flowerpots	41.75	**47.69**	43.25	39.02	**43.37**	40.59
Lampshade1	41.03	**46.31**	39.92	38.27	**39.97**	38.21
Laundry	31.57	**35.8**	32.18	33.36	**37.21**	34.33
Midd1	38.07	**40.7**	37.94	36.97	**40.34**	37.07
Moebius	35.02	**40.49**	36.92	33.81	**37.83**	35.19
Monopoly	34.81	**39.9**	36.3	33	**36.57**	34.28
Plastic	45.18	**46.83**	38.87	38.83	**40.72**	36.76
Reindeer	34.38	**39.12**	35.94	33.96	**38.86**	35.64
Rocks1	37.45	**39.42**	36.98	37.39	**40.78**	37.2
Wood1	42.05	**45.08**	41.81	39.55	**43.68**	40.45

Table A.7 : PSNR of TCC estimated by Hamilton's demosaicing method at left pixels belonging to different subsets.

Matching cost	SSD		SAD		NCC	
Subset	$\mathcal{PO}^{0.5,3}$	$\mathcal{SO}^{0.5,3}$	$\mathcal{PO}^{0.5,3}$	$\mathcal{SO}^{0.5,3}$	$\mathcal{PO}^{0.5,3}$	$\mathcal{SO}^{0.5,3}$
Dataset						
Plante	29.5	**31.9**	30.3	**32.7**	29.1	**31.5**
Journaux	25.3	**31.6**	25.8	**32.6**	25.4	**31.5**
Barn	33.8	**34.2**	33.7	**34.2**	33.2	**33.8**
Cones	28	**30**	28	**30.3**	27.4	**30.1**
Poster	25.2	**26.8**	25.5	**27**	25	**27.5**
Sawtooth	31.9	**32.6**	32.2	**32.9**	31.8	**33.2**
Teddy	**31.4**	30.9	30.7	**31.3**	30.3	**31.2**
Tsukuba	**35.1**	33.9	33.6	**35.8**	34.4	**34.9**
Venus	29.8	**32.3**	30.4	**32.3**	29.8	**32.1**
Aloe	29.3	**29.9**	29	**30.3**	29.6	**29.9**
Art	32.9	**34.6**	33.4	**34.6**	33	**35.2**
Baby1	29.5	**32.6**	30.2	**32.4**	30.6	**31.4**
Books	28.1	**33.4**	29.4	**33.7**	28.6	**32.9**
Bowling2	35.8	**40.3**	37.2	**39**	39	**40.3**
Cloth1	28.8	**29.2**	29.1	**29.5**	29.1	**29.1**
Dolls	32.3	**33**	31.7	**32.7**	31.8	**33.1**
Flowerpots	**40.9**	39.9	**40.9**	40.1	**42.9**	41.4
Lampshade1	37.1	**40**	37.3	**38.6**	37.5	**39.6**
Laundry	26.5	**30.8**	26.8	**32.2**	28.8	**30.8**
Midd1	36.8	**36.9**	36.4	**39.1**	**38.4**	36.7
Moebius	33.1	**34.8**	33.4	**34.3**	**34.3**	33.9
Monopoly	**34.1**	33.4	**33.7**	32.9	**35.9**	35
Plastic	**40.6**	38	**40.6**	39.7	**41.9**	38.7
Reindeer	28.5	**33.7**	28.9	**34.1**	29.5	**32.1**
Rocks1	30.5	**35**	30.9	**35.2**	31.9	**35.3**
Wood1	39.8	**40.6**	39.6	**40.7**	40.4	**40.5**

Table A.8 : PSNR of TCC estimated by Dubois' demosaicing method at left pixels belonging to different subsets.

Matching cost	SSD		SAD		NCC	
Subset	$\mathcal{PO}^{0.5,3}$	$\mathcal{SO}^{0.5,3}$	$\mathcal{PO}^{0.5,3}$	$\mathcal{SO}^{0.5,3}$	$\mathcal{PO}^{0.5,3}$	$\mathcal{SO}^{0.5,3}$
Dataset						
Plante	34	**35.1**	33.8	**34.5**	33.7	**34.5**
Journaux	30.2	**36.5**	31.3	**36.9**	30.5	**35.6**
Barn	32.3	**33.8**	32.4	**32.9**	31.6	**33.9**
Cones	27.3	**28.6**	27.4	**28.8**	27.4	**27.6**
Poster	24.6	**27**	24.7	**26.2**	24.5	**27.4**
Sawtooth	33.5	**33.7**	33.9	**34**	32.9	**33.9**
Teddy	31	**31.2**	**31.6**	30.3	30.3	**31.6**
Tsukuba	**34.4**	33.2	33.8	**34.4**	34.4	**35.6**
Venus	30.7	**33**	32.3	**32.5**	31	**33.4**
Aloe	29.4	**30.4**	28.9	**30.1**	29.1	**30.4**
Art	32.8	**33.2**	31.6	**32.7**	**33.8**	33.3
Baby1	34.1	**35.4**	**34.7**	33.8	**35.3**	35.2
Books	31.9	**32.5**	31	**31.3**	**32.1**	32
Bowling2	35.2	**38.3**	36.4	**37.9**	**38.8**	37.9
Cloth1	28	**29.1**	28.4	**29.1**	27.7	**29.4**
Dolls	30.7	**33**	31.7	**32**	**34.4**	33.3
Flowerpots	**39.2**	38.7	**39.2**	38.2	**40.8**	39.9
Lampshade1	38.4	**40**	38.8	**39.4**	39.1	**39.7**
Laundry	30.6	**32.5**	30.9	**32.7**	32.4	**32.8**
Midd1	35.9	**36.6**	**36.8**	35.7	**38.4**	36.4
Moebius	32.9	**34.3**	32.8	**34.3**	**34.1**	34
Monopoly	30.9	**32.4**	32.4	**32.6**	34.3	**34.3**
Plastic	**38.6**	38.3	**38.9**	36.9	**38.4**	38.3
Reindeer	33.1	**34.8**	35.1	**35.1**	33.8	**36.9**
Rocks1	29.4	**37.2**	31.7	**35.5**	31.4	**36.6**
Wood1	38.9	**39.5**	39.2	**39.3**	**39.5**	39.2

