Comment voit-on en relief

Introduction

La capacité d'un humain à percevoir le relief repose principalement sur sa vision binoculaire, appelée *stéréoscopie*. Grâce à ses yeux, il voit le monde depuis deux points de vue légèrement différents, desquels son cerveau extrait une vue unique en relief. Cette propriété a été observée dès le X^{e} siècle, notamment par un savant nommé Alhazen [22]. Dans un traité de François d'Aiguilon, publié en 1613, une illustration de Rubens dépeint ainsi un vieil homme borgne appréciant mal les distances à cause de sa monophtalmie [1] [22, page 16]. Cette particularité de la vision humaine reste cependant peu exploitée, jusqu'à l'invention de la photographie et des premiers stéréogrammes. Ces derniers sont composés de deux photographies d'une même scène, légèrement décalées l'une par rapport à l'autre. Au début du XX^{e} siècle, ils connaissent un succès important, sous le nom de cartes stéréoscopiques. Grâce à un stéréoscope, dont le premier modèle est inventé dès 1838 par Charles Wheatstone, ces cartes offrent une

1. Le fait de ne voir que d'un seul œil.

vue *en relief* des scènes photographiées, à partir de développements photographiques *plans*. Plus récemment, la stéréoscopie connaît un regain de popularité avec les films dits *en 3 dimensions* : le spectateur, équipé de lunettes spéciales, expérimente une projection du film où personnages, objets et décors semblent posséder volume et profondeur réalistes. Les constructeurs de consoles de jeu et de téléviseurs ne sont pas en reste et ont conçu des écrans offrant un rendu en relief des images affichées. Dans tous les cas, le principe est le même : ce que voit l'œil gauche diffère de ce que voit l'œil droit, ce qui permet au cerveau de reconstituer une information de relief.

Dans le domaine du traitement de l'image et de la vision par ordinateur, la stéréovision binoculaire est depuis des décennies une branche très active, notamment depuis la mise en ligne en 2001 du banc d'essai MIDDLEBURY [2] par Daniel SCHARSTEIN, Richard SZELISKI et Heiko HIRSCHMÜLLER [36]. L'objectif est reconstruire le relief d'une scène à partir de deux photographies de celle-ci, prises de deux points de vue différents, connaissant les paramètres des systèmes optiques impliqués dans la prise de vue. Nous verrons qu'il s'agit fondamentalement d'un problème de *mise en correspondance* (section 2.1). Du fait d'un phénomène appelé *occultation*, il est malheureusement mal posé (section 2.2) et, par ailleurs, difficile à résoudre. De nombreuses stratégies ont été explorées ces dernières années (section 2.3), mais nous nous pencherons dans ce mémoire sur une classe de méthodes dites *globales*, avec d'une part une approche reposant sur une relaxation convexe du problème variationnel sous-jacent (chapitre 3) et d'autre part une méthode de coupures de graphes (*graph cuts*) tirant parti de l'efficacité des algorithmes de flot maximal (chapitre 4).

2.1 La mise en correspondance stéréoscopique

2.1.1 Modèle de formation des images

Modèle sténopé Commençons par présenter le modèle de formation des images classiquement choisi en stéréovision binoculaire. Il s'agit du modèle dit *sténopé* [3]. Dans ce modèle, le système optique (l'appareil photographique) est caractérisé par son *plan image* et son *centre optique*, la distance entre ces deux éléments étant appelée *distance focale*. On appelle alors *scène* le demi-espace délimité par le plan image et ne contenant pas le centre optique. La prise de vue est ainsi modélisée : tout point physique de la scène est visible par ce système optique s'il existe une droite (qui modélise la trajectoire du rayon lumineux) reliant sans obstacle le point physique au centre optique. Son *image* par ce système optique est alors l'intersection de cette droite avec le plan image [4]. On pourra se reporter à la figure 2.1 pour mieux visualiser le modèle décrit. On utilisera par ailleurs désormais l'anglicisme *caméra* pour désigner l'appareil photographique.

Cadre et champ d'une caméra En pratique, les photographies ont un domaine fini et rectangulaire, appelé *cadre* de la caméra. Les points physiques de la scène dont la projection sur le plan image est située à l'intérieur du cadre de la caméra forment le *champ* de la caméra. Les autres points sont dit *hors-champ*. Sauf mention contraire, nous ne considérons désormais plus que les points physiques du champ de la caméra,

2. http ://vision.middlebury.edu/stereo/
3. Appelé *pin-hole model* en anglais
4. On considère à des fins de clarté l'image virtuelle des points de la scène, car elle n'est pas inversée (le haut et le bas sont en particulier préservés), contrairement à l'image réelle, qui se trouve elle sur le *plan image* du système, symétrique du plan image par rapport au centre optique.

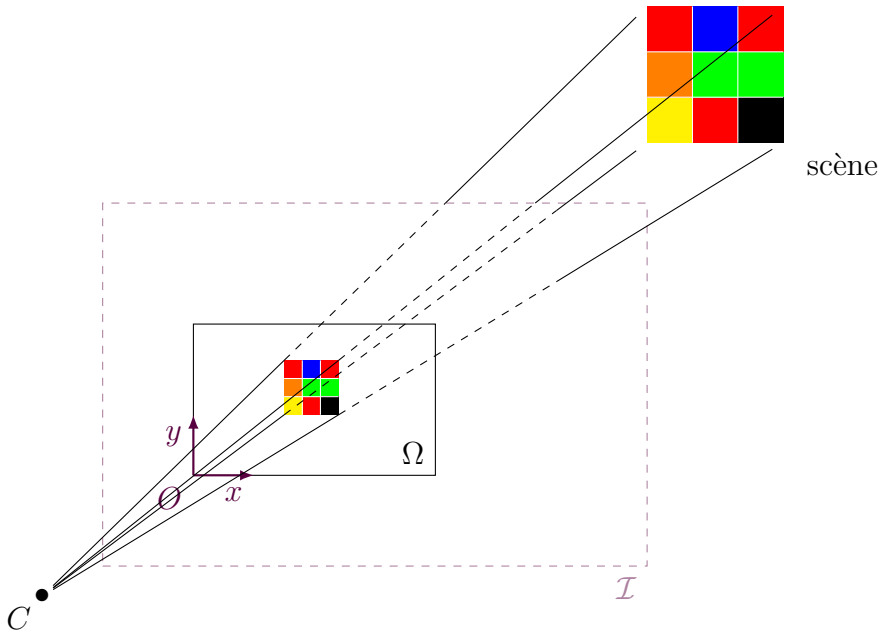


FIGURE 2.1 – Modèle sténopé de formation des images. Le système optique est modélisé par son centre optique C, son plan image $\mathcal{I}$ et sa distance focale f. La scène correspond au demi-espace ne contenant pas C et délimité par le plan $\mathcal{I}$. Le domaine Ω de l'image est matérialisé par le rectangle en trait plein, qui est muni d'un repère orthonormé. On choisit par convention de placer l'origine de ce repère au coin inférieur gauche de ce domaine, les deux directions du repère étant données par les côtés du rectangle.

et le terme *domaine de l'image* désignera la restriction du plan image au cadre de la caméra. Notons que le fait d'être situé dans le champ de la caméra n'assure pas à un point d'être visible par celle-ci.

Paramètres intrinsèques On munit le plan image d'un repère orthonormé. Son origine est le coin inférieur gauche du cadre de la caméra et les deux axes sont portés par les deux côtés issus de l'origine. Appelons *point principal* le projeté du centre optique sur le plan image. La distance focale et les coordonnées dans le repère précédemment introduit du point principal sont appelées *paramètres intrinsèques* de la caméra. La donnée des paramètres intrinsèques d'une caméra et de son cadre est suffisante pour en déduire toutes les caractéristiques du système optique étudié. On notera que, dans le cas des caméras réelles, le centre optique se projette généralement sur le centre du cadre[5], auquel cas on parlera de *caméra parfaite*.

Intensité d'un pixel On distinguera le *point physique* de la scène $M \in \mathbb{R}^3$, de coordonnées (X,Y,Z) dans un repère donné de l'espace, de sa projection (si elle existe) $m \in \mathbb{R}^2$ sur le plan image, de coordonnées (x,y) dans le repère de l'image, que l'on appellera *pixel*. Une *image* I désigne une fonction qui, à tout pixel du cadre de la caméra, associe son intensité, enregistrée par la caméra. L'intensité désigne de manière générique le niveau de gris dans le cas des images en niveaux de gris ou la couleur dans le cas des images couleurs. On choisit comme système de représentation des couleurs le système RGB (*red, green, blue*). L'image I est donc une fonction, définie sur le domaine rectangulaire $\Omega \subset \mathbb{R}^2$, et à valeurs dans $\mathbb{R}$ ou dans $\mathbb{R}^3$. En l'absence de bruit ou d'aberration

5. C'est pourquoi le point principal est parfois appelé *centre de l'image*.

chromatique et sauf cas particulier (surface réfléchissante, par exemple), l'intensité d'un pixel ne dépend que du point M correspondant, et ne varie donc pas selon le point de vue.

2.1.2 Géométrie épipolaire et rectification des images

Paire stéréoscopique Supposons maintenant que la scène est photographiée par deux caméras, caractérisées par leur plan image, leur centre optique, leur distance focale et le domaine de leur image. On supposera ce dernier de dimension identique pour les deux caméras. La scène est alors définie comme l'intersection des champs associés aux deux caméras. On impose pour le moment les contraintes suivantes :

- les deux centres optiques sont distincts ;
- chaque centre optique n'appartient pas à la scène de l'autre caméra ;
- on écarte le cas trivial où la scène est vide.

La première condition élimine le cas d'une simple rotation de la caméra autour de son centre optique. La seconde évite en particulier que l'une des deux caméras soit visible par l'autre (et notamment que les deux caméras se fassent face).

En pratique, les images sont capturées soient par la même caméra, qui se déplace dans l'espace, soit par deux caméras simultanément. Dans le premier cas, les paramètres intrinsèques de la caméra restent inchangés, mais les objets de la scène peuvent avoir bougé entre deux prises de vue (par exemple : des voitures pour les vues aériennes). Dans le second cas, les paramètres intrinsèques des deux caméras peuvent être différents.

Droites épipolaires, plan épipolaire Soit M un point de la scène. Les contraintes présentées plus haut assurent que le point M et les deux centres optiques, notés O_{L} et O_{R}, ne peuvent être alignés, car la droite $(O_{\mathrm{L}},O_{\mathrm{R}})$ ne peut être dans le champ des deux caméras à la fois. Ils définissent donc un plan, que l'on appelle *plan épipolaire* associé au point M. Ce plan coupe le plan image de la caméra de gauche selon une droite, appelée *droite épipolaire* de l'image de gauche associée au point M et notée $\ell_L(M)$ et coupe de la même manière le plan image de la caméra de droite selon la droite épipolaire de l'image de droite associée au point M et notée $\ell_R(M)$. Le pixel m_{L}, image du point M par la caméra de gauche, appartient à la droite épipolaire $\ell_L(M)$, tandis que l'image m_{R} du point M par la caméra de droite, appartient à la droite épipolaire $\ell_R(M)$.

Déplacement fronto-parallèle de la caméra Dans le cas général, pour un point M donné, les droites épipolaires associées ont des directions totalement arbitraires. On va à présent imposer certaines contraintes sur les droites épipolaires et en déduire les conditions nécessaires sur les deux systèmes optiques que cela entraîne.

On demande dans un premier temps que, pour tout point M, les droites épipolaires soient confondues dans les deux images. Ces droites appartenant chacune au plan image de sa caméra associée, on en déduit que les deux plans image doivent être confondus.

On souhaite dans un second temps contraindre toutes les droites épipolaires à être horizontales, c'est-à-dire parallèles à l'axe horizontal du repère de leur image respective. Supposons donc que c'est le cas. Soient M et M' deux points dont les droites épipolaires ℓ et ℓ' (qui sont maintenant les mêmes dans les deux images) sont distinctes et horizontales, situées dans le plan image commun des deux caméras. Les deux plans épipolaires associés contiennent par définition les centres optiques C_{L} et C_{R}, ils se

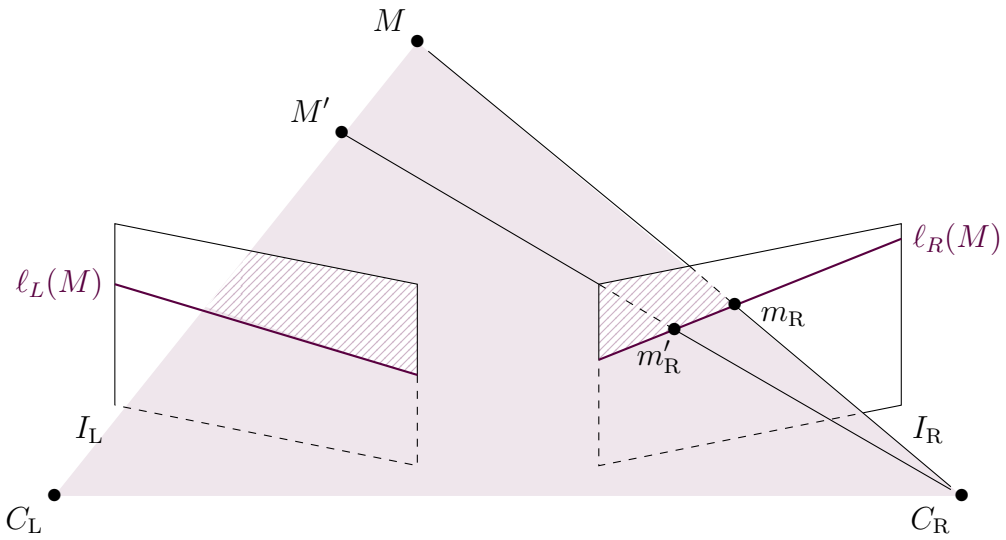


FIGURE 2.2 – Géométrie épipolaire. Le plan épipolaire, représenté ici par un triangle plein, est le plan passant par les trois points non alignés C_{L}, M et C_{R}. Il coupe chacun des deux plans images selon une droite, $\ell_L(M)$ et $\ell_R(M)$, appelées droites épipolaires. Tous les points de la scène appartenant à la droite (MC_{R}) (resp. (MC_{L})) ont pour projection un point de la droite épipolaire $\ell_L(M)$ (resp. $\ell_R(M)$).

coupent donc selon la droite $(C_{\mathrm{L}}C_{\mathrm{R}})$, appelée *baseline*. Or, les deux plans épipolaires sont parallèles par hypothèse aux droites épipolaires ℓ et ℓ', d'où l'on en conclut que c'est également le cas de leur intersection. La *baseline* est donc parallèle au plan image commun, ce qui implique que les deux systèmes optiques ont même distance focale. On en déduit également que la *baseline* est parallèle à l'axe horizontal commun du repère de chacune des images.

