

gerer les occultations par m'ethode variationnelle

| | |
|---|------------|
| Introduction | 69 |
| 3.1 Fonctionnelle d'énergie | 71 |
| 3.1.1 Une corrélation adaptative pour limiter l'adhérence | 72 |
| 3.1.2 Une contrainte de pente pour gérer l'occultation | 73 |
| 3.2 Relaxation convexe du problème initial | 74 |
| 3.2.1 Énergie d'interface | 75 |
| 3.2.2 Relaxation convexe | 76 |
| 3.2.3 Formulation primale-duale | 77 |
| 3.3 Résolution numérique par algorithme primal-dual | 77 |
| 3.3.1 Discrétisation du problème | 77 |
| 3.3.2 Algorithme primal-dual | 80 |
| 3.3.3 Accélération par convexification | 82 |
| 3.4 Détection et gestion de l'occultation | 84 |
| 3.4.1 Détection de l'occultation par saturation de la pente | 85 |
| 3.4.2 Densification des zones occultées | 86 |
| 3.5 Résultats expérimentaux | 87 |
| 3.5.1 Modification des cartes d'occultation | 87 |
| 3.5.2 Récapitulatif de l'algorithme | 89 |
| 3.5.3 Première carte de disparité | 89 |
| 3.5.4 Traitement des occultations | 92 |
| 3.6 Discussion | 102 |
| 3.6.1 Résultats | 102 |
| 3.6.2 Différences entre les deux algorithmes proposés | 103 |
| 3.6.3 Choix des paramètres | 104 |
| 3.6.4 Comparaison avec d'autres algorithmes | 108 |
| Conclusion | 110 |

Introduction

En stéréovision binoculaire, les méthodes globales traduisent le problème de mise en correspondance par une fonctionnelle d'énergie. Celle-ci encode les propriétés de la

carte de disparité recherchée, en pénalisant tout écart à ces conditions. Cette approche conduit à optimiser des problèmes de grande taille, ce qui est très coûteux en calculs. Le choix de la méthode d’optimisation est donc crucial. C’est pourquoi il est généralement prioritaire sur le choix de la fonctionnelle d’énergie. Toutes les fonctionnelles ne sont en effet pas minimisables par une méthode donnée. C’est donc *le choix de la méthode d’optimisation qui conditionne la forme de la fonctionnelle* et, de fait, le choix du modèle de scène sous-jacent. Toute méthode globale peut être évaluée sur ces deux points : la pertinence du modèle considéré et l’efficacité algorithmique de la minimisation.

Une méthode comme celle basée sur l’utilisation des *graph cuts* de KOLMOGOROV et ZABIH [10], que nous étudierons en détails dans le chapitre suivant, présente une bonne efficacité algorithmique. En revanche, le modèle de régularité est biaisé par l’utilisation des graphes et la minimisation n’est pas exacte, mais seulement approchée. Néanmoins, c’est l’une des rares méthodes à gérer de manière raisonnable les occultations de la scène, en proposant à un pixel soit d’être mis en correspondance, soit d’être occulté. Toutefois, ce choix n’est basé que sur le coût de corrélation, et ne repose donc sur aucun modèle de scène. La méthode de programmation dynamique proposée par BOBICK et INTILLE [1] repose quant à elle sur l’étude menée au chapitre précédent sur l’occultation. Lorsque l’ordre est préservé dans les deux images, on a montré que la pente horizontale de la disparité est strictement inférieure à 1, et que la largeur des occultations valait exactement le saut de disparité autour de l’occultation. Leur méthode repose sur ce modèle de régularité de la scène, dont on a montré la pertinence. Malheureusement, la méthode d’optimisation utilisée, qui par ailleurs est très efficace, n’est valable qu’en dimension 1, et leur méthode ne peut donc pas intégrer la régularité verticale. On voit ainsi que, dans ces deux exemples de méthodes gérant l’occultation, le modèle de régularité est défaillant, car il doit s’adapter à la méthode d’optimisation choisie.

Dans leur article de 2010, POCK et coll. proposent une méthode qui permet de transformer une classe très large de fonctionnelles en un problème convexe [12], par relaxation. La résolution de ce dernier problème fait alors appel à l’optimisation convexe, pour laquelle on dispose de davantage d’outils efficaces, du fait de certaines propriétés des fonctions convexes (dont en particulier l’existence d’un minimum). Parmi les fonctionnelles admissibles, se trouvent celles possédant un terme d’attache aux données quelconque (mais continu) et un terme de régularité convexe. Leur méthode est donc relativement peu contraignante pour la forme de la fonctionnelle, mais ne peut pas gérer un terme d’occultation comme celui que proposent KOLMOGOROV et ZABIH [10]. D’où l’idée de reprendre le modèle de scène utilisé par BOBICK et INTILLE [1] et d’y ajouter une régularité verticale.

Dans ce chapitre, nous proposons donc une nouvelle fonctionnelle d’énergie (section 3.1), qui est à la fois admissible par la méthode de relaxation convexe [12] et qui tienne compte du phénomène d’occultation de la même manière que dans [1], c’est-à-dire en intégrant le modèle de scène détaillé dans le chapitre précédent (paragraphe 3.1.2). Nous introduirons également un nouveau terme d’attache aux données adaptatif (paragraphe 3.1.1) qui a pour but de limiter l’effet d’adhérence tout en bénéficiant de l’efficacité de la corrélation combinée de couleur et de gradient. Pour relaxer ce problème de manière convexe (section 3.2), nous reprenons une à une chaque étape de l’article [12]. Le problème convexe résultant peut s’écrire de manière primale-duale (paragraphe 3.2.3), et nous proposerons deux méthodes de résolution numérique (section 5.2). Puisque la fonctionnelle choisie permet de détecter les occultations dans la carte de disparité obtenue par relaxation convexe, une étape de densification est proposée (section 3.4). Nous présenterons enfin les résultats obtenus avec cette mé-

thode (section 3.5) : elle sera testée sur les images du banc d'essai Middlebury, et nous comparerons la détection des occultations avec la méthode des *graph cuts* de [10].

3.1 Fonctionnelle d'énergie

On rappelle que le domaine de image de référence est rectangulaire, noté $\Omega \subset \mathbb{R}^2$. La paire stéréoscopique considérée est donnée par (I_L, I_R) : ce sont des images en couleur RGB, dont les valeurs sur chaque canal sont comprises entre 0 et 255. Les caméras associées sont supposées en mouvement fronto-parallèle. La disparité u est donc une fonction définie sur Ω , à valeurs dans l'intervalle de disparité $I_{\text{disp}} \subset \mathbb{R}$, connu. On supposera dans un premier temps que u appartient à l'espace $W^{1,2}(\Omega; I_{\text{disp}})$ des fonctions d'énergie finie, dérivables et dont la dérivée est intégrable sur Ω .

On adopte ici la méthode proposée par POCK et coll. dans [12]. Il s'agit d'une méthode globale : on définit une fonctionnelle d'énergie E sur l'ensemble des fonctions $W^{1,2}(\Omega; I_{\text{disp}})$. Celle-ci encode les propriétés de la fonction de disparité que l'on recherche, en ce sens que $E(u)$ sera d'autant plus grand que u ne satisfait pas ces propriétés. Si la fonctionnelle E est correctement choisie, la disparité de la scène la plus probable sera donnée par un minimiseur de E .

L'approche choisie par [12] est une approche variationnelle : la fonctionnelle d'énergie prend la forme

$$E(u) = \int_{\Omega} f(x, u(x), \nabla u(x)) \, dx.$$

Le lagrangien f est supposé continu en ses deux premières variables et convexe en sa troisième variable. Un choix particulier de f permet de séparer ces deux groupes de variables :

$$\forall (x, t) \in \Omega \times \mathbb{R}, \quad \forall p^x \in \mathbb{R}^2, \quad f(x, t, p^x) = g(x, t) + r(p^x)$$

où g est continue sur $\Omega \times \mathbb{R}$ et r convexe. Ce choix conduit à considérer une fonctionnelle d'énergie à deux termes : un terme d'attache aux données (ou de fidélité)

$$E_{\text{data}}(u) = \int_{\Omega} g(x, u(x)) \, dx$$

qui contraint l'algorithme à préférer mettre en correspondance des pixels semblables, et un terme de régularisation

$$E_{\text{reg+vis}}(u) = \int_{\Omega} r(\nabla u(x)) \, dx$$

qui incite la fonction de disparité à être régulière. Nous intégrerons dans ce terme une contrainte de visibilité. Ces deux termes sont sommés et, quitte à remplacer g par g/μ , on peut faire apparaître un paramètre de pondération $\mu > 0$ qui permet de moduler l'importance relative des deux termes :

$$E(u) = \mu E_{\text{data}}(u) + E_{\text{reg+vis}}(u).$$

L'objectif est de trouver un minimum de cette fonctionnelle, c'est-à-dire de résoudre le problème d'optimisation

$$\min_{u \in W^{1,2}(\Omega; I_{\text{disp}})} E(u).$$

Détaillons dans ce qui suit le choix de chacun de ces deux termes.