Table A.9 : Rate of pixels belonging to different subsets that are correctly matched by $COST_{TCC}$ (TCC estimated by Hamilton's demosaicing method) (δ set to 0.5 and w set to 3).

Matching cost	SSD		SAD		NCC	
Subset	$\mathcal{PO}^{0.5,3}$	$\mathcal{SO}^{0.5,3}$	$\mathcal{PO}^{0.5,3}$	$\mathcal{SO}^{0.5,3}$	$\mathcal{PO}^{0.5,3}$	$\mathcal{SO}^{0.5,3}$
Dataset						
Plante	0	**77.4**	0	**77.6**	0.04	**74.3**
Journaux	0.08	**83.9**	0.23	**83.7**	0.25	**84.7**
Barn	0.16	**76.1**	0.1	**74.4**	3.87	**80.2**
Cones	0.11	**90.2**	0.26	**90.4**	1.68	**85.3**
Poster	0.09	**83.8**	0.06	**82.3**	0.62	**84.8**
Sawtooth	0.09	**87**	0.08	**84.6**	2.36	**84.2**
Teddy	0.14	**81.1**	0.25	**80.3**	5.08	**77.6**
Tsukuba	0	**81.2**	0	**82.1**	0.22	**81.7**
Vensus	0	**82.4**	0.24	**80.4**	0.37	**80.9**
Aloe	0	**91.7**	0	**92.9**	1.07	**88.2**
Art	0	**90.1**	0	**90.6**	0.24	**86.6**
Baby1	1.71	**90.7**	0	**91.4**	0.35	**86.2**
Books	0	**87.6**	1.47	**85.2**	0.22	**85**
Bowling2	1.17	**87**	1.92	**86**	0.73	**78.1**
Cloth1	1.92	**91.4**	0.59	**95.8**	0.45	**94.5**
Dolls	0	**91.5**	1.44	**91.1**	1.01	**88.2**
Flowerpots	0.21	**89.3**	0.22	**90**	0.81	**77.1**
Lampshade1	0.94	**70**	0.14	**75.2**	1.23	**64.7**
Laundry	0	**78.1**	0.77	**79.8**	0.2	**79.7**
Midd1	0	**90.7**	0	**89.1**	0.68	**82.2**
Moebius	0	**88.2**	0.43	**90.3**	0.4	**86.9**
Monopoly	1.59	**89**	2.06	**83.5**	3.68	**74.6**
Plastic	0.52	**73.7**	0.92	**71.6**	1	**70.8**
Reindeer	0.36	**81.6**	0.44	**79.8**	0.02	**78**
Rocks1	0.2	**94.3**	1.26	**88.4**	0.97	**85.1**
Wood1	0	**88**	0	**86.9**	0.73	**84.6**

Table A.10 : Rate of pixels belonging to different subsets that are correctly matched by $COST_{TCC}$ (TCC estimated by Dubois' demosaicing method) (δ set to 0.5 and w set to 3).

Matching cost	SSD		SAD		NCC	
Subset	$\mathcal{PO}^{0.5,3}$	$\mathcal{SO}^{0.5,3}$	$\mathcal{PO}^{0.5,3}$	$\mathcal{SO}^{0.5,3}$	$\mathcal{PO}^{0.5,3}$	$\mathcal{SO}^{0.5,3}$
Dataset						
Plante	0	**78.5**	0	**80.2**	0.12	**77.5**
Journaux	0.33	**85.9**	0.57	**86**	0.96	**86.7**
Barn	0.38	**81.9**	0.6	**80.8**	4.88	**86.2**
Cones	0.07	**91.8**	0.47	**92.5**	1.07	**86.2**
Poster	0.1	**86.8**	0	**83.3**	1.58	**87.3**
Sawtooth	0.24	**92.5**	0.2	**90.1**	0.29	**91.2**
Teddy	0.08	**85.9**	0.24	**83.8**	3.07	**83.3**
Tsukuba	0	**91.3**	0	**90.1**	0	**88.4**
Vensus	0.05	**89.4**	0.09	**87.3**	0.65	**86.3**
Aloe	0	**91.5**	0	**92.1**	1.37	**85.9**
Art	0	**91.2**	0	**89.3**	0.27	**86.5**
Baby1	0	**92.4**	0	**89.9**	0.09	**90.4**
Books	0	**91.9**	0	**91.1**	0.08	**89.2**
Bowling2	0	**89**	0	**90.9**	0.26	**87.3**
Cloth1	0	**97.3**	0	**97.9**	0.12	**97.5**
Dolls	0	**92.5**	0	**93**	0.33	**88.3**
Flowerpots	0	**87.5**	0	**89.2**	0.43	**79.1**
Lampshade1	0	**75.3**	0	**80**	0.85	**68.9**
Laundry	0	**82.4**	0	**83.8**	0.23	**83.7**
Midd1	0	**93.5**	0	**93.8**	0.14	**89.7**
Moebius	0	**90.1**	0	**91.1**	0.09	**87.6**
Monopoly	0	**86.9**	0	**86.1**	1.97	**75.1**
Plastic	0	**78.2**	0	**74.6**	1.08	**74.7**
Reindeer	0	**88.2**	0	**86.1**	0.19	**89.9**
Rocks1	0	**97.8**	0	**97.2**	0.13	**92.2**
Wood1	0	**91.4**	0	**91.4**	0.57	**83.6**