Lorsque les deux caméras sont dans cette configuration particulière, leurs paramètres intrinsèques (distance focale et coordonnées du point principal) sont identiques. On parle alors de *déplacement fronto-parallèle* de la caméra (cf. figure 2.3). En effet, si la scène est statique, on peut considérer qu'il s'agit de la même caméra que l'on a translatée selon la direction horizontale du repère associé à son image. Réciproquement, on montre que, lorsque les deux caméras ont mêmes paramètres intrinsèques et que le repère associé à l'image de droite est la translatée horizontale du repère associé à l'image de gauche, alors les droites épipolaires sont confondues dans les deux images et sont horizontales.

Rectification épipolaire Dans le cas général, il est possible de se ramener au cas où les droites épipolaires sont confondues d'une image à l'autre et horizontales, *via* une étape de *rectification épipolaire*. Cette opération consiste à déterminer deux homographies [30], qui permettent de transformer les deux images afin d'aligner les droites épipolaires. Cela revient à simuler deux nouvelles caméras et leur image respective. Les homographies sont estimées en mettant en correspondance des points SIFT [25] des deux images.

Il faut cependant noter que la rectification épipolaire est stable par translation horizontale et par translation verticale *simultanée* des deux images. En d'atures termes, l'abcisse des points principaux des caméras simulées est arbitraire, de même que leur ordonnée (commune). Ainsi, bien qu'elles aient même distance focale, on ne peut plus parler de déplacement fronto-parallèle, car les paramètres intrinsèques de deux caméras

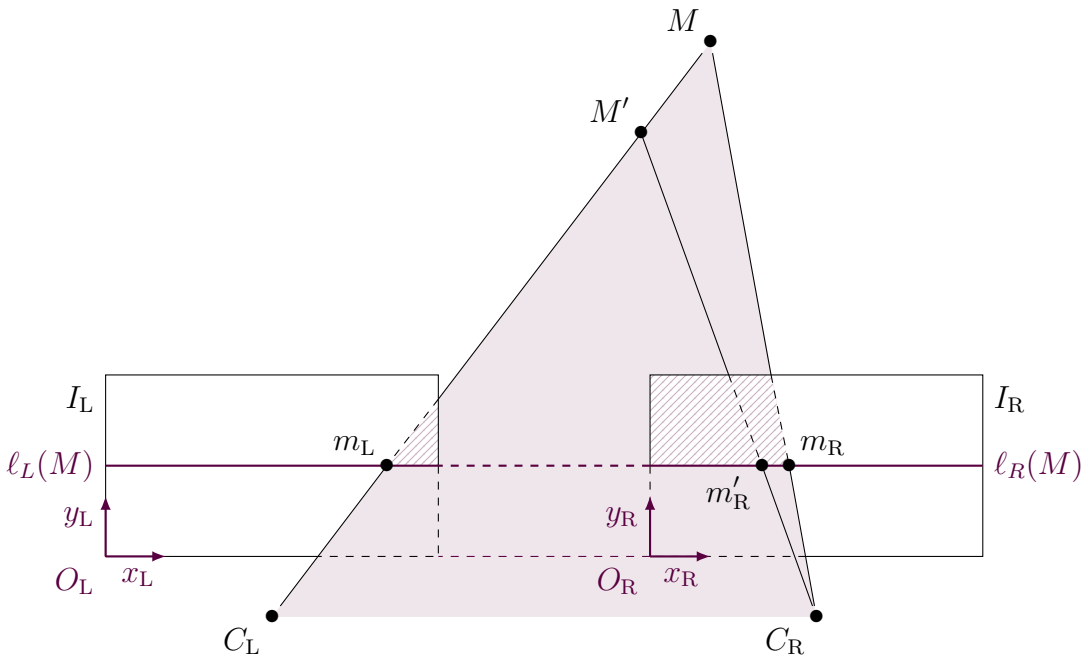


FIGURE 2.3 – Déplacement fronto-parallèle de la caméra. Les deux droites épipolaires $\ell_L(M)$ et $\ell_R(M)$ sont confondues, parallèles à l'axe horizontal (commun) des deux repères respectifs $(O_{\mathrm{L}}; \vec{x_{\mathrm{L}}}, \vec{y_{\mathrm{L}}})$ et $(O_{\mathrm{R}}; \vec{x_{\mathrm{R}}}, \vec{y_{\mathrm{R}}})$ des deux images. Tout pixel de l'image de droite situés sur la droite épipolaire $\ell_R(M)$ est la projection d'un point appartenant à la droite $(C_{\mathrm{L}}M)$, et réciproquement, la projection de tout point de la scène appartenant à la droite $(C_{\mathrm{L}}M)$ est située sur $\ell_R(M)$. Les paramètres intrinsèques des deux caméras étant identiques, on peut se ramener au cas d'une unique caméra, translatée du vecteur $\vec{C_{\mathrm{L}}C_{\mathrm{R}}}$ parallèle à l'axe $(O_{\mathrm{L}}; \vec{x_{\mathrm{L}}})$.

simulées sont potentiellement différents. Il est néanmoins facile de se ramener dans ce cas grâce à une translation (horizontale) du repère d'une des caméras simulées.

La contrainte de déplacement fronto-parallèle de la caméra est naturelle. Elle correspond d'une part à la configuration de la vision humaine (où la paire d'images est obtenue grâce à nos deux yeux). Un rendu en relief naturel tel que ceux proposés par l'industrie cinématographique suppose que les caméras sont en déplacement fronto-parallèle l'une par rapport à l'autre, avec un écartement équivalent à celui des yeux. D'autre part, ainsi qu'on va le voir dans le paragraphe suivant, cette configuration simplifie la reconstruction du relief. C'est pourquoi la plupart des algorithmes de stéréovision suppose que les images sont rectifiées au préalable. Nous en ferons de même dans tout ce qui suit.

2.1.3 Disparité et mise en correspondance

Le principe central de la vision stéréoscopique est la parallaxe, c'est-à-dire le mouvement apparent des objets lorsqu'ils sont vus depuis des points de vue différents. Explicitons ce phénomène. On rappelle que l'on se place désormais dans le cas d'un déplacement fronto-parallèle de la caméra (avec éventuellement une translation horizontale du point principal).

Quelques remarques préliminaires Les droites épipolaires sont confondues dans les deux images (on ne précisera donc plus l'image concernée) et sont horizontales. En particulier, les axes horizontaux des deux images sont confondus. Il s'ensuit que tout

(a) (b) (c) (d)

FIGURE 2.4 – Rectification épipolaire d'une paire stéréoscopique. (a) et (b) : Paire originale. (c) et (d) : Paire rectifiée. Les droites épipolaires sont maintenant horizontales et confondues, mais le point principal a été translaté horizontalement. (Code : [30])

point du plan image a même ordonnée dans chacun des deux repères des images. On ne précisera donc plus le repère lorsque l'on mentionnera l'ordonnée des points dans le plan image. Par ailleurs, puisque la projection (lorsqu'elle existe) d'un point physique est située sur la droite épipolaire de la caméra associée, on en déduit que tout point visible de la scène se projette sur la même ligne dans les deux images.

Soit M un point de la scène. On suppose qu'il est visible depuis les deux caméras. Notons $m_{\mathrm{L}}(x_{\mathrm{L}},y_{\mathrm{L}})$ sa projection sur le plan image de gauche, et $m_{\mathrm{R}}(x_{\mathrm{R}},y_{\mathrm{R}})$ sa projection sur le plan image de droite. On note e la droite épipolaire associée au point M. Les remarques d'introduction assurent que $y_{\mathrm{L}} = y_{\mathrm{R}}$, qui est également l'ordonnée de la droite épipolaire. Plaçons-nous à présent dans le plan épipolaire (cf. figure 2.5). Celui-ci coupe le plan image selon la droite e. Il contient par définition les deux centres optiques C_{L} et C_{R}, et en particulier la *baseline*. Cette dernière est parallèle à la droite épipolaire (car parallèle à la fois au plan épipolaire et au plan image d'après l'analyse menée dans le paragraphe précédent), et la distance entre la *baseline* et la droite épipolaire vaut exactement la distance focale f. Notons b la distance entre les deux centres optiques. L'intersection entre le plan image et le plan épipolaire est la droite $(m_{\mathrm{L}}m_{\mathrm{R}})$, dont l'intersection avec le domaine de chaque image est un segment. Notons o_{L} et o_{R} les extrémités gauche de ces deux segments. Il s'agit des pixels des coordonnées $(0,y_{\mathrm{L}})$ dans chacun des deux repères. Le vecteur $\overrightarrow{o_{\mathrm{L}}m_{\mathrm{L}}}$ est donc un vecteur horizontal, d'abscisse x_{L}, tandis que le vecteur (horizontal également) $\overrightarrow{o_{\mathrm{R}}m_{\mathrm{R}}}$ a pour abscisse x_{R}.

Disparité Désormais, les deux caméras n'auront plus un rôle symétrique. On choisit la caméra de gauche comme la caméra de *référence*. On définit alors la *disparité* du point M (ou, indifféremment, du pixel m_{L} dans l'image de référence), notée $u(M)$ (ou $u(m_{\mathrm{L}})$), comme le déplacement apparent de sa projection entre la vue de droite et la vue de gauche, lorsqu'il est visible depuis les deux caméras. Plus précisément, avec les notations introduites ici, on choisit la convention

$$u(M) = u(m_{\mathrm{L}}) = \begin{pmatrix} x_{\mathrm{L}} - x_{\mathrm{R}} \\ y_{\mathrm{L}} - y_{\mathrm{R}} \end{pmatrix} \in \mathbb{R}^2.$$

Les remarques qui précèdent assurent que la disparité est un vecteur horizontal :

$$u(M) = u(m_{\mathrm{L}}) = \begin{pmatrix} x_{\mathrm{L}} - x_{\mathrm{R}} \\ 0 \end{pmatrix} \in \mathbb{R}^2$$

et on confondra désormais le vecteur disparité et sa coordonnée horizontale.

Distance à la caméra Montrons que la disparité ne dépend que de la distance du point M à la *baseline*. On note h cette distance, et on l'appelle, par abus de langage,

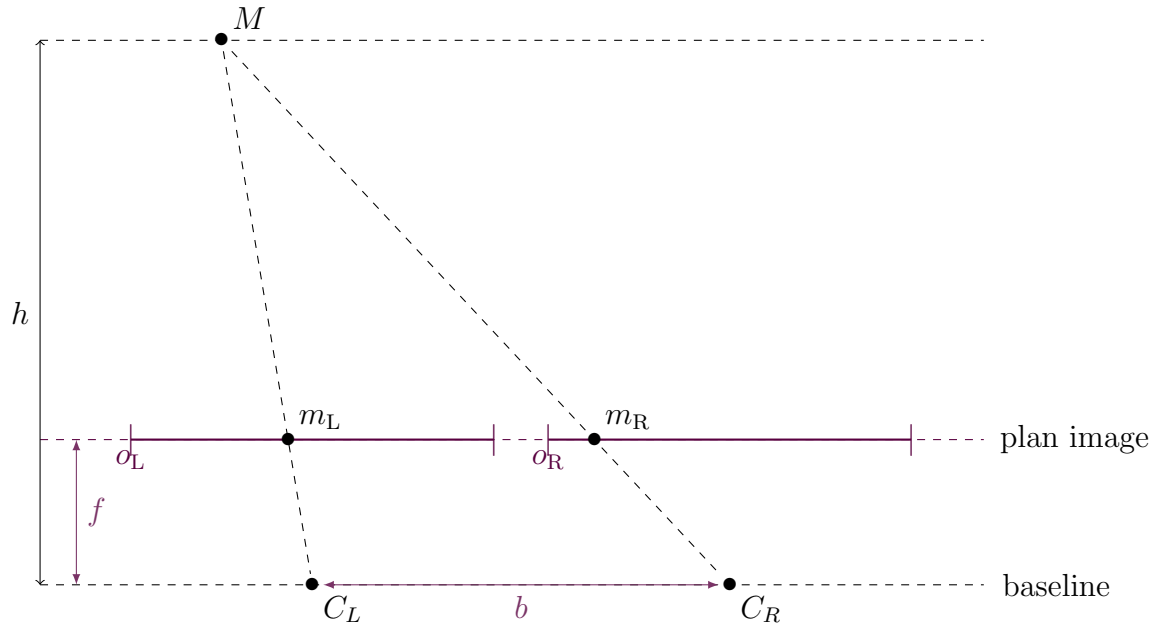


FIGURE 2.5 – Disparité et distance à la caméra. On se place ici dans le plan épipolaire. On note m_{L} et m_{R} les projections respectives du point M par chacune des deux caméras, de centres optiques C_{L} et C_{R}. L'intersection des domaines des images avec le plan épipolaire est réduit à deux segments, matérialisés ici par deux traits pleins, dont les extrémités gauche sont notés o_{L} et o_{R}. La distance du point M à la *baseline*, appelée par abus de langage *distance à la caméra*, est notée h. En utilisant le théorème de THALÈS, on peut exprimer la distance entre les deux images m_{L} et m_{R} en fonction de la distance entre les deux caméras b, la distance focale f et la distance à la caméra h. On en déduit alors la valeur de la *disparité* du point M, qui vaut par définition $|o_{\mathrm{L}}m_{\mathrm{L}}| - |o_{\mathrm{R}}m_{\mathrm{R}}|$.

distance du point M à la caméra. On commence par remarquer que la quantité $x_\mathrm{L} - x_\mathrm{R}$ vaut $|o_\mathrm{L}m_\mathrm{L}| - |o_\mathrm{R}m_\mathrm{R}|$. Le point m_L étant toujours situé entre les points o_L et m_R, on en déduit que

$$|o_\mathrm{L}m_\mathrm{L}| = |o_\mathrm{L}m_\mathrm{R}| - |m_\mathrm{L}m_\mathrm{R}|;$$

de manière similaire, le point o_R est toujours situé entre les deux points o_L et m_R, donc

$$|o_\mathrm{L}m_\mathrm{R}| = |o_\mathrm{L}o_\mathrm{R}| - |o_\mathrm{R}m_\mathrm{R}|.$$