3.1.1 Une corrélation adaptative pour limiter l'adhérence

Le rôle du terme d'attache aux données est de privilégier la mise en correspondance de pixels semblables. Il est donc naturel de le définir à l'aide d'une mesure de dissimilarité D_{I_L, I_R} (cf. section 2.3.1). Soit u une fonction de disparité candidate. Le pixel homologue de $x \in \Omega$ est alors le pixel $x - u(x)$ ¹. Le coût de corrélation pour cette mise en correspondance est donc donné par $g(x, u(x))$, où g est défini par

$$\forall (x, t) \in \Omega \times \mathbb{R}, \quad g(x, t) = D_{I_L, I_R}(x, x - t)$$

Le coût total de corrélation pour la fonction u est donnée par l'intégrale de ces coûts individuels, ce qui nous permet de définir le terme de fidélité E_{data} .

Corrélation d'intensité combinée à la corrélation de gradient Puisque la combinaison de la corrélation d'intensité et la corrélation de gradient a montré son efficacité [8, 13], nous choisissons pour la mesure de dissimilarité une variante de la mesure utilisée dans [13, 14]. On rappelle que l'idée est de sommer la mesure AD et la corrélation des gradients dans une combinaison convexe, dont le coefficient α est choisi égal à 0,9 dans [13, 14]. L'intérêt est de tirer parti des performances des deux corrélations.

Malheureusement, l'utilisation d'un gradient revient essentiellement à comparer des voisinages. En effet, si le gradient de l'image est calculé à l'aide d'un opérateur linéaire de différences finies, alors il s'agit de comparer deux quantités $a(p, \{p'\})$ et $a(q, \{q'\})$, dépendant respectivement linéairement des voisins p' de p et des voisins q' de q . Or, on sait que la comparaison de voisinages provoque de l'adhérence près des discontinuités de scène. Nous proposons donc de modifier le poids de la corrélation de gradient dans le coût de corrélation, selon la position du pixel considéré : plus le pixel p est situé près d'une discontinuité de scène, moins la corrélation de gradient doit jouer. Cette approche implique de connaître la carte de disparité, ce qui est évidemment exclu. Une manière de procéder est donc de considérer les discontinuités *d'intensité*, qui, si elles ne coïncident pas avec les discontinuités de scène, les contiennent.

Coefficient variable Introduisons donc un coefficient variable $\alpha : \Omega \rightarrow [0; 1]$, choisi de sorte que α soit petit près d'une discontinuité d'intensité et grand ailleurs. Une discontinuité d'intensité étant caractérisée par un fort gradient, nous définissons α comme une fonction décroissante de l'amplitude du gradient de l'image. Pour ne prendre en compte que des gradients significatifs, on effectue au préalable un débruitage ROF de l'image de référence (cf. chapitre 6), ce qui donne la version lissée \tilde{I}_L . Le gradient au point x d'une image couleur $I = (I^r, I^g, I^b)$ est donné par une matrice de taille 2×3 , définie par les dérivées horizontales et verticales (notées respectivement ∂_x et ∂_y) dans chacun des canaux couleur :

$$\forall x \in \Omega, \quad \nabla I(x) = \begin{pmatrix} \partial_x I^r(x) & \partial_x I^g(x) & \partial_x I^b(x) \\ \partial_y I^r(x) & \partial_y I^g(x) & \partial_y I^b(x) \end{pmatrix}. \quad (3.1)$$

On peut alors définir la norme (ou l'amplitude) du gradient en x , notée $\|\nabla I(x)\|$, comme la norme euclidienne du vecteur de \mathbb{R}^6 associé au gradient (c'est-à-dire la norme de FROBENIUS de la matrice $\nabla I(x)$). Dans ce cas, α est défini par

$$\forall x \in \Omega, \quad \alpha(x) = \frac{1}{1 + G \star \|\nabla \tilde{I}_L\|^2(x)/a} \quad (3.2)$$

1. On note par abus de notation $x - t = x - {}^t(t, 0)$ pour tout réel t

avec $a > 0$ un paramètre que l'on choisira ultérieurement. L'amplitude du gradient est ici convoluée avec une gaussienne, ce qui permet de diffuser dans un voisinage les forts gradients.

Corrélation de couleur et de gradient Pour la corrélation d'intensité, on choisit d'exploiter l'information contenue dans la couleur. Contrairement à ce qui est fait dans [13, 14], on utilise la distance induite par la norme euclidienne de \mathbb{R}^3 , notée $\|\cdot\|$, pour comparer les vecteurs couleurs, ce qui conduit à considérer

$$\forall (x, t) \in \Omega \times \mathbb{R}, \quad D_{I_L, I_R}^{\text{AD}}(x, x - t) = \|I_L(x) - I_R(x - t)\| \quad (3.3)$$

Pour la corrélation de gradient, on choisit à nouveau de considérer les variations horizontales et verticales de l'image dans les trois canaux couleurs donnée par la matrice (3.1), puis d'utiliser comme mesure de dissimilarité la distance issue de la norme de FROBENIUS :

$$\forall (x, t) \in \Omega \times \mathbb{R}, \quad D_{I_L, I_R}^{\text{grad}}(x, x - t) = \|\nabla I_L(x) - \nabla I_R(x - t)\|. \quad (3.4)$$

Nouvelle mesure de dissimilarité Les deux coûts de corrélation (3.3) et (3.4) sont combinés à l'aide de la pondération variable α donnée par (3.2) :

$$\forall (x, t) \in \Omega \times \mathbb{R}, \quad g(x, t) = (1 - \alpha(x)) D_{I_L, I_R}^{\text{AD}}(x, x - t) + \alpha(x) D_{I_L, I_R}^{\text{grad}}(x, x - t).$$

Même si la fonction g n'est théoriquement pas continue en (x, t) (car les images considérées ne le sont pas), il est possible de se ramener artificiellement à ce cas en considérant une interpolation régulière de cette fonction, qui est en pratique construite à partir d'un échantillonnage des images et de l'intervalle de disparité.

3.1.2 Une contrainte de pente pour gérer l'occultation

Le terme de régularisation $E_{\text{reg+vis}}$ que nous proposons possède en réalité deux composantes distinctes, une première forçant la régularisation proprement dite de la disparité et une seconde traduisant une contrainte de visibilité :

$$E_{\text{reg+vis}}(u) = E_{\text{reg}}(u) + E_{\text{vis}}(u).$$

Terme de régularisation On choisit d'utiliser la régularisation TV (variation totale), dont nous considérons ici la version isotrope :

$$E_{\text{reg}}^{\text{TV iso}}(u) = \int_{\Omega} \|\nabla u(x)\| \, dx = \|\nabla u\|_1^2.$$

Cette régularisation a été introduite en 1992 par RUDIN, OSHER et FATEMI dans un modèle de débruitage qui porte depuis le nom de *modèle ROF*. L'intérêt principal de ce terme est le suivant : si on étend l'ensemble des fonctions admissibles u à un espace plus grand (voir le paragraphe **Extension aux fonctions BV**), alors ce terme produit des solutions *régulières*, tout en *préservant certaines discontinuités* [3]. Généralement, suivant le poids accordé à ce terme, les solutions sont constantes par morceaux.

2. Cette notation, bien qu'usuelle, est abusive dans le cas d'images multi-valuées. La variation totale, telle qu'elle est définie ici, correspond en réalité à la norme L^1 de la fonction $u \mapsto \|\nabla u(x)\|_2$. Néanmoins, pour éviter les notations trop lourdes, nous continuerons ici d'utiliser la notation standard.

Terme de visibilité Dans l'analyse proposée à la section 2.2, on a montré que, si l'ordre était préservé dans les images, alors la largeur de l'occultation dans l'image de référence est exactement égale au saut de disparité autour de l'occultation. Par ailleurs, nous avons établi que la disparité ne peut pas avoir une pente horizontale excédant 1 sans violer les contraintes de visibilité. De ces deux remarques, on déduit que, si on interpole les occultations de manière affine sur la ligne, alors la pente de la disparité interpolée vaut exactement 1 dans les zones occultées et est strictement inférieur à 1 dans les zones non occultées. Cela nous conduit à introduire le terme de visibilité suivant :

$$E_{\text{vis}}(u) = \int_{\Omega} \chi_{\{\partial_x u \leq 1\}}(x) dx = \begin{cases} 0 & \text{si } \forall x \in \Omega, \quad \partial_x u(x) \leq 1 \\ +\infty & \text{sinon.} \end{cases}$$

qui contraint la pente horizontale de la disparité à être inférieure à 1.

Finalement, le terme de régularisation/visibilité peut s'écrire à l'aide de la fonction

$$\forall p^x = (p_x^x, p_y^x) \in \mathbb{R}^2, \quad r(p^x) = \begin{cases} \|p^x\| & p_x^x \leq 1 \\ +\infty & \text{sinon.} \end{cases}$$

On vérifie aisément que la fonction r est bien convexe.