Table A.11 : $RMSE^3$ (Hamilton's demosaicing method).

Matching cost Images	SSD_{PA}	SSD_{ST}	SAD_{PA}	SAD_{ST}	NCC_{PA}	NCC_{ST}
Plante	**10.4**	10.6	**9.47**	9.64	**12.4**	12.5
Journaux	**8.46**	9.18	**8.58**	9.08	**8.44**	9.49
Barn	3.13	**2.96**	3.24	**2.98**	3.51	**3.4**
Cones	**8.39**	8.4	**8.32**	8.34	8.88	**8.84**
Poster	**3.25**	3.31	**3.26**	3.32	**3.5**	3.5
Sawtooth	2.3	**2.26**	**2.14**	2.14	2.65	**2.62**
Teddy	**7.99**	8.02	**8.02**	8.03	**8.28**	8.32
Tsukuba	**1.95**	2.02	**1.9**	1.95	**2.14**	2.21
Venus	**3.91**	3.93	**3.88**	3.88	4.26	**4.2**
Aloe	11.2	**11.1**	**10.7**	10.7	12.1	**12**
Art	**15.9**	15.9	**15.6**	15.6	**16.2**	16.3
Baby1	**9.29**	9.32	9.76	**9.64**	**6.97**	7.02
Books	**11.1**	11.1	**11.4**	11.7	**11.7**	11.9
Bowling2	18	**17.9**	18.2	**18.1**	**16.2**	16.3
Cloth1	9.12	**8.83**	**8.8**	8.82	9.63	**9.44**
Dolls	11.1	**11**	11.4	**10.7**	11.6	**11.5**
Flowerpots	**9.89**	9.94	**9.93**	9.96	**9.91**	10
Lampshade1	13.2	**12.6**	14.1	**13.4**	**11.9**	12
Laundry	**15**	15.4	16.2	**15.4**	**16.3**	16.3
Midd1	**15.6**	15.7	**15.7**	15.7	**15.9**	15.9
Moebius	11.5	**11.3**	12.3	**12.1**	9.81	**9.73**
Monopoly	**23.4**	24.7	**23.5**	24	**15.6**	15.9
Plastic	23	**22.9**	**23.3**	23.3	**20.1**	20.2
Reindeer	**14**	14.2	**14**	14.1	**14.3**	14.6
Rocks1	**10.1**	10.2	10.2	**10.1**	**10.2**	10.4
Wood1	**13.6**	13.6	**14.3**	14.3	**11.8**	11.9

Table A.12 : $RMSE^3$ (Dubois' demosaicing method).

Matching cost Dataset	SSD_{PA}	SSD_{ST}	SAD_{PA}	SAD_{ST}	NCC_{PA}	NCC_{ST}
Plante	**10.4**	10.4	**9.41**	9.52	12.4	**12.3**
Journaux	**8**	8.08	8.15	**8.14**	**7.62**	7.69
Barn	3.14	**3.03**	3.13	**3.01**	3.45	**3.39**
Cones	8.24	**8.23**	8.23	**8.19**	8.78	**8.72**
Poster	**3.15**	3.2	**3.16**	3.22	**3.41**	3.41
Sawtooth	**2.31**	2.32	**2.16**	2.2	2.72	**2.68**
Teddy	7.9	**7.89**	7.92	**7.85**	**8.05**	8.06
Tsukuba	**1.99**	2.02	**1.94**	1.97	**2.19**	2.22
Venus	**3.92**	3.92	**3.82**	3.82	4.32	**4.25**
Aloe	11.2	**11.1**	**10.7**	10.7	12.2	**12.1**
Art	15.7	**15.6**	**15.4**	15.4	**16**	16
Baby1	9.25	**9.15**	9.63	**9.45**	**6.9**	6.99
Books	**11.1**	11.1	**11.5**	11.5	**11.3**	11.3
Bowling2	**17.6**	17.7	**18**	18	**16.1**	16.3
Cloth1	**8.36**	8.38	**8.36**	8.38	**8.88**	8.9
Dolls	11.4	**11.3**	11.2	**11.1**	11.3	**11.2**
Flowerpots	**9.32**	9.38	**9.3**	9.31	**9.18**	9.18
Lampshade1	13	**12.4**	13.8	**13.1**	**11.7**	11.9
Laundry	15	**14.9**	15.6	**15**	15.9	**15.7**
Midd1	15.6	**15.5**	15.7	**15.6**	**15.7**	15.7
Moebius	11.4	**11.2**	12.2	**12**	9.44	**9.33**
Monopoly	**23.8**	25.4	**24.8**	26.3	**14.6**	14.8
Plastic	23.1	**22.9**	23.4	**23.3**	20.4	**20.3**
Reindeer	**13.5**	13.5	**13.5**	13.5	**13.7**	13.8
Rocks1	9.81	**9.76**	9.83	**9.75**	**10.1**	10.1
Wood1	**13.5**	13.5	**14.2**	14.2	**11.6**	11.7

Table A.13 : RCMP (Hamilton's demosaicing method used by $COST_{PA}$, w is set to 3 and δ is set to 0.5).