Par conséquent, la disparité du pixel m_L vaut

$$u(m_\mathrm{L}) = |o_\mathrm{L}o_\mathrm{R}| - |m_\mathrm{L}m_\mathrm{R}|.$$

Si les deux caméras sont en déplacement fronto-parallèle, la position du centre optique par rapport au cadre est la même dans les deux caméras (car la distance focale et la position du point principal dans le cadre sont les mêmes). On en déduit que les droites $(C_\mathrm{L}o_\mathrm{L})$ et $(C_\mathrm{R}o_\mathrm{R})$ sont parallèles, ce qui entraîne que la distance $|o_\mathrm{L}o_\mathrm{R}|$ vaut exactement la distance entre les deux centres optiques $|C_\mathrm{L}C_\mathrm{R}| = b$. Si les caméras n'ont pas le même point principal (après une rectification épipolaire par exemple), la distance $|o_\mathrm{L}o_\mathrm{R}|$ vaut une valeur positive arbitraire que l'on peut calculer si on connaît les paramètres intrinsèques des deux caméras. Elle ne dépend cependant pas du point M et est donc considérée comme constante. Calculons à présent la longueur $|m_\mathrm{L}m_\mathrm{R}|$. En appliquant le théorème de Thalès dans les triangles $C_\mathrm{L}MC_\mathrm{R}$ et $o_\mathrm{L}Mo_\mathrm{R}$, on obtient la relation de proportionnalité suivante :

$$\frac{|m_\mathrm{L}m_\mathrm{R}|}{b} = \frac{h-f}{h}$$

d'où l'on déduit l'expression suivante de la disparité :

$$u(m_\mathrm{L}) = |o_\mathrm{L}o_\mathrm{R}| - b + \frac{f\,b}{h}. \tag{2.1}$$

Autrement dit, si les deux caméras sont en déplacement fronto-parallèle, alors la disparité du point M est inversement proportionnelle à sa distance à la caméra ; si ce n'est pas le cas, il y a un terme (constant) $|o_\mathrm{L}o_\mathrm{R}|-b$ qui s'ajoute. Ainsi, dans tous les cas, si on connaît la disparité d'un point M, on peut retrouver grâce aux paramètres intrinsèques des deux caméras sa distance à la caméra en inversant la formule précédente :

$$h = \frac{f\,b}{u(m_\mathrm{L}) + b - |o_\mathrm{L}o_\mathrm{R}|}.$$

En d'autres termes, si on parvient à calculer la disparité de tous les points de la scène, on en connaît précisément le relief.

Mise en correspondance Déterminer le relief d'une scène donnée par deux caméras repose donc sur deux problèmes complémentaires :

1. calculer ou mesurer les paramètres intrinsèques des caméras réelles et/ou virtuelles, dans le cas d'une rectification épipolaire ;
2. calculer la carte de disparité de la vue de référence dans la paire d'images rectifiée.

Ce dernier problème revient à trouver, pour chaque point M de la scène visible depuis les deux caméras, ses deux images m_L et m_R. De manière équivalente, cela revient à déterminer, pour chaque pixel m_L de l'image de référence, le pixel m_R, appelé *pixel homologue* du pixel m_L, tel que les deux pixels m_L et m_R soient images du même point physique. Ce processus est appelé *mise en correspondance*.

Les droites épipolaires étant confondues dans les deux images, le pixel homologue, s'il existe, d'un pixel de l'image de gauche est situé sur la même ligne que celui-ci. La recherche d'un pixel homologue se fait donc sur une seule ligne de l'image de droite. Elle peut même être restreinte à un intervalle appelé *intervalle de disparité* si des mesures préalables ont permis d'estimer la disparité minimale et maximale. La formule (2.1) assure en effet que la disparité est une fonction décroissante de la distance à la caméra. Elle est en particulier minimale pour les objets à l'infini, et vaut alors $|o_{\mathrm{L}}o_{\mathrm{R}}| - b$. Si l'objet le plus proche de la caméra est situé à une distance h_0, alors la disparité est par ailleurs majorée par la quantité $|o_{\mathrm{L}}o_{\mathrm{R}}| - b + f\,b/h_0$. Ainsi, si on est capable d'estimer la distance (ou la disparité) de l'objet le plus proche de la caméra, il est possible d'obtenir un intervalle de disparité $\mathrm{I}_{\mathrm{disp}} = [\,u_{\min}\,;u_{\max}\,]$. Le pixel homologue de tout pixel $m_{\mathrm{L}}(x_{\mathrm{L}},y_{\mathrm{L}})$ de l'image de gauche est alors à rechercher parmi les pixels de l'image de droite de coordonnées (x,y_{L}), avec $x \in x_{\mathrm{L}} - [\,u_{\min}\,;u_{\max}\,]$.

2.1.4 Difficultés à surmonter

Le problème de mise en correspondance est intrinsèquement difficile à résoudre, car il s'agit d'associer deux pixels issus du même point physique sans autre information que leurs intensités respectives (il n'y a pas de modèle de la scène). À cela, il faut ajouter des difficultés supplémentaires qui rendent la tâche encore plus ardue.

Mouvement des objets La scène doit être supposée statique pour que le lien entre disparité et distance à la caméra établi au paragraphe précédent soit vrai. Or, si les images ne sont pas prises simultanément (par exemple, lorsqu'il s'agit d'un satellite qui prend une image par passage au-dessus d'un certain point du sol), il y a de fortes chances pour des objets aient bougé entre-temps (avec de plus un mouvement apparent qui ne soit pas dans la direction épipolaire). Même si la mise en correspondance est correctement réalisée, la disparité des objets qui ont bougé n'est plus inversement proportionnelle à leur distance à la caméra.

Changement d'illumination ou de contraste Si les images sont prises simultanément, cela suppose qu'elles sont prises par deux caméras différentes. L'étape de rectification permet de simuler des caméras aux mêmes paramètres intrinsèques (à une translation horizontale du point principal près), mais ne peut pas corriger les différences de qualité ou de dynamique entre les deux images. Par exemple, certaines caméras adaptent automatiquement le contraste ou la balance des blancs. Cette opération ne peut être réalisée exactement de la même manière sur les deux caméras (même s'il s'agit du même modèle). Dans ce cas, les deux images peuvent présenter des aspects très différents, ce qui rend la mise en correspondance plus difficile.

C'est également le cas lorsque les images n'ont pas été prises au même moment : le soleil peut avoir changé de position dans le ciel, celui-ci peut s'être couvert. La scène, même en restant immobile, change alors visuellement d'aspect.

Reflets Certaines surfaces comme les vitres ou des métaux brillants renvoient partiellement la lumière qu'elles reçoivent. Cela a deux conséquences sur la mise en correspondance stéréoscopique. Tout d'abord, une surface complètement réfléchissante donne l'illusion d'une apparente profondeur. C'est ce que l'on observe par exemple en plaçant un miroir sur un mur : le regard porte au-delà du mur et on a l'illusion que la scène se prolonge *dans le mur*. Dans ce cas également, cela implique que la disparité n'est plus

une information pertinente sur la distance à la caméra : le miroir semblera plus éloigné qu'il ne l'est réellement. Ensuite, si la surface n'est pas plate, elle renvoie la lumière différemment selon la direction sous laquelle on l'observe. En d'autres termes, les reflets ne sont pas situés au même endroit suivant l'image. Or, sans information permettant d'interprêter le reflet comme tel, la mise en correspondance dicte d'associer les deux reflets, alors qu'ils ne correspondent pas au même point physique.

Végétation La végétation et les objets fins peuvent également présenter des aspects très différents suivant le point d'observation. Ils sont en effet composés de surfaces (parfois réfléchissantes) de très petites tailles (comme le feuillage), qui sont orientés dans de nombreuses directions. Ce sont donc généralement des zones difficiles à mettre en correspondance, car il est difficile d'identifier le pixel homologue qui change beaucoup d'apparence.

Régions plates et effet de Strobes Dans le cas des régions peu ou pas texturées (d'une couleur unie par exemple), le problème est inverse : il est difficile de sélectionner le pixel homologue car, visuellement, il y a beaucoup de candidats possibles (on parle de problème d'ouverture). Le cerveau rencontre parfois ce problème : lorsqu'on regarde de près un mur blanc, lisse (mais mat, donc sans reflets), il arrive que l'on se mette à loucher et à éprouver un léger vertige. Le cerveau ne parvient pas à mettre correctement en correspondance les deux images qu'il possède du mur. Il hésite entre plusieurs solutions, et c'est cette hésitation qui donne au mur un mouvement apparent (il semble avancer et reculer) qui donne le tournis. Il suffit alors de remarquer une petite aspérité dans le mur pour que le regard accroche et que le malaise cesse.

Ce phénomène se produit également lorsqu'une région présente une répétition de motifs identiques (comme par exemple des rayures). On parle alors d'effet de Strobes.

Bruit Enfin, il faut signaler que toute image numérique présente du bruit, ce qui signifie que l'intensité capturée n'est pas exactement celle du point physique. Il existe de nombreux types de bruit possibles, parmi lesquels on peut citer le bruit thermique (dû à l'agitation naturelles des électrons dans les capteurs), le bruit électronique (lorsque le nombre de photons est trop faible), le bruit de lecture (qui se produit pendant la conversion numérique du signal acquis), le bruit de quantification (dû à la discrétisation des valeurs du signal). Le bruit total est aléatoire, donc les deux images du même point ne sont pas affectées de la même manière.

On voit que les difficultés rencontrées peuvent être classées suivant trois catégories :

- la mise en correspondance est possible, mais la disparité ne donne aucune information significative sur la distance de l'objet à la caméra (mouvement dans la scène, reflets) ;
- la mise en correspondance n'est pas possible car les pixels homologues sont visuellement trop dissemblables (changement d'illumination ou de contraste, végétation, bruit important) ;
- la mise en correspondance n'est pas possible car il y a trop de pixels candidats et aucun moyen de les départager (régions plates, motifs répétés).

Dans ce qui suit, on supposera que les images sont prises simultanément et par la même caméra, ce qui revient à supposer que les objets n'ont pas bougé et que le contraste et l'illumination de la scène restent les mêmes. Les reflets ne seront pas spécifiquement

gérés, ni l'effet de STROBES, mais on verra que ce sont des difficultés atténuées par les approches globales. Malgré ces hypothèses simplificatrices, le prochain paragraphe montre que le problème est en réalité mal posé.

2.2 Le phénomène d'occultation

2.2.1 Occultation, désoccultation

Déterminer la disparité d'un point suppose que ce point est visible par les deux caméras. Or, dès que la scène possède un relief, certains points créent des obstacles entre d'autres points et au moins une des caméras. Il s'agit du phénomène d'*occultation*[6]. Plus précisément, on peut classer les points de la scène en quatre catégories :

- les points visibles depuis les deux caméras ;
- les points invisibles depuis les deux caméras ;
- les points uniquement visibles depuis la caméra de référence ;
- les points uniquement visibles depuis l'autre caméra.

Les points visibles depuis la vue de référence mais invisibles depuis l'autre vue (et, par extension, leur image dans la vue de référence) sont qualifiés d'*occultés*. Ceux qui, à l'inverse, ne sont visibles que depuis l'autre vue sont dits *désoccultés*. Les objets à l'origine de l'occultation seront appelés *occultants*.

Puisque la carte de disparité n'est calculée que sur l'image de référence, seuls les points occultés seront considérés. Ces points n'ont par définition aucun pixel homologue dans l'autre vue, ce qui implique que leur disparité n'est pas définie. Le problème de mise en correspondance est donc mal posé.

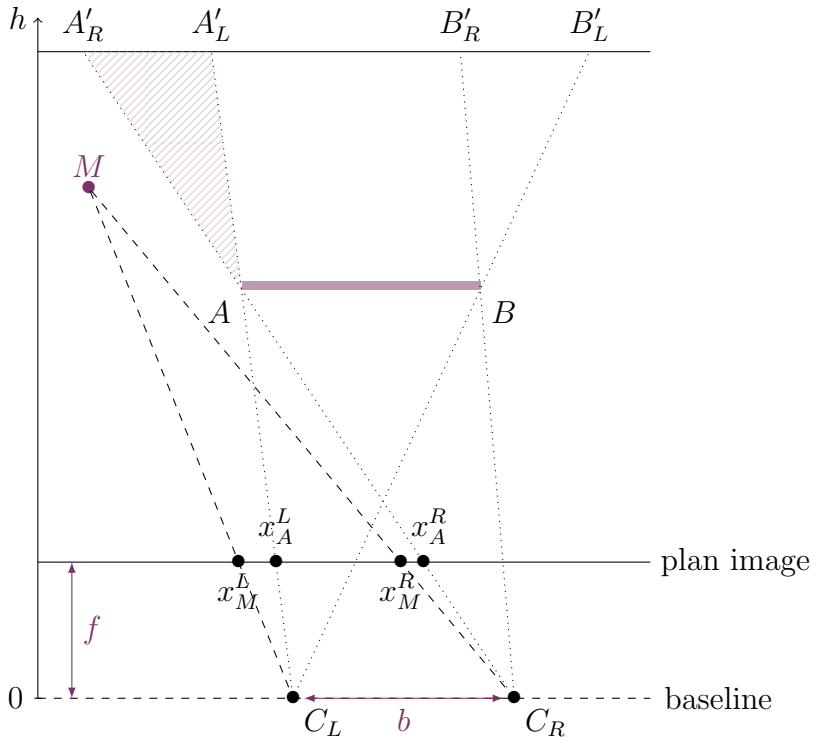
2.2.2 Préservation de l'ordre et largeur des objets

Largeur de l'objet Pour simplifier notre analyse, nous allons partir du cas simple d'un objet d'épaisseur nulle, par exemple un rectangle parallèle au plan image. Ce choix est motivé par le fait que la plupart des méthodes de stéréovision supposent que les objets de la scène ont une disparité constante, du moins localement. On supposera par ailleurs que l'objet est entièrement visible depuis les deux vues. On se place désormais dans un plan épipolaire coupant l'objet étudié, ce qui nous permet de nous ramener à des représentations planes.