Extension aux fonctions BV Les fonctions de l'espace $W^{1,2}(\Omega; \mathbb{R})$ ne peuvent présenter aucune discontinuité le long d'une ligne [3] : en d'autres termes, de telles fonctions ne peuvent pas modéliser des contours d'objets. Or, la scène étant composée d'objets distincts, les images et la carte de disparité présentent de telles ruptures de discontinuités (appelées *bords* ou *contours*). L'étude menée dans 2.2 assure qu'elles correspondent, dans la disparité, à des désoccultations. C'est pourquoi il est intéressant d'étendre le domaine de définition de la fonctionnelle E à l'ensemble $BV(\Omega; I_{\text{disp}})$ des fonctions à variations bornées (voir par exemple [7] pour plus de précisions sur l'espace BV).

Le terme de fidélité reste bien défini pour ces fonctions. En revanche, le terme de régularisation/visibilité n'est *a priori* pas défini dans le cas où u n'est pas différentiable. On remarque alors que E_{vis} est une fonction convexe de ∇u , ce qui permet, grâce à [2], de l'appliquer à la mesure de RADON Du associée aux variations de u .

Nous sommes donc à présent amenés à étudier le problème

$$\min_{u \in BV(\Omega; I_{\text{disp}})} \left\{ \int_{\Omega} g(x, u(x)) dx + \int_{\Omega} r(Du) \right\}. \quad (3.5)$$

3.2 Relaxation convexe du problème initial

La fonctionnelle d'énergie E n'est malheureusement pas convexe. L'idée est donc de transformer le problème initial non convexe (3.5) en un problème convexe, d'abord en transformant la fonctionnelle d'énergie E en une énergie d'interface convexe, puis en relaxant le problème. On verra (théorème 10) que les solutions du problème relaxé convexe et celles du problème initial non convexe sont reliées, si bien qu'il suffit de résoudre le problème relaxé pour obtenir des solutions du problème initial.

Nous allons reprendre les différentes étapes présentées dans [12], auquel le lecteur pourra se reporter pour les différentes démonstrations (non reproduites ici).

3.2.1 Énergie d'interface

Indicatrice du sous-graphe Commençons par montrer qu'il est possible de représenter de manière unique toute fonction BV par l'indicatrice d'une partie de \mathbb{R}^3 , son *sous-graphe*. Définissons le sous-graphe d'une fonction u . Si u est une fonction à valeurs réelles définie sur l'ouvert Ω , alors on définit son sous-graphe comme l'ensemble des points $(x, t) \in \Omega \times \mathbb{R}$ vérifiant l'inégalité $u(x) \geq t$. On peut en définir l'indicatrice, notée $\mathbb{1}_u$, définie sur $\Omega \times \mathbb{R}$ et à valeurs dans $\{0, 1\}$, par

$$\forall (x, t) \in \Omega \times \mathbb{R}, \quad \mathbb{1}_u(x, t) = \begin{cases} 1 & \text{si } u(x) \geq t \\ 0 & \text{si } u(x) < t. \end{cases}$$

On notera que, pour tout $x \in \Omega$, la fonction $t \mapsto \mathbb{1}_u(x, t)$ est une fonction constante par morceaux, décroissante de 1 vers 0. On observe alors que

$$\forall x \in \Omega, \quad u(x) = \sup \left\{ t \in \mathbb{R} \mid \mathbb{1}_u(x, t) = 1 \right\}. \quad (3.6)$$

Autrement dit, il est suffisant de connaître $\mathbb{1}_u$ pour connaître entièrement la fonction u .

Cette remarque permet de motiver un changement de variable dans le problème (3.5). Au lieu de chercher à minimiser la fonctionnelle E définie sur la fonction u , on va minimiser une nouvelle énergie F définie sur l'indicatrice $\mathbb{1}_u$. Pour que les deux problèmes restent équivalents, il faut choisir F telle que $F(\mathbb{1}_u) = E(u)$ pour tout $u \in \text{BV}(\Omega; \mathbb{R})$.

Notons que, puisque u est supposée à valeurs dans I_{disp} , alors pour tout $t > \max(I_{\text{disp}})$, le couple (x, t) n'appartient pas au sous-graphe de u , ce qui implique que $\mathbb{1}_u(x, t) = 0$. De même, pour tout $t < \min(I_{\text{disp}})$, on a $\mathbb{1}_u(x, t) = 1$.

Nouvelle fonctionnelle d'énergie Introduisons [12] le lagrangien h suivant, défini pour tout $(x, t) \in \Omega \times \mathbb{R}$ et tout $p = (p^x, p^t) \in \mathbb{R}^2 \times \mathbb{R}$ par :

$$h(x, t, p) = \begin{cases} |p^t| g(x, t) + |p^t| r(p^x / |p^t|) & \text{si } p^t < 0 \\ \lim_{\lambda \rightarrow 0^+} \lambda r(p^x / \lambda) & \text{si } p^t = 0 \\ +\infty & \text{si } p^t > 0 \end{cases}$$

qui se réécrit ici

$$h(x, t, p) = \begin{cases} |p^t| g(x, t) + r(p^x) & \text{si } p^t \leq 0 \text{ et } p_x^x \leq -p^t \\ +\infty & \text{si } p^t > 0 \text{ ou } p_x^x > -p^t. \end{cases}$$

On peut décomposer ce lagrangien en deux termes h^{TV} et h^{vis} . Le premier terme est donné par

$$h^{\text{TV}}(x, t, p) = \begin{cases} |p^t| g(x, t) + \|p^x\| & \text{si } p^t \leq 0 \\ +\infty & \text{si } p^t > 0 \end{cases}$$

et correspond au lagrangien associé à la régularisation TV sans contrainte de visibilité. Le second terme ne dépend que de la variable p et vaut

$$h^{\text{vis}}(p) = \begin{cases} 0 & \text{si } p_x^x \leq -p^t \\ +\infty & \text{si } p_x^x > -p^t. \end{cases}$$

Dans ce terme, qui est convexe, est encodée la contrainte de visibilité. Si on pose pour tout $v \in \text{BV}(\Omega \times \mathbb{R}; [0; 1])$

$$F(v) = \int_{\Omega \times \mathbb{R}} h(x, t, Dv) = \int_{\Omega \times \mathbb{R}} h^{\text{TV}}(x, t, Dv) + \int_{\Omega \times \mathbb{R}} h^{\text{vis}}(Dv) \quad (3.7)$$

alors $F(\mathbb{1}_u) = E(u)$ quelle que soit la fonction u [12]. En effet, notons que, si u est à variation bornée, alors l'indicatrice de son sous-graphe $\mathbb{1}_u$ est dans $\text{BV}(\Omega \times \mathbb{R}; [0; 1])$. En d'autres termes, le sous-graphe de toute fonction BV est à périmètre fini dans \mathbb{R}^3 . Le théorème de GAUSS-GREEN assure alors que

$$D\mathbb{1}_u = \nu_{\Gamma_u}(x, t) \, d\mathcal{H}^2(x, t) \llcorner \Gamma_u$$

où on note Γ_u la frontière du sous-graphe de u ³ et ν_{Γ_u} la normale extérieure au sous-graphe de u , définie par

$$\forall (x, t) \in \Gamma_u, \quad \nu_{\Gamma_u}(x, t) = \begin{pmatrix} \nabla u(x) \\ -1 \end{pmatrix}$$

si $u \in W^{1,1}(\Omega; \mathbb{R})$, et \mathcal{H}^2 est la mesure de HAUSDORFF [17, chapitre V] de dimension 2. On en déduit l'identité souhaitée (\llcorner symbolisant la restriction).

On notera que F est une fonctionnelle convexe. Par ailleurs, $F(u)$ est s'écrit en fonction de la frontière du sous-graphe de u . C'est pourquoi on parle d'énergie d'*interface*.

Problème équivalent Le problème initial (3.5) est donc équivalent au problème

$$\min_{u \in \text{BV}(\Omega; \text{I}_{\text{disp}})} \left\{ \int_{\Omega \times \mathbb{R}} h(x, t, D\mathbb{1}_u) \right\} \quad (3.8)$$

qui s'écrit encore

$$\min_{\substack{v \in \text{BV}(\Omega \times \mathbb{R}; [0; 1]) \\ v = \mathbb{1}_u, u \in \text{BV}(\Omega; \text{I}_{\text{disp}})}} \left\{ \int_{\Omega \times \mathbb{R}} h(x, t, Dv) \right\}.$$

Le problème (3.8) est donc un problème de minimisation de l'énergie convexe F sur l'ensemble non convexe des fonctions qui sont indicatrices du sous-graphe d'une fonction à variation bornée de Ω .

3.2.2 Relaxation convexe

Pour rendre ce problème convexe, il suffit de procéder à une relaxation convexe, c'est-à-dire de remplacer l'ensemble des fonctions admissibles par un ensemble plus grand, mais convexe. Ainsi, le problème considéré devient la minimisation d'une énergie convexe sur un ensemble de fonctions admissibles convexes, c'est-à-dire un problème d'optimisation convexe. On pourra alors exploiter les outils de l'analyse convexe, qui assurent en particulier que le problème possède un minimum global.