Matching cost Images	SSD_{PA}	SSD_g	SAD_{PA}	SAD_g	NCC_{PA}	NCC_g
Plante	45.51	**45.53**	49.17	**50.31**	**42.78**	41.88
Journaux	**87**	86.45	**86.4**	85.89	**89.25**	88.52
Barn	75.35	**79.41**	75.08	**78.65**	75.37	**79.8**
Cones	67.19	**72.26**	65.99	**70.71**	70.92	**76.11**
Poster	61.89	**70.83**	61.53	**69.19**	62.66	**74.35**
Sawtooth	75.17	**80.72**	76.42	**80.66**	74.14	**80.12**
Teddy	59.36	**65.35**	59.65	**64.5**	61.27	**70.51**
Tsukuba	**69.77**	67.46	**69.82**	65.81	**69.75**	67.15
Venus	67.68	**75.73**	67.54	**74.66**	66.96	**76.28**
Aloe	63.02	**64.76**	64.3	**66.01**	64.59	**66.47**
Art	35.31	**35.96**	35.1	**35.5**	43.98	**46.04**
Baby1	58.15	**63.45**	55.36	**58.91**	67.53	**75.11**
Books	46.68	**47.08**	**44.18**	44.16	51.66	**51.85**
Bowling2	24.53	**26.53**	21.28	**21.64**	43.84	**55.51**
Cloth1	79.67	**82.86**	79.86	**82.77**	79.96	**83.08**
Dolls	49.85	**51.44**	50.44	**51.77**	55.43	**58.12**
Flowerpots	34.77	**34.81**	**32.04**	31.01	51.95	**55.28**
Lampshade1	32.98	**34.88**	29.72	**30.38**	42.2	**52.48**
Laundry	33.22	**34.79**	31.47	**32.35**	40.77	**45.31**
Midd1	29.56	**30.99**	28.38	**29.54**	34.71	**36.93**
Moebius	54.11	**54.48**	53.1	**53.27**	59.07	**61.19**
Monopoly	**31.78**	31.24	**29.16**	27.55	40.43	**47.92**
Plastic	**14.53**	14.11	**13.17**	12.49	21.88	**25.83**
Reindeer	46.55	**50.92**	44.95	**49.98**	53.95	**58.46**
Rocks1	58.46	**58.68**	**58.13**	58.13	62.11	**63.99**
Wood1	54.23	**55.62**	**49.45**	48.98	63.97	**67.29**

Table A.14 : RCMP (Dubois' demosaicing method used by $COST_{PA}$) (w set to 3 and δ set to 0.5).

Matching cost Images	SSD_{PA}	SSD_g	SAD_{PA}	SAD_g	NCC_{PA}	NCC_g
Plante	**45.88**	45.53	50.17	**50.31**	**43.27**	41.88
Journaux	**87.58**	86.45	**86.88**	85.89	**90.19**	88.52
Barn	74.92	**79.41**	74.93	**78.65**	74.72	**79.8**
Cones	65.37	**72.26**	63.91	**70.71**	68.35	**76.11**
Poster	55.4	**70.83**	53.69	**69.19**	55.78	**74.35**
Sawtooth	76.18	**80.72**	77.39	**80.66**	74.94	**80.12**
Teddy	56.69	**65.35**	56.46	**64.5**	58.15	**70.51**
Tsukuba	65.1	**67.46**	64.38	**65.81**	63.75	**67.15**
Venus	65.22	**75.73**	64.39	**74.66**	64.34	**76.28**
Aloe	61.74	**64.76**	62.57	**66.01**	63.04	**66.47**
Art	34.45	**35.96**	33.79	**35.5**	42.56	**46.04**
Baby1	52.85	**63.45**	50.09	**58.91**	60.16	**75.11**
Books	**47.38**	47.08	**44.51**	44.16	**51.86**	51.85
Bowling2	24.36	**26.53**	20.81	**21.64**	42.43	**55.51**
Cloth1	76.18	**82.86**	74.5	**82.77**	75.96	**83.08**
Dolls	48.12	**51.44**	48.43	**51.77**	53.05	**58.12**
Flowerpots	33.95	**34.81**	**31.66**	31.01	47.74	**55.28**
Lampshade1	29.34	**34.88**	26.32	**30.38**	36.32	**52.48**
Laundry	31.53	**34.79**	29.79	**32.35**	37.79	**45.31**
Midd1	28.49	**30.99**	27.21	**29.54**	32.34	**36.93**
Moebius	52.54	**54.48**	51.36	**53.27**	57	**61.19**
Monopoly	30.65	**31.24**	**28.06**	27.55	39.66	**47.92**
Plastic	**14.52**	14.11	**12.88**	12.49	21.3	**25.83**
Reindeer	47.2	**50.92**	46.06	**49.98**	54.5	**58.46**
Rocks1	**58.93**	58.68	**58.57**	58.13	61.91	**63.99**
Wood1	51.45	**55.62**	46.89	**48.98**	61.25	**67.29**

Table A.15 : $RMSE^3$ (Hamilton's demosaicing method used by $COST_{PA}$).