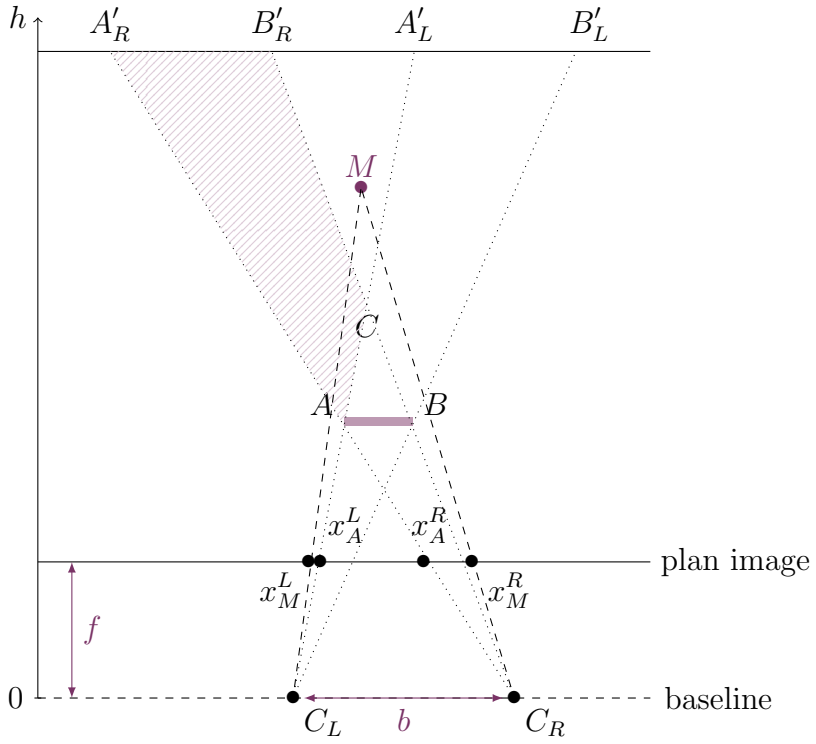
On définit à présent un objet *large* comme étant un objet dont la largeur (c'est-à-dire la taille selon l'axe horizontal) est supérieure ou égale à la distance b entre les centres optiques de deux caméras. Un objet *fin* est alors de largeur inférieure strictement à b.

Préservation de l'ordre Considérons la figure 2.6. On note $[AB]$ l'objet étudié. Plaçons-nous dans le cas où $|AB|$ est supérieur à b (figure 2.6(a)). Intéressons-nous tout d'abord aux points visibles par chacune des deux caméras. La région délimitée par les demi-droites $[AA'_L)$ et $[BB'_L)$ et le segment $[AB]$ est invisible depuis la vue de gauche, tandis que celle délimitée par les demi-droites $[AA'_R)$ et $[BB'_R)$ et le segment $[AB]$ est invisible depuis la vue de droite. Par conséquent, la région occultée (hachurée) est délimitée par les demi-droites $[AA'_L)$ et $[AA'_R)$. Considérons à présent un point non occulté M dans le voisinage de $[AB]$, d'images respectives x_M^L et x_M^R par les caméras

6. En anglais, l'occultation est nommée *occlusion*.



(a)



(b)

FIGURE 2.6 – Préservation de l'ordre. (a) Lorsque l'objet est large, le point M est vu à gauche du point A dans les deux vues. (b) Ce n'est plus le cas lorsque l'objet est fin : si M est situé dans un secteur très spécifique de la scène, alors il est vu à gauche de A dans la vue de gauche mais à droite de A dans la vue de droite.

de gauche et de droite. Si, dans la vue de référence, le point M est placé à gauche de l'objet $[AB]$, alors, nécessairement, il est situé dans la zone située à gauche du segment $[C_{\mathrm{L}}A]$ et de la demi-droite $[AA'_R)$. Or, M possède une image par la caméra de droite également située à gauche de $[AB]$, car la demi-droite $[C_{\mathrm{R}}M)$ est également située à gauche de $[C_{\mathrm{R}}A)$. On dit que l'*ordre est préservé* dans les deux images [7].

Lorsque l'objet est fin, comme c'est le cas dans la figure 2.6(b), il existe une zone située derrière l'objet qui est non occultée. Elle est délimitée par les deux demi-droites $[CA'_L)$ et $[CB'_R)$, où C est l'intersection des droites $(C_{\mathrm{L}}A)$ et $(C_{\mathrm{R}}B)$. Considérons M un point de cette zone. La demi-droite $[C_{\mathrm{L}}M)$ est située à gauche de $[C_{\mathrm{L}}A)$, donc depuis la vue de gauche, le point M est vu *à gauche* de l'objet $[AB]$. En revanche, la demi-droite $[C_{\mathrm{R}}M)$ est située à droite de $[C_{\mathrm{R}}A)$, donc, dans la vue de droite, le point M est vu *à droite* de l'objet $[AB]$: l'ordre est *inversé*. Cette configuration persiste tant que la zone délimitée par $[CA'_L)$ et $[CB'_R)$ existe, ce qui impose que $|AB|$ est strictement inférieur à b.

On voit donc que la largeur de l'objet joue sur la préservation de l'ordre des points dans les deux images. Il est à noter que, si les objets larges ne permettent pas une inversion de cet ordre, la présence d'objets fins ne garantit pas que cet ordre sera nécessairement inversé. Il faut pour cela qu'un point de la scène (donc visible depuis les deux vues) se situe dans le triangle $A'_L C B'_R$, ce qui n'est plus le cas si un autre objet occulte cette région.

Sauf mention contraire, on se placera désormais dans le cas où l'ordre est préservé dans les deux images.

2.2.3 Occultation et contrainte de visibilité

Comme on le verra dans la section 2.3, le phénomène d'occultation est généralement ignoré dans les méthodes de mise en correspondance stéréoscopique. Dans l'approche que nous proposons au chapitre 3, nous avons choisi de l'intégrer au modèle considéré. Nous développons donc ici une analyse préliminaire très fine de l'occultation. Commençons par étudier les conditions nécessaires à la présence d'occultation.

Si un point M est visible depuis les deux vues, alors seul le voisin de gauche de son image x^L_M dans l'image de référence peut être occulté, car l'occultation se produit toujours sur le bord gauche des objets dans la vue de gauche. Soit M' le point dont l'image $x^L_{M'}$ est un voisin à gauche de x^L_M (sur la même ligne). Démontrons que, s'il n'est pas occulté, alors sa disparité doit vérifier une certaine contrainte, appelée *contrainte de visibilité* [8].

Puisque M est visible depuis les deux scènes, on a $d(M) = x^L_M - x^R_M$ sa disparité. Par ailleurs, sur la droite $(C_{\mathrm{R}}M)$, seuls les points situés sur le segment $[Mx^R_M]$ sont visibles par la caméra de droite. Supposons que M' est également visible depuis les deux vues. Puisque son image $x^L_{M'}$ est située à gauche de celle de M dans la vue de gauche, on en déduit que le segment $[C_{\mathrm{L}}M']$ se situe à gauche de la droite $(C_{\mathrm{L}}M)$. Puisque l'ordre est supposé préservé, on en déduit que l'image $x^R_{M'}$ du point M' se situe également à gauche de l'image x^R_M du point M dans la vue de droite. Par conséquent, le segment $[C_{\mathrm{R}}M']$ est situé à gauche de la droite $(C_{\mathrm{R}}M)$. Cette première étude permet de situer le point M' dans le secteur délimité à droite par les droites $(C_{\mathrm{L}}M)$ et $(C_{\mathrm{R}}M)$.

7. La préservation de l'ordre est appelée *contrainte de monotonie* dans [17].
8. On utilise ici la terminologie de [31], même si la contrainte se traduit dans des termes différents.

Posons $\varepsilon = x_M^L - x_{M'}^L$, dont la composante horizontale est positive. Pour que M' reste visible depuis la vue de droite, sa distance à la caméra, notée h' est majorée par celle de l'intersection des droites $(C_{\mathrm{R}}M)$ et $(C_{\mathrm{L}}M')$, que l'on note C. On en déduit que la disparité du point M' est minorée par celle du point C, s'il était visible. Calculons-la. Puisque C se projette dans l'image de droite sur le pixel x_M^R, et dans l'image de gauche sur le pixel $x_{M'}^L$, on en déduit que

$$d(C) = x_{M'}^L - x_M^R = x_{M'}^L - x_M^L + x_M^L - x_M^R = d(M) - \varepsilon.$$

Puisque $d(M) = d(x_M^L)$ et que $d(M') = d(x_{M'}^L) = d(x_M^L - \varepsilon)$, on en déduit que

$$d(x_M^L) - d(x_M^L - \varepsilon) \leq \varepsilon.$$

En faisant tendre ε vers 0, on en déduit que, pour ne pas créer de l'occultation, les variations horizontales de la disparité d ne doivent pas atteindre ou excéder 1.

2.2.4 Analyse de l'occultation : lien avec la disparité

Nous allons à présent établir le lien entre la largeur d'une occultation et le saut de disparité correspondant. Une étude similaire a été proposée dans [17].

On suppose dans cette analyse que l'objet, appelé objet *occultant*, occulte partiellement un objet situé derrière lui, désigné sous le nom d'objet *occulté*. L'objet occulté sera supposé de distance à la caméra constante, notée h', donc parallèle au plan image. Sa disparité est donc calculable, même dans la région occultée. Il sera modélisé par un plan. Puisque l'occultation se produit sur le bord gauche des objets, appelons A le bord gauche de l'objet occultant, visible depuis les deux vues. Son image par chacune des caméras est notée respectivement x_A^L et x_A^R.

Calculons la longueur maximale que peut atteindre l'occultation dans la vue de gauche. Considérons pour cela la figure 2.7, dans laquelle O_{L} et O_{R} désignent les projetés des centres optiques sur l'intersection entre le plan image et le plan épipolaire. Calculer la largeur de l'occultation dans la carte de disparité revient à calculer la longueur du segment $[x_A^L x_{A'_L}^L]$. Remarquons que, par construction, les points A et A'_R ont même image dans l'image de droite, que l'on note x_A^R. En particulier, on en déduit que la longueur $|x_A^L x_{A'_L}^L|$ peut s'exprimer en fonction du point O_L, car O_L n'appartient pas au segment $[x_A^L x_{A'_L}^L]$,

$$|x_A^L x_{A'_L}^L| = \Big| |O_L x_A^L| - |O_L x_{A'_L}^L| \Big|$$

où on peut faire apparaître la longueur $|O_R x_A^R|$:

$$|x_A^L x_{A'_L}^L| = \Big| (|O_R x_A^R| - |O_L x_{A'_L}^L|) - (|O_R x_A^R| - |O_L x_A^L|) \Big|.$$

On reconnaît alors les disparités respectives des pixels x_A^L et $x_{A'_R}^L$, ce qui assure finalement que

$$|x_A^L x_{A'_L}^L| = |d(x_{A'_L}^L) - d(x_A^L)|.$$

Autrement dit, la largeur de l'occultation vaut la différence entre la disparité de l'objet occulté et la disparité de l'objet occultant, que l'on désignera désormais sous le nom de *saut de disparité* autour de la région occultée. Ce saut est positif, car l'objet occulté se situe derrière l'objet occultant. Par ailleurs, on a montré que, dans la vue de référence, l'occultation était positionnée immédiatement à gauche de l'image de l'objet occultant.

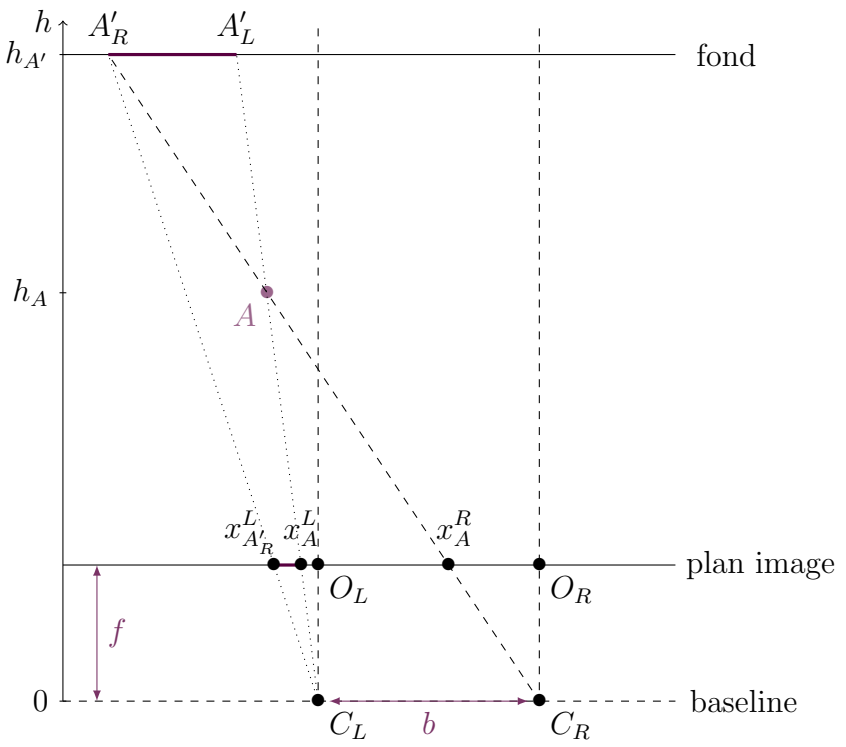


FIGURE 2.7 – Largeur de l'occlusion. La région du fond occultée par le point A est matérialisée par le segment $[A'_R A'_L]$. La largeur de l'occlusion dans la fonction de disparité est alors donnée par la longueur du segment $[x^L_A x^L_{A'_R}]$, où x^L_A (resp. $x^L_{A'_R}$) désigne l'image du point A (resp. A'_R) dans l'image de gauche.

2.3 L'état de l'art

Pour une revue plus complète des méthodes de stéréovision, on pourra se reporter à [36, 41]. Les mesures de dissimilarité sont évaluées plus spécifiquement dans [21] par exemple.

2.3.1 Mesurer la similarité de deux pixels

Deux pixels homologues étant les images d'un même point physique, la mise en correspondance stéréoscopique repose essentiellement sur des critères de similarité entre deux pixels. L'idée sous-jacente est que, sous l'hypothèse d'une absence de changement d'illumination ou de contraste, plus deux pixels sont visuellement ressemblants, plus il y a de chance pour qu'ils soient issus du même point.