Une manière de procéder est de considérer l'enveloppe convexe de l'ensemble des fonctions $\{v = \mathbb{1}_u \mid u \in \text{BV}(\Omega; \mathbb{R})\}$, qui est l'ensemble des fonctions de $\text{BV}(\Omega \times \mathbb{R}; [0; 1])$ décroissantes selon leur seconde variable. On va toutefois suivre la démarche proposée par [12] et choisir une relaxation plus importante, en considérant l'ensemble suivant

$$\mathcal{C} = \left\{ v \in \text{BV}(\Omega \times \mathbb{R}; [0; 1]) \mid \forall x \in \Omega, \begin{cases} \forall t \leq \min(\text{I}_{\text{disp}}), v(x, t) = 1 \\ \forall t \geq \max(\text{I}_{\text{disp}}), v(x, t) = 0 \end{cases} \right\} \quad (3.9)$$

3. Il s'agit du graphe de u quand u est continue.

qui est convexe et contient bien l'ensemble des indicatrices de sous-graphe de fonctions BV. Cette relaxation conduit donc à considérer le problème convexe

$$\min_{v \in \mathcal{C}} \left\{ \int_{\Omega \times \mathbb{R}} h(x, t, Dv) \right\}. \quad (3.10)$$

Quel est alors le lien entre les solutions du problème convexe (3.10) et celles (si elles existent) du problème initial (3.5) ? C'est l'objet du théorème suivant :

Théorème 10 ([12]) *Soit v^* un minimiseur global du problème relaxé (3.10). Alors, pour tout $s \in [0; 1[$, la fonction caractéristique $\mathbf{1}_{\{v^* > s\}}$ est l'indicatrice du sous-graphe d'une fonction u^* , qui est un minimiseur du problème initial (3.5).*

Autrement dit, pour obtenir un minimiseur du problème initial, il suffit de résoudre le problème relaxé, puis d'en seuiller la solution avec n'importe quel $s \in [0; 1[$. À l'aide de la formule de reconstruction (3.6), on obtient une solution du problème initial.

3.2.3 Formulation primale-duale

Le théorème 10 assurant que des minimiseurs globaux de la fonctionnelle E peuvent être obtenus à partir de minimiseurs globaux de l'énergie d'interface F sur le convexe \mathcal{C} , on s'attache à présent à résoudre le problème relaxé (3.10). On montre pour cela qu'il est possible de l'écrire sous une forme primale-duale.

Soit $v \in \mathcal{C}$. D'après le théorème 3.2 de [12], on a

$$\int_{\Omega \times \mathbb{R}} h^{\text{TV}}(x, t, Dv) = \sup_{\phi \in \mathcal{K}} \int_{\Omega \times \mathbb{R}} \phi Dv,$$

où \mathcal{K} est le convexe défini par

$$\mathcal{K} = \left\{ \phi \in \mathcal{C}^0(\Omega \times \mathbb{R}; \mathbb{R}^3) \mid \forall (x, t) \in \Omega \times \mathbb{R}, \phi^t(x, t) + \mu g(x, t) \geq 0 \text{ et } \|\phi^x(x, t)\| \leq 1 \right\}$$

avec $\phi = (\phi^x, \phi^t)$. On est donc amené à résoudre le problème de recherche de point-selle suivant

$$\min_{v \in \mathcal{C}} \sup_{\phi \in \mathcal{K}} \left\{ \int_{\Omega \times \mathbb{R}} \phi Dv + \int_{\Omega \times \mathbb{R}} h^{\text{vis}}(Dv) \right\}. \quad (3.11)$$

3.3 Résolution numérique par algorithme primal-dual

Pour résoudre numériquement le problème primal-dual (3.11), en vue d'obtenir des solutions du problème initial (3.5), on commence par discrétiser les données du problème primal-dual. Nous proposerons ensuite deux algorithmes pour résoudre le problème discret qui en découle.

3.3.1 Discrétisation du problème

Images Notons I_L^h et I_R^h les deux images numériques de la paire stéréoscopique. Dans le cas d'images en niveaux de gris, ces deux images sont deux matrices de taille $N_x \times N_y$ et dans le cas d'images couleur RGB, on les représente comme deux triplets de matrices

de taille $N_x \times N_y$. On suppose que le domaine de l'image continue I_L est rectangulaire, donné par $\Omega =]0; A_x[\times]0; A_y[$. Sa discrétisation est alors donnée par la grille régulière

$$\Omega^h = \left\{ (i h_x, j h_y) \mid (i, j) \in \llbracket 0; N_x - 1 \rrbracket \times \llbracket 0; N_y - 1 \rrbracket \right\}$$

avec $N_x = \lfloor A_x/h_x \rfloor + 1$ et $N_y = \lfloor A_y/h_y \rfloor + 1$ deux entiers positifs, correspondant respectivement à la largeur (horizontale) et à la hauteur (verticale) de l'image en pixels. Les deux réels positifs h_x et h_y sont les pas d'échantillonnage (horizontal et vertical) des images. Les pas de discrétisation de la grille d'échantillonnage sont désormais choisis égaux à $h_x = h_y = 1$, ce qui signifie que les images ne sont ni sous-échantillonnées ni sur-échantillonnées.

Soit $(i, j) \in \llbracket 0; N_x - 1 \rrbracket \times \llbracket 0; N_y - 1 \rrbracket$. Le pixel d'indice (i, j) de l'image de référence est donné par le coefficient (ou le vecteur, dans le cas d'image couleur) $(I_L^h)_{i,j}$, et, dans une première approximation, on supposera que $(I_L^h)_{i,j} = I_L(i, j)$ ⁴ (et de même pour l'image de droite).

Intervalle de disparité On discrétise ensuite l'intervalle de disparité $I_{\text{disp}} = [d_{\min}; d_{\max}]$:

$$I_{\text{disp}}^h = \left\{ d_{\min} + k h_t \mid k \in \llbracket 0; N_t - 1 \rrbracket \right\}$$

où l'entier naturel $N_t = \lfloor (d_{\max} - d_{\min})/h_t \rfloor + 1$ correspond au nombre de profondeurs différentes que compte la carte du relief recherchée et h_t est le pas de quantification de la carte de disparité. On note alors $h = (1, 1, h_t)$ le vecteur des pas d'échantillonnage et $G^h = \Omega^h \times I_{\text{disp}}^h$ la grille d'échantillonnage du volume $\Omega \times I_{\text{disp}}$.

Terme d'attache aux données On introduit ensuite le *volume de coût* g^h , défini par

$$\forall (i, j, k h_t) \in G^h, \quad g_{i,j,k}^h = g((i, j), k h_t).$$

Le volume de coût est un volume 3D tel que $g_{i,j,k}^h$ donne pour le pixel (i, j) le coût de corrélation associé à la disparité $k h_t$.

Dans le cas d'un pas unitaire dans la grille d'échantillonnage des images, le coefficient $g_{i,j,k}^h$ compare le pixel $(i, j) \in \mathbb{N}^2$ de l'image de référence avec le pixel $(i, j - k h_t)$ de l'image de droite. Supposons que la corrélation choisie est la corrélation d'intensité. Si $(i, j - k h_t)$ appartient à la grille d'échantillonnage de l'image de droite (c'est-à-dire si $k h_t$ est un entier), alors l'intensité de l'image en ce point est connue, et le coût $g_{i,j,k}^h$ est une fonction de $(I_L^h)_{i,j}$ et $(I_R^h)_{i,j-k h_t}$. Si $(i, j - k h_t)$ n'appartient pas à la grille d'échantillonnage de l'image de droite, alors une étape de sur-échantillonnage de l'image de droite est nécessaire pour interpoler l'intensité de I_R^h au point $(i, j - k h_t)$. L'interpolation que nous choisissons est l'interpolation B-spline d'ordre 5 [16, 15].

Convexes On peut ensuite discrétiser les convexes \mathcal{C} et \mathcal{K} , en posant

$$\mathcal{C}^h = \left\{ v^h \in [0; 1]^{N_x N_y N_t} \mid \forall (i, j) \in \Omega^h, v_{i,j,0}^h = 1 \text{ et } v_{i,j,N_t-1}^h = 0 \right\}$$

(suite à la remarque, soulignée au paragraphe 3.2.1, concernant les valeurs prises par l'indicatrice $\mathbb{1}_u$ lorsque t est supérieur à $\max(I_{\text{disp}})$ et lorsque t est inférieur à $\min(I_{\text{disp}})$), et

$$\mathcal{K}^h = \left\{ \phi^h \in \mathbb{R}^{3N_x N_y N_t} \mid \forall (i, j, k h_t) \in G^h, (\phi^h)_{i,j,k}^t + \mu g_{i,j,k}^h \geq 0 \text{ et } \|((\phi^h)_{i,j,k}^x, (\phi^h)_{i,j,k}^y)\| \leq 1 \right\}$$

4. En réalité, cette égalité est simpliste, car l'échantillonnage implique une étape préalable de filtrage de l'image, qui correspond à un filtre *anti-aliasing*, ou anti-recouvrement de spectre.

où on note cette fois $\phi_{i,j,k}^h = ((\phi^h)_{i,j,k}^x, (\phi^h)_{i,j,k}^y, (\phi^h)_{i,j,k}^t) \in \mathbb{R}^3$.