Matching cost Images	SSD_{PA}	SSD_g	SAD_{PA}	SAD_g	NCC_{PA}	NCC_g
Plante	**10.4**	11.4	**9.47**	10.1	**12.4**	15.3
Journaux	**8.46**	9.01	**8.58**	9.03	**8.44**	10.2
Barn	3.13	**2.73**	3.24	**2.81**	3.51	**3.08**
Cones	8.39	**8.3**	8.32	**8.21**	**8.88**	9
Poster	**3.25**	3.38	**3.26**	3.5	**3.5**	3.5
Sawtooth	2.3	**2.08**	2.14	**2.07**	**2.65**	3.02
Teddy	**7.99**	8.3	**8.02**	8.25	**8.28**	8.45
Tsukuba	1.95	**1.93**	1.9	**1.87**	**2.14**	2.16
Venus	3.91	**3.68**	3.88	**3.74**	**4.26**	4.32
Aloe	**11.2**	11.3	10.7	**10.6**	**12.1**	12.8
Art	**15.9**	16.6	**15.6**	16.4	**16.2**	18.1
Baby1	**9.29**	10.6	**9.76**	11.5	**6.97**	8
Books	**11.1**	11.5	**11.4**	12	**11.7**	12
Bowling2	**18**	18.1	**18.2**	18.4	16.2	**14.2**
Cloth1	9.12	**8.64**	8.8	**8.61**	9.63	**9.09**
Dolls	**11.1**	11.9	**11.4**	11.8	**11.6**	13.2
Flowerpots	9.89	**9.56**	9.93	**9.63**	**9.91**	11.5
Lampshade1	**13.2**	16.1	**14.1**	16.9	**11.9**	15.6
Laundry	**15**	16	16.2	**16.1**	**16.3**	16.4
Midd1	**15.6**	16.7	**15.7**	16.8	**15.9**	18.6
Moebius	**11.5**	12.5	**12.3**	13.2	**9.81**	11.3
Monopoly	**23.4**	26.1	**23.5**	26.5	**15.6**	15.6
Plastic	**23**	24.6	**23.3**	24.9	20.1	**19.5**
Reindeer	14	**13.7**	14	**13.6**	**14.3**	14.5
Rocks1	**10.1**	10.1	**10.2**	10.2	10.2	**9.75**
Wood1	**13.6**	13.7	**14.3**	14.6	11.8	**10.6**

Table A.16 : $RMSE^3$ (Dubois' demosaicing method used by $COST_{PA}$).

Matching cost Images	SSD_{PA}	SSD_g	SAD_{PA}	SAD_g	NCC_{PA}	NCC_g
Plante	**10.4**	11.4	**9.41**	10.1	**12.4**	15.3
Journaux	**8**	9.01	**8.15**	9.03	**7.62**	10.2
Barn	3.14	**2.73**	3.13	**2.81**	3.45	**3.08**
Cones	**8.24**	8.3	8.23	**8.21**	**8.78**	9
Poster	**3.15**	3.38	**3.16**	3.5	**3.41**	3.5
Sawtooth	2.31	**2.08**	2.16	**2.07**	**2.72**	3.02
Teddy	**7.9**	8.3	**7.92**	8.25	**8.05**	8.45
Tsukuba	1.99	**1.93**	1.94	**1.87**	2.19	**2.16**
Venus	3.92	**3.68**	3.82	**3.74**	**4.32**	4.32
Aloe	**11.2**	11.3	10.7	**10.6**	**12.2**	12.8
Art	**15.7**	16.6	**15.4**	16.4	**16**	18.1
Baby1	**9.25**	10.6	**9.63**	11.5	**6.9**	8
Books	**11.1**	11.5	**11.5**	12	**11.3**	12
Bowling2	**17.6**	18.1	**18**	18.4	16.1	**14.2**
Cloth1	**8.36**	8.64	**8.36**	8.61	**8.88**	9.09
Dolls	**11.4**	11.9	**11.2**	11.8	**11.3**	13.2
Flowerpots	**9.32**	9.56	**9.3**	9.63	**9.18**	11.5
Lampshade1	**13**	16.1	**13.8**	16.9	**11.7**	15.6
Laundry	**15**	16	**15.6**	16.1	**15.9**	16.4
Midd1	**15.6**	16.7	**15.7**	16.8	**15.7**	18.6
Moebius	**11.4**	12.5	**12.2**	13.2	**9.44**	11.3
Monopoly	**23.8**	26.1	**24.8**	26.5	**14.6**	15.6
Plastic	**23.1**	24.6	**23.4**	24.9	20.4	**19.5**
Reindeer	**13.5**	13.7	**13.5**	13.6	**13.7**	14.5
Rocks1	**9.81**	10.1	**9.83**	10.2	10.1	**9.75**
Wood1	**13.5**	13.7	**14.2**	14.6	11.6	**10.6**

Bibliographie

[ASH05] D. Alleysson, S. Süsstrunk, and J. Hérault. Linear demosaicing inspired by the human visual system. *IEEE Transactions on Image Processing*, 14(4) :439–449, 2005.