Cette approche nécessite donc de définir des *mesures de dissimilarité*[9], qui quantifient la ressemblance visuelle de deux pixels. Une mesure de dissimilarité est une fonction D à valeurs positives. Si on note I_{L} l'image de référence et I_{R} l'image de droite, alors, pour tout pixel p dans l'image de gauche et tout pixel q dans l'image de droite, la quantité $D_{I_{\mathrm{L}},I_{\mathrm{R}}}(p,q)$, appelée *coût de corrélation*, est d'autant plus faible que les deux pixels p et q sont *semblables*.

Corrélation d'intensité Les mesures les plus faciles à définir sont les mesures de corrélation de niveaux de gris, car il s'agit de comparer les deux valeurs réelles que sont

9. Certains auteurs parlent de *mesure de corrélation*.

les intensités respectives de p dans I_{L} et de q dans I_{R}. Celles-ci sont notées respectivement $I_{\mathrm{L}}(p)$ et $I_{\mathrm{R}}(q)$. Toute distance sur $\mathbb{R}$ appliquée au couple $\big(I_{\mathrm{L}}(p),I_{\mathrm{R}}(q)\big)$ peut être utilisée. Les plus classiques [36] sont la distance qui découle de la norme euclidienne (aussi appelée mesure AD pour *absolute difference*) et le carré de celle-ci (appelée SD pour *squared difference*). Elles sont en effet simples à implémenter et peu coûteuses en calculs [18], tout en produisant des résultats raisonnables tant qu'il n'y a pas de changement d'illumination.

En réalité, les images numériques sont échantillonnées, ce qui signifie qu'un pixel n'est jamais l'image d'un point unique de la scène, mais plutôt une moyenne des images d'une petite région de l'espace. Ainsi, un point physique peut se projeter dans deux pixels qui, du fait de ce moyennage, apparaissent légèrement dissemblables, même en l'absence de bruit ou tout autre transformation chromatique. Pour réduire ce biais, Birchfield et Tomasi [3] propose d'utiliser une interpolation horizontale de l'image. Une variante exploitant l'information verticale est proposée dans [24]. L'interpolation choisie est l'interpolation bilinéaire, ce qui permet de rendre ce procédé peu coûteux en calculs.

Pour exploiter l'information supplémentaire contenue dans la couleur, une possibilité est de généraliser les mesures de corrélation de niveaux de gris à la couleur. Pour ce faire, [8] propose de combiner les coûts de corrélation obtenus pour chacun des canaux couleur, grâce à une fonction de *fusion*. Parmi les choix les plus classiques pour cette fonction, on peut citer la moyenne arithmétique, la valeur médiane, le minimum, le maximum, ainsi qu'une fonction proposée dans [2].

Exploitation des variations locales d'intensité La corrélation d'intensité est très sensible au bruit, ainsi qu'aux changements d'illumination et de contraste. Pour ajouter davantage d'informations, on peut exploiter les variations locales d'intensité.

Une première façon de procéder est d'appliquer sur les images une transformation, avant de les comparer avec une mesure de dissimilarité comme celles présentées au paragraphe précédent, si les images résultantes sont mono-valuées, ou avec des mesures plus adaptées dans les autres cas. Une transformation naturelle consiste à calculer le gradient des images[10]. Dans [35], une mesure de dissimilarité est ensuite définie pour comparer les vecteurs gradients en p et q. Cette mesure tient compte de la fiabilité de la comparaison, qui augmente avec la longueur des vecteurs comparés. Maar et Hildreth [26] proposent quant à eux d'appliquer le LoG (*laplacian of gaussian*) sur les images. Les mises en correspondance les plus précises concernent en effet les points où le gradient est le plus grand, d'où l'idée de localiser les maxima de la norme du gradient. Cela revient à localiser les zéros de la dérivée seconde. Comme celle-ci est sensible au bruit (hautes fréquences), un filtre passe-bas (convolution avec une gaussienne) est appliqué sur les images avant de calculer le laplacien. Zabih et Woodfill [45] définissent quant à eux deux transformées dites non paramétriques basées sur le *rang* du pixel. Dans une fenêtre autour de p (resp. de q), on identifie les pixels dont l'intensité est plus faible que celle de p (resp. de q). Puis, on peut soit les compter (*rank filter*), ce

10. L'œil humain interprète en réalité les couleurs à partir de leurs variations relatives plus qu'il ne les perçoit de manière absolue. Ainsi, la perception d'une couleur dépend fortement de son environnement. Cette observation a été énoncée dès 1839 par le chimiste Michel-Eugène Chevreul et est connue sous le nom de *loi du contraste simultané des couleurs*. Plus récemment, en 2015, une photographie de robe a fait le tour du monde et des réseaux sociaux car certaines personnes la voyaient blanche et dorée, d'autres la voyaient au contraire bleue et noire, tandis qu'une minorité de personnes pouvaient la voir des deux manières. Il s'agissait pourtant de la même image, mais dont les couleurs ont certainement été interprétées de manière différente par le cerveau.

qui revient à ordonner les pixels de la fenêtre suivant leur intensité, puis à déterminer le rang de p ; soit conserver la localisation de ces pixels, en la codant sous la forme d'un vecteur booléen (*census rank*). Dans le premier cas, on utilise sur les images transformées une mesure de dissimilarité pour image mono-valuée, dans le second cas, les vecteurs obtenus sont comparés grâce à la distance de Hamming (qui compte de nombre de coefficients différents). Ces deux transformations sont insensibles à tout changement de contraste.

Une seconde approche consiste à définir directement des mesures de dissimilarité comparant les variations d'intensité. Parmi ces approches, une méthode très populaire est le calcul de la NCC (*normalized cross correlation*) entre deux fenêtres, bien qu'elle soit plus coûteuse en calculs [18]. On considère deux fenêtres carrées autour des pixels p et q, que l'on normalise. Puis on en calcule le produit scalaire : plus celui-ci est grand, plus les deux voisinages sont semblables. Cette méthode gère les changements de contraste affine entre les deux images. On peut également mentionner l'utilisation de l'information mutuelle (MI) [43, 20] pour définir une mesure de corrélation. On considère que les deux images p et q (prises avec leur voisinage) sont des réalisations d'une certaine distribution. Dans ce cas, plus ces réalisations sont indépendantes, plus l'information mutuelle est faible.

Ces mesures, définies sur les images en niveaux de gris, sont plus robustes aux changements d'illumination et de contraste. Dans [4], les auteurs montrent qu'elles sont plus efficaces que les mesures de corrélation de couleur, et conseillent donc de les préférer à ces dernières.

Combinaison de l'intensité et des variations L'utilisation des variations est performante dans les régions texturées, mais dans les zones plates, en présence de bruit, elles peuvent se révéler désastreuses. L'idée est donc de combiner une corrélation d'intensité avec une corrélation basée sur les variations locales.

Une premier choix consiste à combiner la mesure AD et une mesure basée sur le gradient. Le gradient peut être utilisé dans son intégralité [23] ou seule sa composante horizontale est prise en compte [33]. La mesure résultante est une combinaison convexe (avec éventuellement un seuillage préalable des valeurs) des deux coûts de corrélation considérés.

Une autre méthode très plébiscitée est l'AD Census [28], qui combine la mesure AD et la corrélation obtenue après le filtrage Census. Le coût de dissimilarité est défini comme la somme des deux coûts initiaux, auxquels on a appliqué au préalable une certaine fonction croissante (avec des paramètres différents pour les deux coûts).

Ces méthodes ont montrées leur efficacité sur le banc d'essai Middlebury (voir paragraphe 2.3.4), d'où leur popularité. Elles donnent effectivement des résultats satisfaisants, tout en étant relativement peu coûteuses en calculs. Néanmoins, elles reposent sur un nombre de paramètres important (en général 3), qui sont difficiles à régler correctement.

2.3.2 Approches locales *versus* approches globales

Selon Scharstein et Szeliski [36], les méthodes de stéréovision peuvent être classées en deux grandes familles : les méthodes locales, qui mettent en correspondance les pixels à l'aide d'informations purement locales, et les méthodes globales, qui résolvent un problème d'optimisation global.

Méthodes locales : comparaison de fenêtres Le principe central est le suivant : pour tout pixel p de l'image de référence donné, le pixel homologue est le pixel q le plus ressemblant (dans l'intervalle de disparité) de l'image de droite. Cette ressemblance étant mesurée par la mesure de dissimilarité choisie, le pixel q devrait être le pixel minimisant le coût de corrélation $D_{I_\mathrm{L},I_\mathrm{R}}(p,q)$. Or, la corrélation pixellique (qui ne compare un pixel qu'avec un autre pixel) n'est pas suffisamment fiable (surtout en présence de bruit ou d'effet de Strobes par exemple). Cette remarque est déjà l'introduction des comparaisons basées sur les variations locales d'intensité.

En partant de l'hypothèse que la disparité est localement constante (ce qui est en réalité faux), on peut renforcer la fiabilité de ce critère en prenant également en compte le coût de corrélation $D_{I_\mathrm{L},I_\mathrm{R}}(p',q')$ des voisins p' (resp. q') du pixel p (resp. q), où $q-p$ et $q'-p'$ sont égaux (ce qui revient à tester la même valeur de disparité sur tout le voisinage). Ce procédé est appelé *agrégation des coûts*. L'idée sous-jacente est de comparer le voisinage de p avec le voisinage de q : on parle de *block-matching* (le *block* désignant le voisinage).

La manière la plus simple pour agréger les coûts de corrélation est de les moyenner dans un voisinage du pixel considéré. Si la mesure de dissimilarité choisie est AD et que la moyenne est une moyenne arithmétique, alors il s'agit de l'agrégation SAD (pour *sum of absolute differences*) et pour SD, on parle de SSD (pour *sum of squared differences*). En effet, si le voisinage considéré est le même pour tous les pixels, alors la moyenne arithmétique est équivalente à une somme.

Le choix de ce voisinage est crucial : les deux paramètres possibles en sont la taille et la forme (et éventuellement la position relative par rapport au pixel considéré).

La taille du voisinage est délicat. S'il est trop petit, alors il y a trop peu d'informations à exploiter et la corrélation reste trop incertaine. S'il est trop grand, alors l'hypothèse de disparité constante dans le voisinage devient fausse. Dans ce dernier cas, apparaît un phénomène dit d'*adhérence*. Au voisinage d'une discontinuité, entraînant nécessairement occultation ou désoccultation, c'est l'objet occultant qui va imposer sa disparité. Dans le cas d'images aériennes par exemple, cela se traduit par un épaississement des immeubles (c'est pourquoi on parle également de *fattening effect*). Pour une taille et une forme de voisinage données, une manière d'éviter l'adhérence est de décentrer le voisinage d'agrégation pour éviter d'y inclure une discontinuité. C'est ce que fait le *MinFilter* [16]. Une autre manière d'éviter cet écueil est d'utiliser des voisinages de tailles variables [1].

Les formes de voisinage les plus simples sont les fenêtres carrées, car faciles à implémenter. Néanmoins, des variantes ont été proposées depuis deux décennies. L'idée est que le meilleur voisinage est le voisinage le plus grand dans lequel la disparité reste constante. Le modèle de scène classiquement retenu étant que les objets sont en réalité des surfaces planes parallèles au plan image, il suffit de considérer comme voisinage l'objet auquel appartient le point considéré. Cela conduit à segmenter la scène. En l'absence de textures, l'intensité reste une méthode fiable pour segmenter une image, avec des méthodes comme le *Mean Shift* [14, 10]. Les segments obtenus sont alors utilisés comme voisinages [5]. Cette procédure reste coûteuse, c'est pourquoi [48] choisit de segmenter l'image en permettant à deux segments (initialisés par des pixels) de fusionner suivant des critères sur la taille des voisins et la proximité des couleurs. Une autre méthode populaire consiste à construire le voisinage en déployant une croix [46] (*cross-based regions*) autour du pixel, qui forme le squelette du voisinage. Ensuite, pour chaque pixel de la branche verticale, on agrandit le voisinage en déployant des branches horizontales de part et d'autre de la branche verticale. À nouveau, l'ajout d'un pixel

au voisinage est conditionné par son intensité.

Cependant, les méthodes les plus efficaces consistent à considérer des voisinages *non opaques*, c'est-à-dire où chaque pixel n'a pas le même poids dans l'agrégation. Cela revient à agréger les coûts en utilisant une moyenne pondérée. Ils sont généralement connus sous le nom de *fenêtres adaptatives*. L'une des méthodes les plus réputées et les plus efficaces est celle proposée par Yoon et Kweon [44, 13] où la pondération associée au voisin p' dépend à la fois de la distance dans l'espace des couleurs entre les intensités $I_{\mathrm{L}}(p)$ et $I_{\mathrm{L}}(p')$ et de la distance spatiale entre les deux pixels. Il s'agit en réalité du filtre bilatéral (voir [40] pour une revue plus détaillée sur les filtres bilatéraux). Une variante moins coûteuse en calculs [19] consiste à utiliser un filtre guidé [33, 40].

Après l'étape d'agrégation de coût, le pixel homologue retenu est celui qui minimise le coût de corrélation agrégé : cette étape est appelée WTA (*winner-take-all*).

Méthodes globales : minimisation d'une énergie Les méthodes globales choisissent d'exploiter la régularité de la scène. Pour ce faire, elles introduisent une fonctionnelle d'énergie qui pénalise la non-régularité de toute carte de disparité, tout en incitant l'algorithme à mettre en correspondance des pixels semblables. Un minimum de cette fonctionnelle est alors calculé, qui est la fonction satisfaisant au mieux les critères pénalisés. La difficulté réside principalement dans l'étape d'optimisation : les fonctionnelles d'énergie considérées ne sont généralement pas convexes, ce qui n'assure pas l'existence d'un minimum global. Par ailleurs, cela conduit à des algorithmes très coûteux en calculs. C'est pourquoi les modèles de régularité choisis dépendent essentiellement de leur compatibilité avec des algorithmes d'optimisation existants.