Opérateurs gradient et divergence On choisit pour la version discrète de l'opérateur gradient un opérateur linéaire $\nabla^h : \mathbb{R}^{N_x N_y N_t} \rightarrow (\mathbb{R}^3)^{N_x N_y N_t}$ construit à l'aide de différences finies avec des conditions de bord de type NEUMANN : pour tout $v^h \in \mathbb{R}^{N_x N_y N_t}$,

$$\forall (i,j,k, h_t) \in G^h, \quad (\nabla^h v^h)_{i,j,k} = \begin{pmatrix} (\delta_x^h v^h)_{i,j,k} \\ (\delta_y^h v^h)_{i,j,k} \\ (\delta_t^h v^h)_{i,j,k} \end{pmatrix}$$

où les différences finies $(\delta_x^h v^h)_{i,j,k}$, $(\delta_y^h v^h)_{i,j,k}$ et $(\delta_t^h v^h)_{i,j,k}$ sont données pour tout indice (i,j,k) par

$$(\delta_x^h v^h)_{i,j,k} = \begin{cases} v_{i+1,j,k}^h - v_{i,j,k}^h & \text{si } i < N_x - 1 \\ 0 & \text{si } i = N_x - 1 \end{cases}$$

$$(\delta_y^h v^h)_{i,j,k} = \begin{cases} v_{i,j+1,k}^h - v_{i,j,k}^h & \text{si } j < N_y - 1 \\ 0 & \text{si } j = N_y - 1 \end{cases}$$

et

$$(\delta_t^h v^h)_{i,j,k} = \begin{cases} \frac{v_{i,j,k+1}^h - v_{i,j,k}^h}{h_t} & \text{si } k < N_t - 1 \\ 0 & \text{si } k = N_t - 1. \end{cases}$$

Cet opérateur admet un adjoint, noté div^h , donné pour tout $\phi^h \in \mathbb{R}^{3N_x N_y N_t}$ par

$$\forall (i,j,k, h_t) \in G^h, \quad (\text{div}^h v^h)_{i,j,k} = (\varepsilon_x^h(\phi^h)^x)_{i,j,k} + (\varepsilon_y^h(\phi^h)^y)_{i,j,k} + (\varepsilon_t^h(\phi^h)^t)_{i,j,k}$$

où les différences finies $(\varepsilon_x^h(\phi^h)^x)_{i,j,k}$, $(\varepsilon_y^h(\phi^h)^y)_{i,j,k}$ et $(\varepsilon_t^h(\phi^h)^t)_{i,j,k}$ sont données pour tout indice (i,j,k) par

$$(\varepsilon_x^h(\phi^h)^x)_{i,j,k} = \begin{cases} -(\phi^h)_{i,j,k}^x & \text{si } i = 0 \\ (\phi^h)_{i-1,j,k}^x - (\phi^h)_{i,j,k}^x & \text{si } 0 < i < N_x - 1 \\ (\phi^h)_{i-1,j,k}^x & \text{si } i = N_x - 1 \end{cases}$$

$$(\varepsilon_y^h(\phi^h)^y)_{i,j,k} = \begin{cases} -(\phi^h)_{i,j,k}^y & \text{si } j = 0 \\ (\phi^h)_{i,j-1,k}^y - (\phi^h)_{i,j,k}^y & \text{si } 0 < j < N_y - 1 \\ (\phi^h)_{i,j-1,k}^y & \text{si } j = N_y - 1 \end{cases}$$

et

$$(\varepsilon_t^h(\phi^h)^t)_{i,j,k} = \begin{cases} -\frac{(\phi^h)_{i,j,k}^t}{h_t} & \text{si } k = 0 \\ \frac{(\phi^h)_{i-1,j,k}^x - (\phi^h)_{i,j,k}^x}{h_t} & \text{si } 0 < k < N_t - 1 \\ \frac{(\phi^h)_{i-1,j,k}^x}{h_t} & \text{si } k = N_t - 1. \end{cases}$$

Problème discret Le problème primal-dual devient alors le problème de recherche de point-selle

$$\min_{v^h \in C^h} \sup_{\phi^h \in K^h} \left\{ \langle \phi^h, \nabla^h v^h \rangle + \sum_{(i,j,k,h_t) \in G^h} \chi_{]-\infty; 0]} \left((\delta_x^h v^h)_{i,j,k} + (\delta_t^h v^h)_{i,j,k} \right) \right\} \quad (3.12)$$

où $\chi_{]-\infty;0]}$ est la fonction caractéristique de l'intervalle $]-\infty;0]$. Elle vaut 0 sur cet intervalle et vaut une valeur infinie sur $]0;+\infty[$. On peut alors introduire un multiplicateur de LAGRANGE supplémentaire pour le second terme, car, pour tout $x \in \mathbb{R}$, on a

$$\chi_{]-\infty;0]}(x) = \sup_{\lambda \geq 0} \lambda x.$$

Le problème (3.12) se réécrit alors

$$\min_{v^h \in \mathcal{C}^h} \sup_{\substack{\phi^h \in \mathcal{K}^h \\ \lambda^h \geq 0}} \left\{ \langle \phi^h, \nabla^h v^h \rangle + \langle \lambda^h, \delta_x^h v^h + \delta_t^h v^h \rangle \right\} \quad (3.13)$$

où, pour simplifier les notations, on écrira $\lambda^h \geq 0$ à la place de $\lambda^h \in ([0;+\infty[)^{N_x N_y N_t}$.

3.3.2 Algorithme primal-dual

On peut résoudre le problème de recherche de point-selle (3.13) en utilisant un algorithme primal-dual analogue à celui proposé dans [12], mais accéléré selon [5].

Algorithme primal-dual sur-relaxé On commence par initialiser l'algorithme avec les variables $(v^h)_0 = (\bar{v}^h)_0 \in \mathcal{C}^h$, $(\phi^h)_0 \in \mathcal{K}^h$ et $(\lambda^h)_0 \geq 0$, puis on effectue pour tout $n \in \mathbb{N}$ les mises-à-jours suivantes

$$\begin{cases} (\hat{\phi}^h)_{n+1} &= \text{proj}_{\mathcal{K}^h} \left((\phi^h)_n + \sigma \nabla^h (\bar{v}^h)_n \right) \\ (\hat{\lambda}^h)_{n+1} &= \text{proj}_{([0;+\infty[)^{N_x N_y N_t}} \left((\lambda^h)_n + \sigma (\delta_x^h + \delta_t^h) (\bar{v}^h)_n \right) \\ (\hat{v}^h)_{n+1} &= \text{proj}_{\mathcal{C}^h} \left((v^h)_n - \tau (\text{div}^h (\phi^h)_{n+1} + (\varepsilon_x^h + \varepsilon_t^h) (\lambda^h)_{n+1}) \right) \\ (\bar{v}^h)_{n+1} &= 2 (\hat{v}^h)_{n+1} - (v^h)_n \end{cases}$$

où $\text{proj}_{\mathcal{K}^h}$ (resp. $\text{proj}_{([0;+\infty[)^{N_x N_y N_t}}$ et $\text{proj}_{\mathcal{C}^h}$) désigne la projection sur le convexe \mathcal{K}^h (resp. $([0;+\infty[)^{N_x N_y N_t}$ et \mathcal{C}^h). On ajoute ensuite une étape de sur-relaxation globale :

$$\begin{pmatrix} (v^h)_{n+1} \\ (\phi^h)_{n+1} \\ (\lambda^h)_{n+1} \end{pmatrix} = (1 - \rho) \begin{pmatrix} (v^h)_n \\ (\phi^h)_n \\ (\lambda^h)_n \end{pmatrix} + \rho \begin{pmatrix} (\hat{v}^h)_{n+1} \\ (\hat{\phi}^h)_{n+1} \\ (\hat{\lambda}^h)_{n+1} \end{pmatrix}$$

de paramètre $\rho \in]0;2[$.

Il s'agit d'un algorithme effectuant dans un premier temps un pas de montée de gradient projeté en les variables duales (ϕ^h, λ^h) et un pas de descente de gradient projeté en la variable primale v^h , suivis d'une première sur-relaxation en v^h . Ensuite, une étape de sur-relaxation globale est réalisée sur les trois variables. On montre [4] que cet algorithme converge vers une solution du problème discret (3.13) sur les paramètres sont correctement choisis. Lorsque le paramètre de sur-relaxation ρ vaut 1, on retrouve l'algorithme proposé dans [11]. Lorsqu'il est proche de 2, on observe en pratique une accélération de la convergence, mais qui n'est pas expliquée de manière théorique.

Choix des paramètres Les pas de temps σ et τ doivent être choisis de sorte que $\sigma \tau L^2 < 1$, avec L la norme de l'opérateur $A = \nabla^h + \delta_x^h + \delta_t^h$. On montre de la même manière que dans [12] que L est majoré par $2\sqrt{4 + 1/h_t^2 + 1/h_t^4}$. On choisira par exemple $\tau = 0,1$ et $\sigma = 1/(\tau L^2)$. Par ailleurs, on choisira $\rho = 1,95$.