[ATV00] A. Fusiello, E. Trucco, and A. Verri. A compact algorithm for rectification of stereo pairs. *Machine Vision and Applications*, 12(1) :16–22, 2000.

[Bay76] B.E. Bayer. Color imaging array. U.S. patent 3,971,065, to Eastman Kodak Co., Patent and Trademark Office, Washington D.C., 1976.

[BBH03] M.Z. Brown, Z. Burschka, and G.D. Hager. Advances in computational stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8) :993–1008, August 2003.

[BCP00] T. Belli, M. Cord, and S. Philipp-Foliguet. Colour contribution for stereo image matching. In *International Conference on Color in Graphics and Image Processing*, pages 317–322, Saint-Etienne, France, October 2000.

[BG07] M. Bleyer and M. Gelautz. Graph-cut-based stereo matching using image segmentation with symmetrical treatment of occlusions. *Signal Processing : Image Communication*, 22(1) :127–143, 2007.

[BGMT08] S. Battiato, M. Guarnera, G. Messina, and V. Tomaselli. Recent patents on color demosaicing. *Recent Patents on Computer Science*, 1(3) :194–207, November 2008.

[BI99] A.F. Bobick and S.S. Intille. Large occlusion stereo. *International Journal of Computer Vision*, 33 :181–200, 1999.

[Ble06] M. Bleyer. *Segmentation-based stereo and motion with occlusions*. Phd thesis, Vienna University of Technology, February 2006.

[BSM08] S. Bayrama, H.T. Sencar, and N. Memon. Classification of digital camera-models based on demosaicing artifacts. *Digital Investigation*, 5(1-2) :49–59, September 2008.

[BT98] S. Birchfield and C. Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(4) :401–406, 1998.

[CBR$^+$07] A. Criminisi, A. Blake, C. Rother, J. Shotton, and P.H. Torr. Efficient dense stereo with occlusions for new view-synthesis by four-state dynamic programming. *International Journal of Computer Vision*, 71(1) :89–110, 2007.

[CC04] S. Chambon and A. Crouzil. Color stereo matching using correlation measures. In SEE, editor, *Proceedings of the First International Conference on Complex Systems Intelligence and Modern Technological Applications*, pages 520–525, Cherbourg, France, September 2004.

[Cha05] S. Chambon. *Mise en correspondance stéréoscopique d'images couleur en présence d'occultations.* Phd thesis, Université Paul Sabatier, Toulouse, France, December 2005.

[CTB06] I. Cabani, G. Toulminet, and A. Bensrhair. A fast and self-adaptive color stereo vision matching : A first step for road obstacle detection. In *Proceedings of IEEE Intelligent Vehicle Symposium*, pages 58 –63, Tokyo, Japan, 2006.

[Dub05] E. Dubois. Frequency-domain methods for demosaicking of Bayer-sampled color images. *IEEE Signal Processing Letters*, 12(12) :847–850, December 2005.

[FHZF93] O. Faugeras, B. Hotz, Z. Zhang, and P. Fua. Real time correlation-based stereo : Algorithm, implementation and applications. Research Report RR-2013, Institut National de Recherche en Informatique et en Automatique, INRIA, August 1993.

[FKO$^+$04] S. Forstmann, Y. Kanou, J. Ohya, S. Thuering, and A. Schmitt. Real-time stereo by using dynamic programming. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, page 29, Washington, DC, USA, June 2004.

[FL01] O. Faugeras and Q.T. Luong. *The geometry of multiple images.* MIT Press, Cambridge, 2001.

[FRT97] A. Fusiello, V. Roberto, and E. Trucco. Efficient stereo with multiple windowing. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 858–863, Washington, DC, USA, June 1997.

[FRT00] A. Fusiello, V. Roberto, and E. Trucco. Symmetric stereo with multiple windowing. *International Journal of Pattern Recognition and Artificial Intelligence*, 14 :1053–1066, 2000.

[GAM02] B.K. Gunturk, Y. Altunbasak, and R.M. Mersereau. Color plane interpolation using alternating projections. *IEEE Transactions on Image Processing*, 11(9) :997–1013, September 2002.

[GB04] K.T. Gribbon and D.G. Bailey. A novel approach to real-time bilinear interpolation. In *Proceedings of the 2nd IEEE International Workshop on Electronic Design, Test and Applications*, pages 126–131, Perth, Australia, 2004.

[GY05] M. Gong and Y.H. Yang. Near real-time reliable stereo matching using programmable graphics hardware. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 924–931, San Diego, USA, June 2005.

[GY07] M. Gong and Y.H. Yang. Real-time stereo matching using orthogonal reliability-based dynamic programming. *IEEE Transactions on Image Processing*, 16(3) :879–884, March 2007.

[HÖ7] J. Häkkinen. *Half-occlusion processing in stereoscopic vision.* Phd thesis, University of Helsinki, 2007.

[HA97] J.F. Hamilton and J.E. Adams. Adaptive color plan interpolation in single sensor color electronic camera. US patent 5,629,734, to Eastman Kodak Co., Patent and Trademark Office, Washington D.C., May 1997.