Les fonctionnelles d'énergie possèdent classiquement plusieurs termes, chaque terme étant dédié à une propriété particulière recherchée pour la carte de disparité. Un premier terme est le terme d'*attache aux données* ou de *fidélité*, qui mesure à quel point les pixels mis en correspondance sont semblables. Il est donc défini à l'aide de mesures de dissimilarité. Puisque l'estimation de la disparité ne repose plus uniquement sur la corrélation, mais est renforcée par d'autres termes que nous allons présenter, il n'est plus nécessaire de choisir une mesure de dissimilarité très performance. C'est pourquoi ce terme est généralement défini à partir des mesures AD ou SD, qui sont les moins coûteuses en calculs. Un second terme classique est le terme de *régularité*. Comme son nom l'indique, il mesure la régularité de la carte de disparité, qui reflète celle de la scène, composée d'objets de surfaces généralement lisses par morceaux. Il est donc généralement défini sur les variations de la carte de disparité. Celles-ci peuvent être pénalisées dès qu'elles existent [24] ou la pénalisation peut dépendre de l'amplitude des variations : c'est le cas par exemple de la régularisation quadratique ou de la régularisation TV (variation totale) [32]. La régularisation TV présente l'avantage de mieux préserver les discontinuités, car elle conduit à des cartes constantes par morceaux, alors que la régularisation quadratique conduit à des cartes très lisses. Une variante plus régulière de TV (régularisation Huber, [32]) permet d'obtenir des cartes avec des discontinuités nettes, tout en autorisant un léger gradient. Le critère de régularisation peut également concerner les segments de l'image [48, 5, 23, 47]. Si l'on retrouve systématique les termes de fidélité et de régularité, certains auteurs ajoutent un ou plusieurs autres termes. Il est ainsi naturel d'introduire un terme forçant l'*injectivité* de la mise en correspondance (*uniqueness term*, [24]). Un terme d'*occultation* ([24]) permettant de tenir compte de ce phénomène peut aussi être ajouté. Enfin, Papadakis et Caselles [31] ont proposé un terme gérant les contraintes de *visibilité*.

Le choix de la fonctionnelle d'énergie est fortement lié à la méthode d'optimisation

utilisée pour la minimiser. Compte tenu de la taille du problème, celle-ci est choisie pour son efficacité. Les premières méthodes globales se sont donc appuyées sur les méthodes d'optimisation 1D basées sur la programmation dynamique. Pour pouvoir utiliser ce genre de méthodes, Bobick et Intille [6] choisissent de définir une fonctionnelle composée d'énergies indépendantes définies sur les lignes de la disparité. Ainsi, le problème se réécrit comme un ensemble de problèmes de dimension 1, qu'ils résolvent indépendemment. Pour incorporer une régularisation verticale tout en tirant parti de l'efficacité des méthodes 1D, Hirschmüller [20] propose d'alterner des minimisations dans différentes directions, tandis que Veksler [42] transforme le problème en un problème sur un arbre, sur lequel elle peut utiliser la programmation dynamique.

Les résultats les plus satisfaisants restent ceux obtenus avec une régularisation 2D. L'utilisation des *graph cuts* [24] permettent de donner une solution approchée en alternant les α-*expansion moves* ou les $\alpha\beta$-*swap* qui font décroître l'énergie. Les premiers consistent à agrandir à chaque itération l'ensemble de niveau α de la disparité (cf. chapitre 4) tandis que les seconds considèrent deux ensembles de niveaux α et β dont ils échangent les éléments. Les approches bayésiennes basées sur le *belief propagation* (BP) ont été également connu un succès important [39]. L'idée est de reformuler le problème avec des champs de Markov, où les différents termes de l'énergie se traduisent par des interactions entre les nœuds du réseau. Il s'agit ensuite de calculer le *maximum a priori* (MAP) en utilisant un algorithme de BP qui met à jour la disparité en faisant passer des messages à travers le réseau. Enfin, Pock et coll. [32] ont proposé une méthode de relaxation convexe de l'énergie, ce qui leur permet d'exploiter les outils d'optimisation convexe. Cette méthode, sur laquelle se base le chapitre 3, possède l'avantage de s'appliquer à une classe très large de fonctionnelles d'énergie. Enfin, on pourra citer les travaux de [29] et [9], qui exploitent également des outils d'optimisation convexe en considérant une version approchée mais convexe du problème initial non convexe qu'ils souhaitent résoudre, en utilisant une linéarisation d'une des images autour d'une première estimation de la disparité. Contrairement à la méthode de Pock et coll., cette approche ne constitue donc pas une relaxation convexe *exacte* du problème.

2.3.3 Occultation, correspondances non fiables

Du fait du phénomène d'occultation et des nombreuses difficultés soulevées dans le paragraphe 2.1.4 qui rendent la mise en correspondance difficile, la gestion des erreurs est un volet important de toute méthode de stéréovision. Il peut être utile de distinguer les erreurs dues à l'occultation des autres, qui peuvent résulter d'une multitude d'origines.

Gestion des occultations Dans les méthodes locales, les occultations sont généralement traitées comme des mises en correspondance non fiables, qui font l'objet du paragraphe suivant. Néanmoins, l'utilisation de mesures robustes aux occultations a été proposée dans [7].

Dans les méthodes globales, l'occultation peut être prise en compte grâce à l'introduction d'un terme dédié. Dans [6], les auteurs exploitent l'analyse de la section 2.2 pour classer les pixels en trois familles en exploitant les variations de chaque ligne de la disparité : ceux qui sont mis en correspondance (variations horizontales), ceux qui correspondant à des occultations (variations diagonales) et ceux qui correspondent à des désoccultations (variations verticales). Le désavantage majeur de cette méthode est qu'elle ne tient pas du tout compte de la régularité verticale de la scène car le problème

est traité indépendamment sur chaque ligne. Dans [24], le terme d'occultation compte le nombre de pixels non mis en correspondance avec un pixel homologue. Ce terme est pondéré par un paramètre choisi de sorte à contrôler le nombre de pixels occultés, ce qui implique de réussir à l'estimer de manière empirique ou heuristique.

Des approches plus complexes peuvent également être envisagées. Dans [38], on alterne estimation de la disparité sur les régions non occultées et estimation des régions occultées.

Détection et rejet des pixels non fiables Les cartes de disparité fournies par une méthode globale possèdent une cohérence globale grâce au terme de régularité et éventuellement au terme d'injectivité. Lorsque l'occultation est prise en compte, les zones occultées n'ont soit aucune disparité d'attribuée, soit celle-ci n'est pas significative. Dans les deux cas, leur localisation est connue et on peut facilement rejeter l'information dans ces régions.

Du fait de leur caractère local, les méthodes locales génèrent des cartes généralement moins fiables. Les erreurs sont principalement dues à l'occultation, à l'adhérence, à l'effet de Strobes ou au manque de textures. Contrairement aux méthodes globales, aucun critère de régularité ne permet de détecter ces erreurs. C'est pourquoi on définit des filtres de rejets que l'on applique une fois la carte de disparité estimée. Le filtre le plus populaire est le filtre LRRL (*left-right right-left*) [27, 33], qui traduit la contrainte d'injectivité. La carte de disparité sur la vue de référence et celle sur la vue de droite sont calculées. Ensuite, on évalue leur cohérence : tous les pixels de l'image de référence dont le pixel homologue dans la vue de droite n'est pas mis en correspondance avec lui dans l'estimation de la disparité de droite sont rejetés. Ce filtre nécessite de calculer deux cartes, donc de doubler le nombre d'opérations, mais ce n'est généralement pas un problème majeur car les méthodes locales sont peu coûteuses en calculs.

Un autre outil puissant est la validation *a contrario* [34]. Les méthodes *a contrario* reposent sur le principe d'Helmholtz qui assure que *dans le bruit, on ne voit rien.* Autrement dit, les structures détectables sont celles qui ont une faible de chance de se produire au hasard. Appliquée à la mise en correspondance stéréoscopique, ce principe permet de rejeter des mises en correspondance en mesurant la probabilité que la similarité entre les deux pixels (après agrégation des coûts) soit due au hasard. L'inconvénient majeur de ce filtre est qu'il rejette beaucoup de points.

Après l'application d'un ou de plusieurs de ces filtres ou l'extraction des zones d'occultation, on se retrouve avec des cartes de disparité incomplètes, dites *éparses* ou *non denses*. Pour obtenir une disparité définie partout, il faut ajouter une étape de *densification.*

Densification des cartes La densification des cartes de disparité consiste à compléter la carte de disparité où elle n'est pas connue. La stratégie choisie dépend de la raison pour laquelle le pixel a été rejeté. Dans [33], les auteurs considèrent que les pixels rejetés le sont principalement car ils sont occultés. Or, l'analyse menée dans 2.2 assure que, dans l'hypothèse la plus simple où les objets partiellement occultés ont une disparité constante (même au niveau de la partie occultée), la disparité de la région occultée est celle de l'objet occulté entier. Or, l'occultation étant située à gauche des discontinuités, la partie non occultée de l'objet occulté est située à gauche de la partie occultée. La densification se fait alors en diffusant la disparité vers la droite dans les régions occultées.

(a) Tsukuba (b) Venus (c) Teddy (d) Cones

FIGURE 2.8 – Vue de référence des quatre paires de la version 2 du banc d'essai Middlebury.

2.3.4 Le banc d'essai Middlebury

Suite à leur article de revue [36], SCHARSTEIN et SZELISKI ont mis en ligne le banc d'essai Middlebury [11]. Les chercheurs sont invités à tester leur algorithme sur les images proposées et à soumettre leurs résultats (mais pas leur algorithme). Leurs résultats sont classés suivant plusieurs critères (détaillés plus bas). La principale règle est que les paramètres ne doivent pas être adaptés manuellement à chaque scène.

Version 2 Pour la version 2 du banc d'essai, quatre paires sont proposées : Tsukuba, Venus [36], Teddy et Cones [37] (cf. figure 2.8). Les tailles des images varient entre 384×288 pour Tsukuba et 450×375 pour Teddy et Cones. Les caméras sont en mouvement fronto-parallèle. Les intervalles de disparité sont fournis : leur longueur varie entre 16 pixels (pour Tsukuba) et 60 pixels (pour Teddy et Cones). Les vérités-terrains sont disponibles pour les quatre paires.

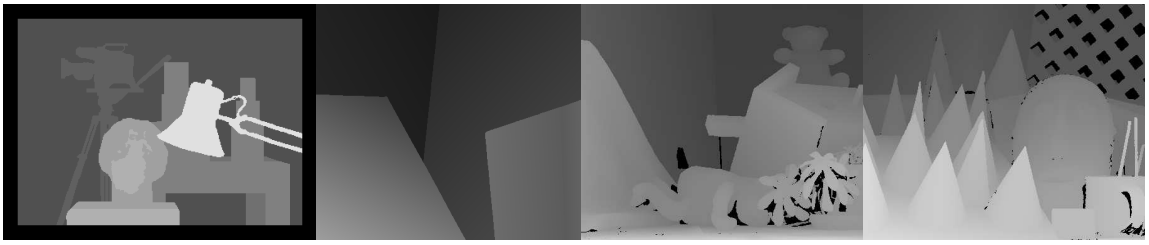
La version 2 n'est plus active depuis 2015. Elle a été remplacée par la version 3 [12] qui propose deux ensembles de paires : un ensemble avec vérité-terrains qui servent d'entraînement et un ensemble sans vérité-terrain pour des tests à l'aveugle. Les paires proposées dans la version 3 sont également beaucoup plus grandes (2880×1988 pour la paire Adirondack par exemple).

Vérité-terrain Les vérités-terrains (cf. figure 2.9) sont générées en projetant sur la scène des motifs réguliers [37] pour encoder chaque point de la scène, puis les disparités gauche et droite sont estimées pour N éclairages différents (ce qui permet en particulier de déplacer les ombres). Les cartes de disparité sont fusionnées et les zones d'occultation détectées. Un sous-échantillonnage des paires et de la carte de disparité finale est finalement effectué. La disparité de certains points de la scène reste inconnue.

Caractéristiques des scènes Ces scènes d'intérieur présentent peu d'ombre, peu de reflets (seule la paire Tsukuba en possède). Les images sont très texturées, les objets opaques. Il n'y a pas de changement d'illumination notable entre les deux vues.

La paire Tsukuba présente une vérité-terrain constante par morceaux et pixellique (alors que la lampe devrait logiquement être un peu bombée). La caméra et la lampe présentent des parties qui sont fines, mais seule la partie basse du fil de la lampe induit une inversion de l'ordre. La lampe et certaines parties métalliques de l'étagère au fond de la scène génèrent des reflets. La paire Venus est composée de panneaux texturés, non parallèles au plan image. Cela induit une vérité-terrain affine par morceaux. La

11. http ://vision.middlebury.edu/stereo
12. http ://vision.middlebury.edu/stereo/eval3



(a) Tsukuba (b) Venus (c) Teddy (d) Cones

FIGURE 2.9 – Vérité-terrain des quatre paires de la version 2 du banc d'essai Middlebury (vue de référence). Plus le pixel est clair, plus sa disparité est grande. En noir, les points dont la disparité n'est pas connue.

paire Teddy possède un intervalle de disparité très large (60 pixels), ce qui crée des zones d'occultation tout aussi larges. Par ailleurs, la texture du sol s'apparente à un bruit. La paire Cones possède également des zones d'occultation larges. De plus, les objets fins (pointes des cones, pinceaux) induisent une inversion de l'ordre.

Évaluation Pour chaque image, différents scores sont proposés : le pourcentage de disparité correcte dans l'image entière (*all*), près des zones de discontinuités de la scène (*nondisc*) et hors des régions occultées (*nonocc*). Pour cela, des masques sont proposés (cf. figure 2.11). Pour chaque masque, les erreurs sont mesurées à un seuil près (égal à 2, 1,5, 1, 0,75 ou 0,5 pixels). Pour chaque paire et chaque masque, le pourcentage d'erreur et le classement sont donnés. Par défaut, l'affichage correspond au classement moyen sur tous les masques pour un seuil de 1 pixel, mais le pourcentage moyen d'erreurs sur toutes les paires est également visible. Dans la figure 2.10 qui présente la tête du classement à la date de clôture de la version 2, on voit en particulier que la méthode PM-Forest est la meilleure pour le pourcentage moyen d'erreur (avec 2,64% d'erreur) mais n'arrive que neuvième au classement général.