Calcul des projections Les trois projections en jeu dans cet algorithme peuvent se calculer de manière indépendante en chaque indice (i, j, k) . La projection sur \mathcal{K}^h est équivalente à une projection sur la boule unité de \mathbb{R}^2 [12], tandis que la projection sur \mathcal{C}^h devient, suivant l'indice, une projection sur 0, 1 ou le segment $[0; 1]$. Enfin, la projection sur $([0; +\infty])^{N_x N_y N_t}$ est immédiate.

REMARQUE : C'est la simplicité de ces projections qui a principalement motivé l'introduction du multiplicateur de LAGRANGE λ^h . On pouvait en effet appliquer directement le résultat démontré dans [12] au lagrangien h , au lieu de le décomposer comme nous l'avons fait. Malheureusement, une telle démarche conduit à introduire un autre convexe \mathcal{K}^h , dont la forme plus complexe rend la projection difficile. De même, c'est la raison pour laquelle la relaxation choisie n'est pas celle où \mathcal{C} est l'enveloppe convexe des fonctions admissibles du problème (3.10), car elle aurait conduit à une projection sur l'ensemble \mathcal{C}^h des vecteurs décroissants selon la troisième dimension (à ce sujet, voir le pseudo-code et les références dans [6]).

Convergence Le théorème 2 de [11] assure la convergence de l'algorithme proposé au début de ce paragraphe, mais ne donne aucune information quant au nombre d'itérations nécessaires pour s'approcher de la convergence.

Un outil généralement utilisé pour mesurer cette convergence est le *primal-dual gap*, défini pour le problème (3.13) et pour toute itération $n \in \mathbb{N}$ comme suit :

$$\begin{aligned} \mathcal{G}((v^h)_n, (\phi^h)_n, (\lambda^h)_n) &= \sup_{\substack{\phi^h \in \mathcal{K}^h \\ \lambda^h \geq 0}} \left\{ \langle \phi^h, \nabla^h(v^h)_n \rangle + \langle \lambda^h, \delta_x^h(v^h)_n + \delta_t^h(v^h)_n \rangle \right\} \\ &\quad - \min_{v^h \in \mathcal{C}^h} \left\{ \langle (\phi^h)_n, \nabla^h v^h \rangle + \langle (\lambda^h)_n, \delta_x^h v^h + \delta_t^h v^h \rangle \right\}. \end{aligned}$$

Cette quantité est positive et tend vers 0 lorsque n tend vers $+\infty$. On peut alors s'en servir comme d'un critère d'arrêt de l'algorithme. Une première façon de procéder est de fixer une valeur de tolérance, et de stopper les itérations dès que le *gap* tombe en-dessous de cette valeur. Une autre démarche [12] consiste à arrêter l'algorithme dès que la valeur du *gap* a été divisée par une certaine valeur (1000 par exemple).

Malheureusement, dans notre cas précis, l'utilisation du *gap* n'est pas toujours appropriée. Les volumes $(v^h)_n$ n'étant pas explicitement contraints à satisfaire la condition

$$\forall (i, j, k, h_t) \in G^h, \quad (\delta_x^h(v^h)_n)_{i,j,k} + (\delta_t^h(v^h)_n)_{i,j,k} \leq 0$$

le premier terme dans $\mathcal{G}((v^h)_n, (\phi^h)_n, (\lambda^h)_n)$ prend généralement une valeur infinie, quel que soit le nombre de coefficients qui ne respectent pas cette contrainte. On peut alors soit ignorer ce terme dans le calcul du *gap*, soit affecter une pénalité (finie) à chaque coefficient qui ne satisfait pas cette majoration.

Seuillage de la solution Pour obtenir la carte de disparité u^h à partir de la solution v^h obtenue après N itérations de l'algorithme, on applique le théorème 10. On réalise un seuillage pour obtenir un niveau $s \in [0; 1[$ de v^h , puis on reconstruit u^h à l'aide de la version discrète de la formule (3.6) :

$$\forall (i, j) \in \Omega^h, \quad u_{i,j}^h = h_t \sup \left\{ k \in \llbracket 0; N_t - 1 \rrbracket \mid v_{i,j,k}^h > s \right\}.$$

3.3.3 Accélération par convexification

Retour sur le problème continu On rappelle qu'on cherche à minimiser sur le convexe \mathcal{C} la fonctionnelle F donnée par la formule (3.7). Le théorème 10 assure même qu'il suffit de trouver un ensemble de niveau d'un minimiseur v^* de F . Par ailleurs, la preuve du théorème 3.1 de [12] assure que, pour presque tout $s \in [0; 1[$, l'indicatrice $\mathbb{1}_{\{v^* > s\}} \in \mathcal{C}$ de cet ensemble de niveau est elle-même minimiseur de la fonctionnelle F . En d'autres termes, l'ensemble $\{v^* > s\}$ est solution du problème convexe

$$\min_{\substack{E \subset \Omega \times \mathbb{R} \\ E \text{ satisfaisant } (\star)}} F(\mathbb{1}_E)$$

$$\text{avec } \Omega \times]-\infty; \min(I_{\text{disp}})] \subset E \quad \text{et} \quad \Omega \times [\max(I_{\text{disp}}); +\infty[\cap E = \emptyset. \quad (\star)$$

Nous allons montrer que les solutions de ce problème sont liées à la solution d'un problème fortement convexe, que l'on cherchera donc à résoudre.

Formule de la co-aire D'après la preuve du théorème 3.1 de [12], la fonctionnelle F satisfait la *formule de la co-aire* : pour tout $v \in \text{BV}(\Omega \times \mathbb{R})$,

$$F(v) = \int_{-\infty}^{+\infty} F(\mathbb{1}_{\{v > s\}}) ds.$$

Puisque la fonction v est supposée à valeurs dans $[0; 1]$, les ensembles de niveaux supérieurs à 1 sont vides et les ensembles de niveaux inférieurs à 0 sont l'ensemble $\Omega \times \mathbb{R}$ tout entier. Ainsi, la formule de la co-aire se réécrit dans ce cas précis

$$F(v) = \int_0^1 F(\mathbb{1}_{\{v > s\}}) ds.$$

Introduction d'un problème fortement convexe En reprenant la démarche du paragraphe 2.2.2 des notes [3], nous allons montrer que

Théorème 11 Soit $M > 0$. La fonction $w^* \in \text{BV}(\Omega \times \mathbb{R})$ est l'unique solution du problème fortement convexe

$$\min_{w \in \mathcal{D}} F(w) + \frac{1}{2} \|w\|_{L^2(\Omega \times I_{\text{disp}})}^2 \quad (3.14)$$

avec

$$\mathcal{D} = \left\{ w \in \text{BV}(\Omega \times \mathbb{R}; \mathbb{R}) \left| \begin{array}{l} \forall t \leq \min(I_{\text{disp}}), w(x, t) = +M \\ \forall t \geq \max(I_{\text{disp}}), w(x, t) = -M \end{array} \right. \right\} \quad (3.15)$$

si et seulement si ses ensembles de niveaux $\{(x, t) \in \Omega \times \mathbb{R} \mid w(x, t) > s\}$ sont solutions des problèmes

$$\min_{\substack{E \subset \Omega \times \mathbb{R} \\ E \text{ satisfaisant } (\star)}} F(\mathbb{1}_E) + s |E \cap (\Omega \times I_{\text{disp}})| \quad (3.16)$$

pour tout $s \in [-M; M]$.

Pour cela, il suffit de montrer que les solutions du problème (3.16) sont des ensembles décroissants pour l'inclusion, et permet donc, à l'aide de la formule (3.6), de définir une fonction. Puis on montre que cette fonction est la solution du problème fortement convexe (3.14). L'unicité de cette solution prouve alors la réciproque du théorème.