[Har99] R.I. Hartley. Theory and practice of projective rectification. *International Journal of Computer Vision*, 35(2) :115–127, November/December 1999.

[Hen00] A. Henrichsen. 3d reconstruction and camera calibration from 2d images. Master's thesis, University of Cape Town, December 2000.

[Hib95] R.H. Hibbard. Apparatus and method for adaptively interpolating a full color image utilizing luminance gradients. U.S. patent 5,382,976, to Eastman Kodak Co., Patent and Trademark Office, Washington D.C., 1995.

[HIG02] H. Hirschmüller, P.R. Innocent, and J. Garibaldi. Real-time correlation-based stereo vision with reduced border errors. *International Journal of Computer Vision*, 47 :229–246, April-June 2002.

[Hir05] H. Hirschmüller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 807–814, June 2005.

[HS97] J. Heikkila and O. Silven. A four-step camera calibration procedure with implicit image correction. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1106–1112, Washington, DC, USA, June 1997.

[HS07] H. Hirschmüller and D. Scharstein. Evaluation of cost functions for stereo matching. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, Minneapolis, USA, June 2007.

[HZ04] R.I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, ISBN : 0521540518, second edition, 2004.

[IMUW08] K. Irie, A.E. McKinnon, K. Unsworth, and I.M. Woodhead. A model for measurement of noise in ccd digital-video cameras. *Measurement Science and Technology*, 19(4) :045207 (5pp), 2008.

[KLCL05] C. Kim, K.M. Lee, B.T. Choi, and S.U. Lee. A dense stereo matching using two-pass dynamic programming with generalized ground control points. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1075–1082, San Diego, CA, USA, June 2005.

[KO94] T. Kanade and M. Okutomi. A stereo matching algorithm with an adaptive window : Theory and experiment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(9) :920–932, September 1994.

[Kos93] A. Koschan. Dense stereo correspondence using polychromatic block matching. In *International Conference on Computer Analysis of Images and Patterns*, volume 719 of *Lecture Notes in Computer Science*, pages 538–542, Budapest, Hongrie, September 1993.

[Kos96] A. Koschan. Using perceptual attributes to obtain dense depth maps. In *IEEE Southwest Symposium on Image Analysis and Interpretation*, pages 155–159, San Antonio, E'tats-Unis, April 1996.

[KZ01] V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions using graph cuts. In *Proceedings of the 8th IEEE International Conference on Computer Vision*, volume 2, pages 508–515, Vancouver, Canada, July 2001.

[LAC07] S. Lefebvre, S. Ambellouis, and F. Cabestaing. A colour correlation-based stereo matching using 1D windows. In *Proceedings of the 3rd International Conference on Signal-Image Technology and Internet-based Systems*, pages 702–710, Shanghai Jiaotong University, China, December 2007.

[LCZT06] N.X. Lian, L. Chang, V. Zagorodnov, and Y.P. Tan. Reversing demosaicking and compression in color filter array image processing : Performance analysis and modeling. *IEEE Transactions on Image Processing*, 15(11) :3261–3278, November 2006.

[LH87] H.C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Readings in computer vision : issues, problems, principles, and paradigms*, pages 61–62, 1987.

[LMY10] O. Losson, L. Macaire, and Y. Yang. Comparison of color demosaicing methods. *Advances in Imaging and Electron Physics*, 162 :173–265, 2010.

[LP93] C.A. Laroche and M.A. Prescott. Apparatus and method for adaptively interpolating a full color image utilizing chrominance gradients. U.S. patent 5,373,322, to Eastman Kodak Co., Patent and Trademark Office, Washington D.C., June 1993.

[LP07] R. Lukac and K.N. Plataniotis. Single-sensor camera image processing. In Rastislav Lukac and Konstantinos N. Plataniotis, editors, *Color Image Processing : Methods and Applications*, pages 363–392. CRC Press / Taylor & Francis, 2007.

[Lyo00] R.F. Lyon. Prism-based color separation for professional digital photography. In *Proceedings of the IST Conference on Image Processing, Image Quality, Image Capture, Systems*, volume 3, pages 50–54, Portland, Oregon, USA, March 2000.

[MP79] D. Marr and T. Poggio. A computational theory of human stereo vision. *Royal Society of London*, B-204 :301–328, 1979.

[MPL04] R.Y. Marc, M. Pollefeys, and S. Li. Improved real-time stereo on commodity graphics hardware. In *Proceedings of IEEE Workshop on Computer Vision and Pattern Recognition*, pages 36–36, Washington, DC, USA, 2004.

[MPP06] W. Miled, J. Pesquet, and M. Parent. Dense disparity estimation from stereo images. In *Proceedings of 3rd International Symposium on Image/Video Communications*, September 2006.

[OA05] A.S. Ogale and Y. Aloimonos. Shape and the stereo correspondence problem. *International Journal of Computer Vision*, 65(3) :147–162, December 2005.

[OFA05] A. S. Ogale, C. Fermüller, and Y. Aloimonos. Motion segmentation using occlusions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6) :988–992, June 2005.

[OT92] M. Okutomi and G. Tomita. Color stereo matching and its application to 3d measurement of optic nerve head. In *International Conference on Pattern Recognition*, volume 1, pages 509–513, La Haye, Hollande, September 1992.