2.3.5 Démos IPOL

La recherche reproductible En sciences, la reproductibilité d'une expérience permet de garantir sa pertinence et de valider les conclusions qui en découlent. Elle implique que tout résultat publié doit pouvoir être obtenu de manière identique si les conditions de l'expérience sont reproduites. Appliquée au domaine du traitement de l'image, la reproductibilité suppose que si l'on implémente la méthode décrite par un article, et qu'on l'applique aux mêmes données que celles testées par les auteurs, alors le résultat sera analogue. Si en théorie, cette condition semble facile à satisfaire (contrairement aux sciences pratiques, le nombre de paramètres influant l'expérience est limité, car un code informatique réagit systématiquement de la même manière), en pratique, elle implique d'avoir à disposition une implémentation analogue et les mêmes jeux de données. Or, ces deux éléments sont rarement disponibles.

Lorsqu'une méthode est publiée, l'article présente généralement une description (plus ou moins détaillée) de l'algorithme, ainsi qu'un pseudo-code présentant l'architecture du code. Malheureusement, ces informations sont loin d'être suffisantes pour réimplémenter l'algorithme proposé. Principalement à cause du nombre limité de pages, des éléments essentiels sont manquants : la valeur des paramètres sont rarement précisés (ou il en manque certains), d'éventuelles étapes de pré-traitement des données

Stereo Evaluation • Datasets • Code • Submit

Middlebury Stereo Evaluation - Version 2

Version 2 is no longer active. Please use the **Stereo Evaluation Version 3**

New features and main differences to version 1.

Open a new window for each link

Error Threshold = 1 (Error Threshold...) — Sort by nonocc ▼ — Sort by all ▼ — Sort by disc ▼

Algorithm	Avg. Rank ▼	Tsukuba ground truth nonocc ▼	Tsukuba all	Tsukuba disc	Venus ground truth nonocc	Venus all ▼	Venus disc	Teddy ground truth nonocc	Teddy all ▼	Teddy disc	Cones ground truth nonocc	Cones all ▼	Cones disc	Average percent of bad pixels (explanation)
IGSM [155]	10.4	0.93 10	1.37 12	5.05 12	0.07 2	0.17 5	1.04 2	4.08 20	5.98 10	11.4 21	2.14 9	6.97 14	6.27 8	3.79
TSGO [141]	13.4	0.87 4	**1.13** 1	4.66 6	0.11 10	0.24 14	1.47 13	5.61 46	8.09 21	13.8 39	1.67 2	6.16 3	4.95 2	4.06
JSOSP+GCP [149]	14.8	**0.74** 1	1.34 9	**3.98** 1	0.08 4	**0.16** 1	1.15 4	3.96 18	10.1 40	11.8 22	2.28 19	7.91 37	6.74 22	4.18
KADI [164]	15.2	1.02 16	1.23 4	5.51 17	0.08 3	0.20 8	1.11 3	5.16 37	9.43 35	13.0 33	2.07 4	7.16 19	5.97 4	4.33
SSCBP [157]	17.6	1.05 19	1.39 14	5.57 19	0.10 7	0.16 2	1.39 10	3.44 14	8.32 26	9.95 15	2.60 34	7.13 18	7.23 33	4.03
ADCensus [82]	18.2	1.07 23	1.48 21	5.73 26	0.09 5	0.25 18	1.15 4	4.10 21	6.22 11	10.9 18	2.42 25	7.25 21	6.95 26	3.97
AdaptingBP [16]	22.2	1.11 26	1.37 11	5.79 28	0.10 8	0.21 13	1.44 12	4.22 23	7.06 19	11.8 23	2.48 29	7.92 39	7.32 36	4.23
CoopRegion [39]	22.2	0.87 6	1.16 2	4.61 5	0.11 9	0.21 10	1.54 17	5.16 38	8.31 25	13.0 31	2.79 48	7.18 20	8.01 56	4.41
CCRADAR [150]	26.8	1.15 28	1.42 18	6.23 41	0.15 22	0.27 21	1.89 27	5.39 41	10.6 45	14.7 50	2.01 3	7.37 23	5.88 3	4.75
PM-Forest [162]	27.2	1.63 78	2.17 79	8.71 97	0.15 24	0.19 7	2.13 36	**1.91** 1	**2.29** 1	**5.47** 1	**1.32** 1	**2.02** 1	**3.69** 1	2.64
RDP [87]	28.7	0.97 12	1.39 15	5.00 11	0.21 46	0.38 38	1.89 27	4.84 29	9.94 39	12.6 28	2.53 33	7.69 29	7.38 37	4.57
MultiRBF [129]	28.7	1.33 53	1.56 27	6.02 37	0.13 15	0.17 4	1.84 24	5.09 35	6.36 12	13.4 37	2.90 58	6.76 11	7.10 31	4.39
DoubleBP [34]	29.0	0.88 8	1.29 7	4.76 9	0.13 16	0.45 56	1.87 26	3.53 17	8.30 24	9.63 11	2.90 57	8.78 69	7.79 48	4.19
OutlierConf [40]	30.0	0.88 7	1.43 19	4.74 8	0.18 35	0.26 20	2.40 45	5.01 31	9.12 33	12.8 30	2.78 47	8.57 58	6.99 27	4.60
SegAggr [144]	30.2	1.99 99	2.39 89	8.59 96	0.12 11	0.21 12	1.68 19	2.19 3	3.73 3	7.02 3	2.16 11	6.52 6	6.37 11	3.58
CVW-RM [146]	30.4	1.12 27	1.42 17	5.99 36	0.16 30	0.36 36	1.40 11	4.70 28	6.94 17	12.1 24	2.96 63	7.71 31	7.72 45	4.38
GC+LocalExp [158]	32.0	1.48 68	1.88 62	6.95 66	0.13 14	0.25 17	1.52 16	3.33 12	4.88 5	8.87 7	2.72 41	7.42 24	7.94 52	3.95
SOS [135]	35.0	1.45 64	1.63 33	7.83 84	0.21 44	0.32 29	2.29 44	3.13 11	8.45 28	9.74 12	2.43 26	7.10 17	7.02 28	4.30
SubPixSearch [109]	35.4	2.04 103	2.48 93	6.40 47	0.14 20	0.40 45	1.74 21	4.00 19	6.39 13	11.0 19	2.24 16	6.87 13	6.50 16	4.18

FIGURE 2.10 – Capture d'écran de la page de la version 2 du banc d'essai Middlebury (prise après sa désactivation). Les différents algorithmes sont classés suivant leur performance sur les quatre paires.

(a) Vérité-terrain (b) *all* (c) *disc* (d) *nonocc*

FIGURE 2.11 – Les masques utilisés par Middlebury pour évaluer les résultats soumis (exemple de la paire Tsukuba). En noir, les pixels dont la disparité n'est pas connue. En blanc, les points sur lesquels l'estimation de la disparité est évaluée : (b) partout où la vérité-terrain est connue ; (c) près des discontinuités de la scène ; (d) dans les régions non occultées.

ou post-traitement des résultats sont passées sous silence. Pourtant, ces détails jugés «techniques» sont nécessaires pour reproduire les résultats publiés.

Pour pallier ce manque, qui discrédite à terme la communauté scientifique, des plateformes ont vu le jour pour permettre aux auteurs de publier leurs codes. En 2011, le CNRS, HEC Paris et l'université d'Orléans lancent RunMyCode [13], qui permet d'associer à tout article de recherche (de tout domaine) une page web «compagnon» sur laquelle les auteurs postent leur code et les données sur lesquelles ils ont testé leur méthode. Le lien de cette page est alors cité dans l'article. L'intérêt est assez évident : la recherche redevient reproductible, puisque n'importe qui peut regénérer les résultats présentés par l'article. Il n'y a pas de paramètre ou d'étape cachée. On n'a pas besoin de réimplémenter la méthode, ce qui permet de gagner du temps. Enfin, on peut tester l'algorithme sur d'autres images que celles (forcément en nombre restreint) proposées par les auteurs.

Néanmoins, contrairement au contenu de l'article qui est expertisé par un comité de lecture, les codes publiés ne font l'objet d'aucune validation scientifique. En particulier, on n'y contrôle pas que le code correspond à l'algorithme décrit et que les résultats présentés sont bien obtenus grâce à ce code.

IPOL Le journal en ligne IPOL [14] (*Image Processing On Line*) est un journal lancé en 2009, à l'initiative de Nicolas Limare, Jean-Michel Morel et l'équipe Traitement d'Images et du Signal du CMLA (Centre de Mathématiques et leurs Applications), à l'ENS Cachan. Son objectif est de proposer des articles de traitement d'images, accompagnés d'un code qui est soumis à une expertise approfondie, ce qui permet de combler les limites de la plateforme RunMyCode.

Toute publication d'IPOL comporte trois composantes :

1. l'article à proprement parler ;
2. l'implémentation de l'algorithme présenté en ANSI C/C++ ou en Matlab (depuis mai 2015) ;
3. la partie *démo*, qui permet à l'utilisateur de faire tourner le code en ligne sur des images proposées par IPOL ou sur ses propres images.

L'article, qui n'est pas limité en nombre de pages, présente une méthode intéressante (non nécessairement originale). L'algorithme doit être décrit de manière exhaustive, en particulier les paramètres. Idéalement, cette description seule doit permettre à tout lecteur d'implémenter sa propre version de la méthode. Enfin, une analyse critique des résultats clôt l'article : les cas satisfaisants sont présentés, aussi bien que les mauvais, pour démontrer les apports et les limites de la méthode.

Un code (en ANSI C/C++ ou en Matlab) est également soumis, qui sera expertisé au même titre que l'article. Il doit être portable, c'est-à-dire pouvoir tourner sur toute machine standard (sous Windows, MacOS ou Linux). Il doit être suffisamment documenté pour permettre à tout lecteur de le comprendre. Enfin, puisqu'il a vocation à être diffusé et la démo maintenue par l'équipe IPOL, il doit être publié sous une licence de logiciel libre. Pour assurer sa diffusion la plus large possible, il doit utiliser des bibliothèques standards et stables. Les codes Matlab sont autorisés depuis 2015 car leur implémentation et leur utilisation sont plus souples que les codes ANSI C/C++. C'est par ailleurs l'un des langages les plus utilisés dans la communauté image.

13. http ://www.runmycode.org
14. http ://www.ipol.im

La partie *démo* constitue le dernier volet d'une publication IPOL. Elle se présente sous la forme d'une page dédiée, sur laquelle le *même* code que celui qui est publié peut être testé en ligne, sur les serveurs d'IPOL. Cela introduit une contrainte sur l'efficacité des méthodes publiées, qui doivent produire un résultat en moins de 30 secondes, quitte à restreindre la taille des données en entrée. Les implémentations exploitant le calcul parallèle sont donc encouragées. Le cas échéant, l'utilisateur peut changer les paramètres de la méthode, à l'aide de curseurs. Les algorithmes peuvent être testés sur les images proposées par les auteurs, mais également sur les images que l'utilisateur charge lui-même. Les résultats sur les images personnelles sont archivés et consultables en ligne (à moins que l'utilisateur ne l'interdise). Cette disposition permet de constituer une base de données plus importante qui montre comment se comporte l'algorithme sur des images variées.

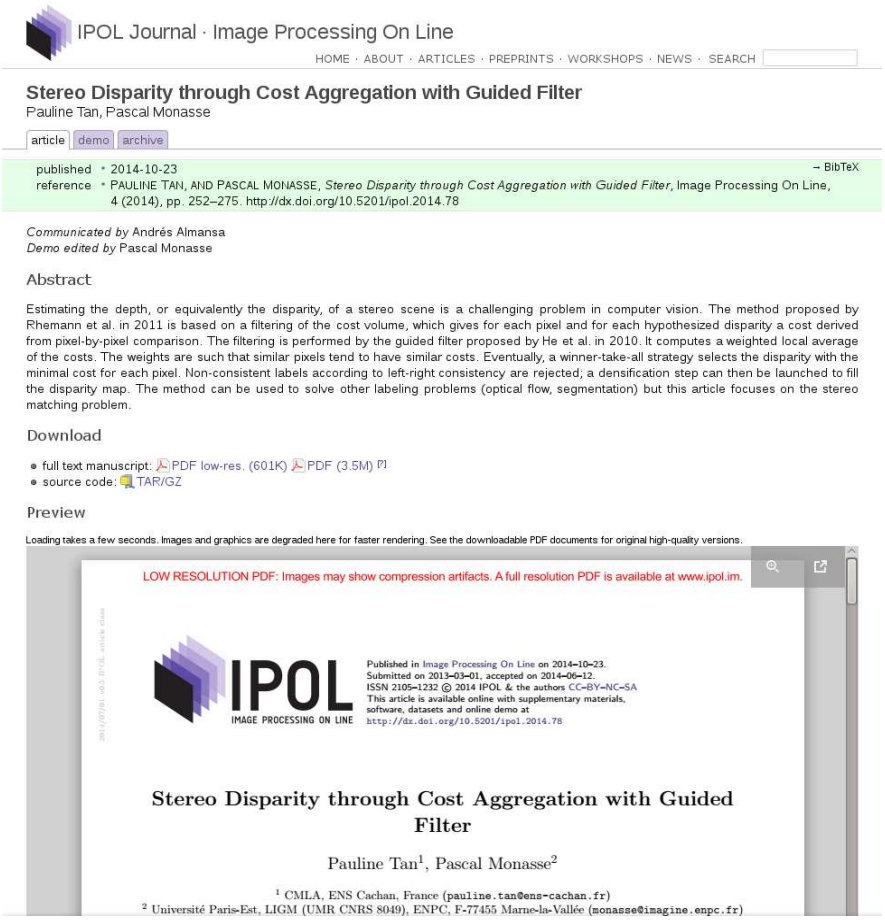
La politique très exigeante d'IPOL garantit une réelle reproductibilité des expériences publiées. Néanmoins, pour les auteurs, elle se traduit par une charge de travail plus lourde. Le code doit être lisible, documenté et maintenu. Ils doivent s'assurer de sa portabilité sur différentes plateformes. Or, ce travail est peu gratifiant, car généralement peu reconnu.

Le journal IPOL souhaite proposer dans chaque domaine du traitement d'images un maximum de méthodes constituant l'état de l'art. Actuellement (en 2016), six algorithmes ont été publiés dans la section stéréovision.