DÉMONSTRATION : Commençons par démontrer que le lemme 2.4 de [3]. Pour tout réel s , posons E_s une solution du problème (3.16). Montrons alors que pour tous $-M \leq s < s' \leq M$, on a l'inclusion $E_{s'} \subset E_s$. Il suffit pour cela de démontrer l'inégalité suivante pour tous ensembles A et B

$$F(\mathbb{1}_{A \cup B}) + F(\mathbb{1}_{A \cap B}) \leq F(\mathbb{1}_A) + F(\mathbb{1}_B).$$

On commence par remarquer que, géométriquement, il est aisé de vérifier que

$$F^{\text{vis}}(\mathbb{1}_{A \cup B}) + F^{\text{vis}}(\mathbb{1}_{A \cap B}) \leq F^{\text{vis}}(\mathbb{1}_A) + F^{\text{vis}}(\mathbb{1}_B)$$

car si le membre de droite est fini (le cas infini étant trivial), alors cela implique uniquement des conditions sur la frontière de chacun des deux ensembles A et B , conditions qui restent préservées par l'union et l'intersection de ces deux ensembles. Par ailleurs, si A n'est pas l'ensemble des ensembles de niveaux d'une fonction u , alors on peut montrer que $F^{\text{TV}}(\mathbb{1}_A) = +\infty$. À nouveau, cela nous conduit à ne considérer que le cas où $\mathbb{1}_A = \mathbb{1}_{u_A}$ et $\mathbb{1}_B = \mathbb{1}_{u_B}$ sont des indicatrices de sous-graphes. Dans ce cas, l'union $A \cup B$ correspond à la fonction $\max(u_A, u_B)$, tandis que l'intersection correspond à la fonction $\min(u_A, u_B)$. On a alors d'une part pour tout $x \in \Omega$

$$g(x, \max(u_A(x), u_B(x))) + g(x, \min(u_A(x), u_B(x))) = g(x, u_A(x)) + g(x, u_B(x))$$

et d'autre part

$$\text{TV}(\max(u_A, u_B)) + \text{TV}(\min(u_A, u_B)) \leq \text{TV}(u_A) + \text{TV}(u_B).$$

On en déduit alors que

$$F^{\text{TV}}(\mathbb{1}_{A \cup B}) + F^{\text{TV}}(\mathbb{1}_{A \cap B}) \leq F^{\text{TV}}(\mathbb{1}_A) + F^{\text{TV}}(\mathbb{1}_B).$$

Pour tout $s \in \mathbb{R}$, l'ensemble E_s désigne désormais une solution du problème (3.16) (dont on peut montrer qu'elle est en réalité unique pour presque tout s). Montrons que

$$w^* : (x, t) \mapsto \begin{cases} \sup \left\{ s \in [-M; M] \mid (x, t) \in E_s \right\} & \text{si } \exists s \in [-M; M], (x, t) \in E_s \\ -M & \text{sinon} \end{cases}$$

est solution de (3.14), en suivant la démonstration du lemme 2.5 de [3]. La fonction w^* est bien de carré intégrable, car elle est bornée sur un domaine borné. Montrons que w^* est bien d'énergie minimale. Pour cela, on commence par remarquer que, pour tout $s \in [-M; M]$,

$$E_s^- = \bigcap_{s' > s} E_{s'} = \{w^* > s\}$$

est la plus petite solution de (3.16). On peut alors reprendre la preuve du lemme 2.5 de [3] (avec $g = 0$ d'une part et en utilisant la formule de la co-aire pour F d'autre part).

Résolution numérique du problème fortement convexe La discrétisation du problème (3.14) conduit à considérer le problème suivant

$$\min_{w^h \in \mathcal{D}^h} \sup_{\substack{\phi^h \in \mathcal{K}^h \\ \lambda^h \geq 0}} \left\{ \langle \phi^h, \nabla^h w^h \rangle + \langle \lambda^h, \delta_x^h w^h + \delta_t^h w^h \rangle + \frac{1}{2} \|w^h\|_2^2 \right\} \quad (3.17)$$

où on utilise à nouveau le multiplicateur de LAGRANGE λ^h . La discrétisation du convexe \mathcal{D} est donnée par

$$\mathcal{D}^h = \left\{ v^h \in \mathbb{R}^{N_x N_y N_t} \mid \forall (i, j) \in \Omega^h, v_{i,j,0}^h = -M \text{ et } v_{i,j,N_t-1}^h = M \right\}$$

On résout ce problème à nouveau par algorithme primal-dual, ce qui nous amène à proposer l'algorithme suivant : on initialise la variable primale $(w^h)_0 = (\bar{w}^h)_0 \in \mathcal{D}^h$ et les variables duales $(\phi^h)_0 \in \mathcal{K}^h$ et $(\lambda^h)_0 \geq 0$, puis on effectue pour tout $n \in \mathbb{N}$ les mises-à-jours suivantes

$$\begin{cases} (\phi^h)_{n+1} &= \text{proj}_{\mathcal{K}^h} \left((\phi^h)_n + \sigma \nabla^h (\bar{w}^h)_n \right) \\ (\lambda^h)_{n+1} &= \text{proj}_{([0; +\infty[)^{N_x N_y N_t}} \left((\lambda^h)_n + \sigma (\delta_x^h + \delta_t^h) (\bar{w}^h)_n \right) \\ (w^h)_{n+1} &= \text{proj}_{\mathcal{D}^h} \left(\frac{(w^h)_n - \tau \left(\text{div}^h (\phi^h)_{n+1} + (\varepsilon_x^h + \varepsilon_t^h) (\lambda^h)_{n+1} \right)}{1 + \tau} \right) \\ (\bar{w}^h)_{n+1} &= (1 + \theta_{n+1}) (w^h)_{n+1} - \theta_{n+1} (w^h)_n \end{cases}$$

qui alterne cette fois une montée de gradient projeté en (ϕ^h, λ^h) et une descente de gradient projeté en w^h . L'intérêt majeur de cet algorithme est qu'il résout un problème fortement convexe : d'une part, cela assure l'unicité de la solution, et d'autre part, la convergence est plus rapide.

Choix des paramètres On choisit d'initialiser les pas de temps avec $\tau_0 = 0,1$ d'une part et $\sigma_0 = 1/(\tau_0 L^2)$ d'autre part, puis de les mettre à jour de la manière suivante :

$$\forall n \in \mathbb{N}, \quad \tau_{n+1} = \tau_n \theta_{n+1} \quad \text{et} \quad \sigma_{n+1} = \frac{\sigma_n}{\theta_{n+1}}.$$

Le paramètre de relaxation variable θ_n est quant à lui choisi vérifiant

$$\forall n \in \mathbb{N}, \quad \theta_{n+1} = \frac{1}{\sqrt{1 + 1,2 \tau_n}}.$$

Convergence Puisque c'est le niveau 0 de la solution w^h qui nous intéresse, on peut également choisir de stopper les itérations lorsque ce niveau se stabilise. Pour cela, toutes les dix itérations par exemple, on compte le nombre de coefficients de la variable primale w^h qui ont changé de signe ; si ce nombre tombe en dessous de certain seuil ($N_x \times N_y / 10\,000$ par exemple), alors l'algorithme s'arrête.

Empiriquement, on remarque que ce critère n'est pertinent qu'après un certain nombre d'itérations (car les premières itérées de w^h varient peu), c'est pourquoi on impose également un minimum de 1 100 itérations avant d'arrêter la boucle.

Seuillage de la solution Pour obtenir la carte de disparité u^h à partir de la solution w^h obtenue après N itérations de l'algorithme proposé, on applique cette fois le théorème 11. On seuille pour obtenir le niveau 0 de w^h , puis on reconstruit u^h à l'aide de la version discrète de la formule (3.6) :

$$\forall (i,j) \in \Omega^h, \quad u_{i,j}^h = h_t \sup \left\{ k \in \llbracket 0; N_t - 1 \rrbracket \mid w_{i,j,k}^h > 0 \right\}.$$

3.4 Détection et gestion de l'occultation

On suppose à présent que la carte de la disparité u^h a été estimée par l'un de deux algorithmes présentés à la section précédente. La carte est dense, c'est-à-dire qu'une disparité est attribuée à chaque pixel, qu'il soit en réalité occulté ou non. Or, la fonctionnelle d'énergie a été conçue spécifiquement pour permettre de distinguer les

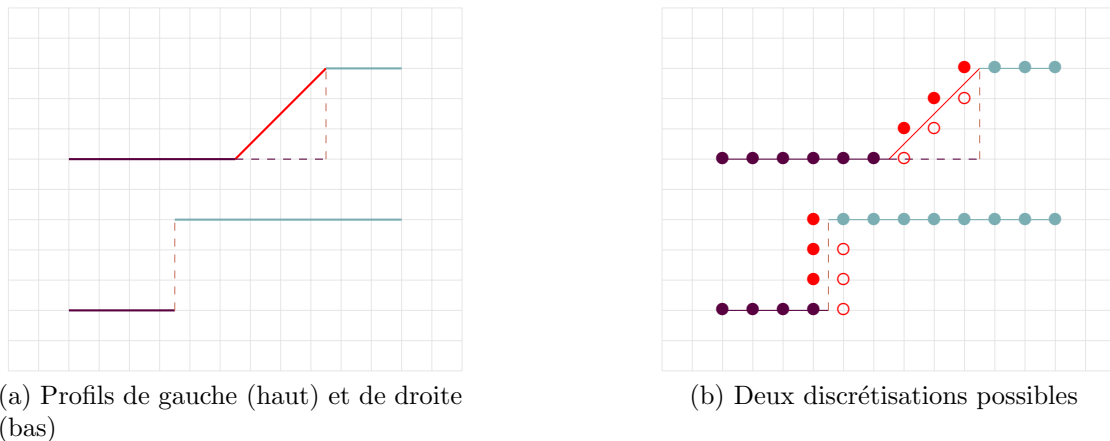


FIGURE 3.1 – Discrétisation et adhérence. Parce que les pixels et les disparités sont discrétisés, si la zone d’occultation n’est pas correctement placée (par rapport à la grille d’échantillonnage), deux profils sont possibles pour approcher le profil de gauche idéal recherché (points rouges pleins ou vides). Or, la première solution engendre un coût d’attache aux données potentiellement plus bas, car elle met en correspondance des pixels appartenant au même objet violet, donc *a priori* semblables (le coût de régularité étant le même dans les deux cas).

pixels occultés des autres. Nous allons donc décrire le traitement destiné à 1) détecter ces pixels ; 2) leur attribuer une disparité raisonnable.