[PBCC04] M. Pérez, P. Bonnet, O. Colot, and F. Cabestaing. A similarity-based adaptive neighborhood method for correlation-based stereo matching. In *Proceedings of the 11th International Conference on Image Processing*, volume 2, pages 1341–1344, Singapore, October 2004.

[PC03] M. Pérez and F. Cabestaing. A comparison of hardware resources required by real-time dense stereo algorithms. In *Proceedings of the 6th IEEE Interna-*

tional Workshop on Computer Architectures for Machine Perception, pages 306–313, New Orleans, USA, May 2003.

[PCC05] M. Pérez, O. Colot, and F. Cabestaing. *A SBAN Stereovision Algorithm using Hue as Pixel Similarity Criterion*, volume 32 of *Computational Imaging and Vision*, pages 552–557. Springer, 2005.

[PSZ05] E. Pinhasov, N. Shimkin, and Y. Zeevi. Optimal usage of color for disparity estimation in stereo vision. In *13th European Signal Processing Conference*, Antalya, Turkey, September 2005.

[RC98] S. Roy and I.J. Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. In *Proceedings of the Sixth IEEE International Conference on Computer Vision*, pages 492–499, Bombai, India, January 1998.

[Rey00] G. Rey. Mutual information as a stereo correspondence measure. Technical report ms-cis-00-20, University of Pennsylvania, 2000.

[Sch94] D. Scharstein. Matching images by comparing their gradient fields. In *Proceedings of International Conference on Pattern Recognition*, pages 572–575, 1994.

[SS02] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47 :7–42, 2002.

[Vek99] O. Veksler. *Efficient graph-based energy minimization methods in computer vision.* Phd thesis, University of Cornell, Ithaca, USA, August 1999.

[WGGY06] L. Wang, M. Gong, M. Gong, and R. Yang. How far can we go with local optimization in real-time stereo matching : A performance study on different cost aggregation approaches. In *Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission*, pages 129–136, Chapel Hill, USA, June 2006.

[Wor07] J.A. Worby. Multi-resolution graph cuts for stereo-motion estimation. Master's thesis, University of Toronto, 2007.

[WTTW06] T.P. Wu, K.L. Tang, C.K. Tang, and T.T. Wong. Dense photometric stereo :

A markov random field approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11) :1830–1846, November 2006.

[Wu00] J. Wu. Bayesian estimation of stereo disparity from phase based measurements. Master's thesis, Queen's University, February 2000.

[XWFS02] Y. Xu, D. Wang, T. Feng, and H.Y. Shum. Stereo computation using radial adaptive windows. In *Proceedings of the 16th International Conference on Pattern Recognition*, volume 3, pages 595–598 vol.3, 2002.

[YK06] K.J. Yoon and I.S. Kweon. Adaptive support-weight approach for correspondence search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4) :650–656, April 2006.

[YLD07] Y. Yang, O. Losson, and L. Duvieubourg. Quality evaluation of color demosaicing according to image resolution. In *Proceedings of the 3rd International Conference on Signal-Image Technology & Internet-based Systems*, pages 640–646, Shanghai Jiaotong University, China, December 2007.

[Zha96] Z. Zhang. Determining the epipolar geometry and its uncertainty : A review. Technical report, INRIA Technical Report 2927, 1996.

[ZW94] R. Zabih and J. Woodfill. Non-parametric local transforms for computing visual correspondence. In *Proceedings of European Conference on Computer Vision*, pages 151–158, Stockholm, Sweden, May 1994.

Résumé

Les coûts de mise en correspondance supposent que les pixels homologues ont presque les mêmes composantes couleur. Les composantes couleur sont dégradeés par le dématriçage quand les images couleur ont été acquises par un caméra équipée d'un seul capteur. L'idée est de tenter une mise en correspondance directement à partir des images CFA. De cette façon, il serait possible de travailler sur des intensités de couleurs dont on est sûr. Le fait de travailler sur les images CFA soulève différents problèmes. Tout d'abord, le nombre d'informations est considérablement réduit par rapport à une image couleur idéale : nous en avons trois fois moins. Il faut donc que nous travaillions avec des outils capables de pallier cette carence. Ensuite, rien ne nous permet d'affirmer que des pixels homologues vont être codés dans des couleurs similaires dans les deux images stéréoscopiques CFA. Nous suggérons d'estimer partiellement les éléments manquants et ensuite de mettre en correspondance les pixels. Nous avons prouvé que ce dématriçage partiel donne de meilleurs résultats que le dématriçage total.

Mots clés : stéréovision couleur, mise en correspondance, images CFA, dématriçage.

Abstract

Most color stereovision setups include single-sensor cameras which provide Color Filter Array (CFA) images. In those images, a single color component is sampled at each pixel rather than the three required ones (R,G,B). We show that standard demosaicing techniques, used to determine the two missing color components, are not well adapted when the resulting color pixels are compared for estimating the disparity map. In order to avoid this problem while exploiting color information, we propose a partial demosaicing scheme designed for dense stereovision based on pairs of Bayer CFA images. Finally, experimental results obtained with benchmark stereo image pairs show that stereo matching applied to partially demosaiced images outperforms stereo matching applied to standard demosaiced images.

Key-words : color stereovision, stereo matching, CFA images, demosaicing.