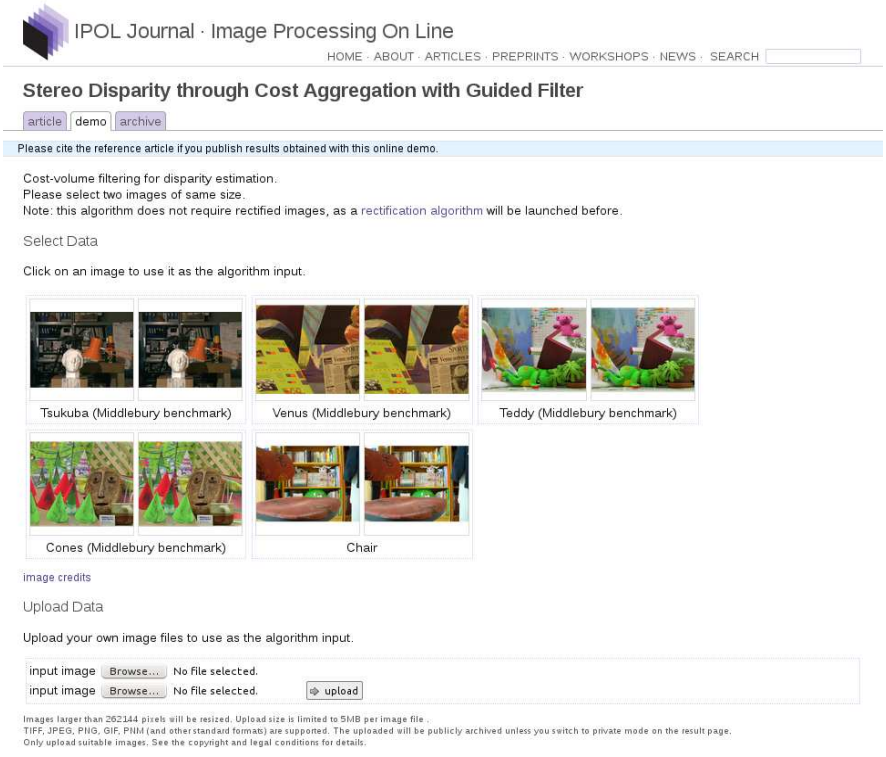
Algorithme de rectification épipolaire Un algorithme de rectification épipolaire [30] a été publié en 2011. Il estime deux homographies qui permettent de simuler deux vues en déplacement fronto-parallèle à partir d'une paire stéréoscopique quelconque. Elle est basée sur la méthode de Fusiello et Irsara [15], qui suppose que les deux caméras initiales sont parfaites (le point principal coïncide avec le centre du cadre) mais de (même) distance focale inconnue. La rectification est réalisée en minimisant le mouvement vertical de certains points mis en correspondance, sélectionnés par la méthode SIFT [25].

Algorithmes de mise en correspondance stéréoscopique Quatre articles ont été publiés sur IPOL à propos de la mise en correspondance stéréoscopique proprement dite. La première, publiée en 2014, est pour l'instant la seule méthode globale disponible sur IPOL. Elle est basée sur la méthode proposée en 2001 par Kolmogorov et Zabih [24]. Le code proposé est une variante du code de Kolmogorov (disponible sur sa page personnelle), plus adaptée aux standards d'IPOL. Malgré l'efficacité de l'implémentation des *graph cuts*, il a fallu découper les paires en bandes horizontales (avec recouvrement), afin de les traiter en parallèle, pour atteindre le temps d'exécution imposé par IPOL.

Les trois autres articles sont des méthodes locales. Le premier [12] présente une méthode qui permet d'agréger efficacement les coûts de corrélation, en s'appuyant sur une table de sommation, proposée par [11]. Ce même algorithme est à l'origine de la méthode présentée par [40], qui décrit l'algorithme initialement publié dans [33]. Il s'agit d'une méthode basée sur une implémentation efficace d'un filtre bilatéral. Enfin, l'article [13] propose une implémentation des célèbres fenêtres adaptatives (qui repose également sur un filtre bilatéral) de Yoon et Kweon [44], déjà évoquées plus haut dans ce chapitre.



(a)



(b)

FIGURE 2.12 – Les onglets d'une publication IPOL (captures d'écran), exemple de [40] : (a) l'article et le lien de téléchargement du code ; (b) la démo.

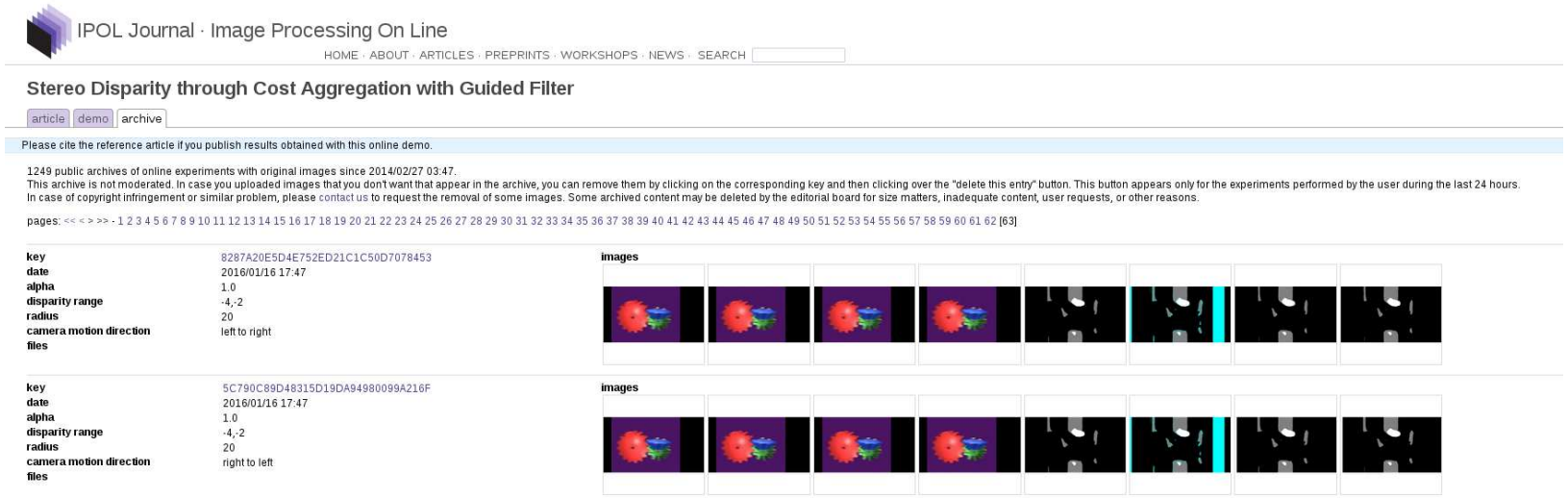


FIGURE 2.13 – Les onglets (suite et fin) d'une publication IPOL (captures d'écran), exemple de [40] : l'archive.

Références

[1] Satyajit Anil Adhyapak, Nasser Kehtarnavaz, and Mihai Nadin. Stereo matching via selective multiple windows. *Journal of Electronic Imaging*, 16(1) :013012, 2007.

[2] Thomas Belli, Matthieu Cord, and Sylvie Philipp-Foliguet. Colour contribution for stereo image matching. In *International Conference on Color in Graphics and Image Processing*, pages 317–322. Citeseer, 2000.

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

[4] Michael Bleyer and Sylvie Chambon. Does color really help in dense stereo matching ? In *International Symposium 3D Data Processing, Visualization and Transmission 2010*, pages 1–8, 2010.

[5] Michael Bleyer and Margrit Gelautz. A layered stereo algorithm using image segmentation and global visibility constraints. In *IEEE International Conference on Image Processing*, volume 5, pages 2997–3000. IEEE, 2004.

[6] Aaron F. Bobick and Stephen S. Intille. Large occlusion stereo. *International Journal of Computer Vision*, 33(3) :181–200, 1999.

[7] Sylvie Chambon. *Mise en correspondance stéréoscopique d'images couleur en présence d'occultations*. PhD thesis, Université Paul Sabatier-Toulouse III, 2005.

[8] Sylvie Chambon and Alain Crouzil. Color stereo matching using correlation measures. *Complex Systems Intelligence and Modern Technological Applications*, pages 520–525, 2004.

[9] Caroline Chaux, Mireille El-Gheche, Joumana Farah, Jean-Christophe Pesquet, and Béatrice Pesquet-Popescu. A parallel proximal splitting method for disparity estimation from multicomponent images under illumination variation. *Journal of mathematical imaging and vision*, 47(3) :167–178, 2013.

[10] Dorin Comaniciu and Peter Meer. Mean shift : A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5) :603–619, 2002.

[11] Franklin C. Crow. Summed-area tables for texture mapping. *SIGGRAPH Computer Graphics*, 18(3) :207–212, 1984.

[12] Gabriele Facciolo, Nicolas Limare, and Enric Meinhardt-Llopis. Integral images for block matching. *Image Processing On Line*, 4 :344–369, 2014.

[13] Laura Fernández Julià and Pascal Monasse. Bilaterally weighted patches for disparity map computation. *Image Processing On Line*, 5 :73–89, 2015.

[14] Keinosuke Fukunaga and Larry D. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, 21(1) :32–40, 1975.

[15] Andrea Fusiello and Luca Irsara. Quasi-euclidean uncalibrated epipolar rectification. In *IEEE International Conference on Pattern Recognition*, pages 1–4. IEEE, 2008.

[16] Andrea Fusiello, Vito Roberto, and Emanuele Trucco. Symmetric stereo with multiple windowing. *International Journal of Pattern Recognition and Artificial Intelligence*, 14(08) :1053–1066, 2000.

[17] Davi Geiger, Bruce Ladendorf, and Alan Yuille. Occlusions and binocular stereo. *International Journal of Computer Vision*, 14(3) :211–226, 1995.

[18] Marsha J. Hannah. Computer matching of areas in stereo images. Technical report, DTIC Document, 1974.

[19] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(6) :1397–1409, 2013.

[20] Heiko Hirschmüller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2) :328–341, 2008.

[21] Heiko Hirschmüller and Daniel Scharstein. Evaluation of cost functions for stereo matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[22] Ian P. Howard and Brian J. Rogers. *Binocular vision and stereopsis*. Oxford University Press, 1995.

[23] Andreas Klaus, Mario Sormann, and Konrad Karner. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In *IEEE International Conference on Pattern Recognition*, volume 3, pages 15–18. IEEE, 2006.

[24] Vladimir Kolmogorov and Ramin Zabih. Computing visual correspondence with occlusions using graph cuts. In *IEEE International Conference on Computer Vision*, volume 2, pages 508–515. IEEE, 2001.

[25] David G. Lowe. Object recognition from local scale-invariant features. In *IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157. IEEE, 1999.

[26] David Marr and Ellen Hildreth. Theory of edge detection. *Proceedings of the Royal Society of London B : Biological Sciences*, 207(1167) :187–217, 1980.

[27] Stefano Mattoccia, Federico Tombari, and Luigi Di Stefano. Stereo vision enabling precise border localization within a scanline optimization framework. In *Asian Conference on Computer Vision*, pages 517–527. Springer, 2007.

[28] Xing Mei, Xun Sun, Mingcai Zhou, Shaohui Jiao, Haitao Wang, and Xiaopeng Zhang. On building an accurate stereo matching system on graphics hardware. In *IEEE International Conference on Computer Vision Workshops*, pages 467–474. IEEE, 2011.

[29] Wided Miled, Jean-Christophe Pesquet, and Michel Parent. A convex optimization approach for depth estimation under illumination variation. *IEEE Transactions on Image Processing*, 18(4) :813–830, 2009.

[30] Pascal Monasse. Quasi-euclidean epipolar rectification. *Image Processing On Line*, 1, 2011.

[31] Nicolas Papadakis and Vicent Caselles. Multi-label depth estimation for graph cuts stereo problems. *Journal of Mathematical Imaging and Vision*, 38(1) :70–82, 2010.

[32] Thomas Pock, Daniel Cremers, Horst Bischof, and Antonin Chambolle. Global solutions of variational models with convex regularization. *SIAM Journal on Imaging Sciences*, 3(4) :1122–1145, 2010.

[33] Christoph Rhemann, Asmaa Hosni, Michael Bleyer, Carsten Rother, and Margrit Gelautz. Fast cost-volume filtering for visual correspondence and beyond. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3017–3024. IEEE, 2011.

[34] Neus Sabater. *Fiabilité et précision en stéréoscopie : application à l'imagerie aérienne et satellitaire à haute résolution.* PhD thesis, École normale supérieure de Cachan, 2009.

[35] Daniel Scharstein. Matching images by comparing their gradient fields. In *IAPR International Conference on Pattern Recognition*, volume 1, pages 572–575. IEEE, 1994.

[36] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1-3) :7–42, 2002.

[37] Daniel Scharstein and Richard Szeliski. High-accuracy stereo depth maps using structured light. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages I–195. IEEE, 2003.

[38] Jian Sun, Yin Li, Sing Bing Kang, and Heung-Yeung Shum. Symmetric stereo matching for occlusion handling. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 399–406. IEEE, 2005.

[39] Jian Sun, Nan-Ning Zheng, and Heung-Yeung Shum. Stereo matching using belief propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7) :787–800, 2003.

[40] Pauline Tan and Pascal Monasse. Stereo disparity through cost aggregation with guided filter. *Image Processing On Line*, pages 252–275, 2014.

[41] Federico Tombari, Stefano Mattoccia, Luigi Di Stefano, and Elisa Addimanda. Classification and evaluation of cost aggregation methods for stereo correspondence. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[42] Olga Veksler. Stereo correspondence by dynamic programming on a tree. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 384–390. IEEE, 2005.

[43] Paul Viola and William M. Wells III. Alignment by maximization of mutual information. *International Journal of Computer Vision*, 24(2) :137–154, 1997.

[44] Kuk-Jin Yoon and In So Kweon. Adaptive support-weight approach for correspondence search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (4) :650–656, 2006.

[45] Ramin Zabih and John Woodfill. Non-parametric local transforms for computing visual correspondence. In *European Conference on Computer Vision*, pages 151–158. Springer, 1994.

[46] Ke Zhang, Jiangbo Lu, and Gauthier Lafruit. Cross-based local stereo matching using orthogonal integral images. *IEEE Transactions on Circuits and Systems for Video Technology*, 19(7) :1073–1079, 2009.

[47] C. Lawrence Zitnick and Sing Bing Kang. Stereo for image-based rendering using image over-segmentation. *International Journal of Computer Vision*, 75(1) :49–65, 2007.

[48] C. Lawrence Zitnick, Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski. High-quality video view interpolation using a layered representation. In *ACM Transactions on Graphics*, volume 23, pages 600–608. ACM, 2004.