3.4.1 Détection de l’occultation par saturation de la pente

On rappelle que la disparité n’est pas définie dans les zones occultées. Celle qui est estimée par la méthode étudiée dans ce chapitre n’est en réalité que le produit d’une interpolation affine (sur la ligne). L’analyse menée au chapitre 2 assure que ces pixels interpolés peuvent être distingués des autres. Les pixels de l’image de référence peuvent en effet être classés en deux groupes : ceux pour lesquels la dérivée horizontale de la disparité interpolée vaut 1, et les autres. Les premiers correspondent à des pixels occultés.

Ainsi, une manière de détecter l’occultation dans l’image de référence une fois que la disparité dense u^h a été estimée est de calculer la pente horizontale de la disparité, puis d’identifier tous les pixels pour lesquels cette pente vaut 1. Comme il s’agit de la pente maximale théorique, on parle de *saturation de la pente*. Le masque qui en résulte permet de localiser précisément les pixels occultés. Numériquement, on utilise comme approximation de la dérivée horizontale l’opérateur $\tilde{\delta}_x^h$ défini pour tout pixel $(i,j) \in \Omega^h$ par les différences finies

$$(\tilde{\delta}_x^h u^h)_{i,j} = \begin{cases} u_{i+1,j}^h - u_{i,j}^h & \text{si } i < N_x - 1 \\ 0 & \text{si } i = N_x - 1. \end{cases} \quad (3.18)$$

Dans le cas discret, une étude rapide montre que, si on utilise directement cette dérivée pour détecter les occultations, on introduit de manière systématique de l’adhérence sur les bords gauches des objets. Pour tenter de comprendre ce phénomène, intéressons-nous à la situation présentée à la figure 3.1. On y présente à gauche deux profils, qui correspondent aux profils de gauche (en haut) et de droite (en bas) d’une scène théorique. Pour distinguer les objets auxquels ils se réfèrent, chaque portion du profil est de couleur différente. Dans le profil de gauche, les pointillés correspondent à

des portions occultées (invisibles depuis la vue de droite), dans le profil de droite, les pointillés correspondent à une portion non visible sur l'image de droite. L'interpolation de la zone d'occultation du profil de gauche est proposé en rouge : il s'agit d'un segment de pente 1.

Sauf cas particulier, la zone d'occultation commence et termine en des points qui n'appartiennent pas à la grille Ω^h . Dans ce cas (figure 3.1, droite), puisque la disparité aussi est discrétisée, il existe deux moyens d'approcher le profil exact. Chacun de ces deux profils discrets induit une mise en correspondance différente des pixels. Dans le premier cas (points rouges pleins), tous les pixels de la zone occultée sont mis en correspondance avec le dernier point non occulté de l'objet violet (qui correspond ici à l'objet occulté) ; dans le second cas (points rouges vides), tous les pixels de la zone occultée sont mis en correspondance avec le premier point non occulté de l'objet bleu clair (qui correspond lui à l'objet occultant). Dans l'exemple étudié, les points occultés appartenant à l'objet violet, il semble raisonnable d'estimer que les pixels occultés ressemblent aux pixels non occultés de cet objet, et sont très différents des pixels de l'objet bleu clair. La première solution produit donc un coût de mise en corespondance *a priori* beaucoup plus bas que la seconde solution, qui met en correspondance tous les points de la région occultée avec un pixel très différent. Par ailleurs, dans les deux cas, le coût de régularisation est le même. La première solution sera donc préférentiellement choisie ; or, elle introduit de l'adhérence (d'une largeur 1).

Pour corriger ce biais pratiquement systématique, on choisit de translater artificiellement les zones occultées d'un pixel vers la droite. On est donc amené à définir le masque d'occultation de la manière suivante

$$\forall (i,j) \in \Omega^h, \quad M_{i,j}^{\text{occ}} = \begin{cases} 1 & \text{si } (\tilde{\delta}_x^h u^h)_{i,j-1} \geq 1 \\ 0 & \text{si } (\tilde{\delta}_x^h u^h)_{i,j-1} < 1 \end{cases} \quad (3.19)$$

qui vaut 1 en cas d'occultation et 0 sinon.

3.4.2 Densification des zones occultées

Une fois les zones occultées localisées, on va y modifier la valeur de la disparité, dont on sait qu'en l'état, elle n'est pas significative. Évidemment, la disparité ne peut pas être connue dans ces régions, mais on peut toutefois en proposer une interpolation raisonnable.

On se base pour cela sur le modèle de scène que nous avons déjà utilisé pour analyser le phénomène d'occultation : un objet est partiellement occulté par un objet qui se situe devant lui, et on suppose que, localement, les objets ont une disparité constante. Sans d'autres informations, on peut donc faire l'hypothèse que la partie occultée est de disparité constante, qui se prolonge (au moins dans un voisinage) en dehors de la zone d'occultation.

L'occultation se produisant (dans l'image de référence) sur les bords *gauches* des objets, on en déduit que le premier point non occulté situé à *gauche* de la région occultée n'est pas sur l'objet occultant. Il s'agit donc *a priori* d'une partie non occultée de l'objet occulté. Par conséquent, la disparité de la région occultée la plus raisonnable est donnée par celle de ce premier point non occulté. On interpole ainsi la carte de disparité en diffusant (horizontalement) vers la droite la disparité connue dans les régions occultées. Plus précisément, l'interpolation \tilde{u}^h s'écrit

$$\forall (i,j) \in \Omega^h, \quad \tilde{u}_{i,j}^h = \begin{cases} u_{i,j}^h & \text{si } M_{i,j}^{\text{occ}} = 0 \\ u_{i_0,j}^h & \text{si } M_{i,j}^{\text{occ}} = 1 \text{ et } i_0 = \max\{i' \leq i \mid M_{i',j}^{\text{occ}} = 0\}. \end{cases} \quad (3.20)$$

On retrouve cette démarche dans [13, 14].

3.5 Résultats expérimentaux

On présente dans cette section les résultats obtenus avec d’une part l’algorithme sur-relaxé et d’autre part l’algorithme accéléré proposé dans la section précédente. Les tests sont effectués sur les paires du banc d’essai Middlebury (voir 2.3.4) et sont évalués à partir des vérités-terrains fournies, ainsi que des cartes d’occultation modifiées selon la méthode décrite au paragraphe suivant, le cas échéant.

3.5.1 Modification des cartes d’occultation

Les cartes d’occultation proposées par le banc d’essai Middlebury présentent deux défauts majeurs, qui sont

1. la suppression des occultations de largeur 1 ;
2. l’introduction des occultations hors-champ. En effet, certains pixels de l’image de référence ne possèdent pas de pixel homologue dans la vue de droite non pas parce qu’ils ont été occultés par un objet occultant, mais parce que leur pixel homologue est hors-cadre dans l’image de droite. Or ceux-ci n’obéissent pas aux mêmes lois que les pixels réellement occultés et ne devraient donc pas être pris en compte dans la carte des occultations (mais plutôt être considérés comme des pixels dont la disparité ne peut être évaluée).

Nous proposons donc ici de modifier les cartes proposées par Middlebury de sorte de supprimer ces deux écueils.

Réintroduction des occultations de largeur 1 Pour réintroduire les occultations fines qui ont été supprimées, on commence par supposer que la contrainte de préservation d’ordre est satisfaite pour les paires considérées. Cette hypothèse nous permet d’utiliser l’analyse du phénomène d’occultation proposée au chapitre 2 et de déduire de la vérité-terrain dense une première carte des occultations (figure 3.2, en rouge dans la colonne de gauche). En effet, chaque discontinuité de disparité entraîne sur sa gauche une occultation, dont la largeur est exactement égale au saut de disparité.

Malheureusement, les scènes de Middlebury ne vérifient pas toujours la contrainte de préservation de l’ordre, en particulier Tsukuba (bras de la lampe) et Cones (pinceaux et sommets des cones), ce qui conduit à une sur-détection des occultations. Pour raffiner ces premières cartes, on choisit alors de tester l’injectivité de la vérité-terrain dans les zones déclarées occultées à la première étape. En d’autres termes, on teste pour chacun des pixels déclarés occultés si, d’après la vérité-terrain, d’autres pixels ont été mis en correspondance avec son pixel homologue. Si c’est le cas, le pixel est définitivement considéré comme occulté (figure 3.2, en rouge dans la colonne de droite).

Suppression des occultations hors-champ Pour détecter les occultations qui sont dues à la restriction des images sur un domaine fini Ω , on teste pour chaque pixel si son pixel homologue est bien dans le domaine de l’image de droite. Si ce n’est pas le cas, il est considéré comme les occultations hors-champ (figure 3.2, en cyan).

Les nouvelles cartes d’occultation obtenues sont présentées à la figure 3.2. Sauf mention contraire, ce sont elles qui sont dorénavant considérées comme la vérité-terrain des cartes d’occultation.

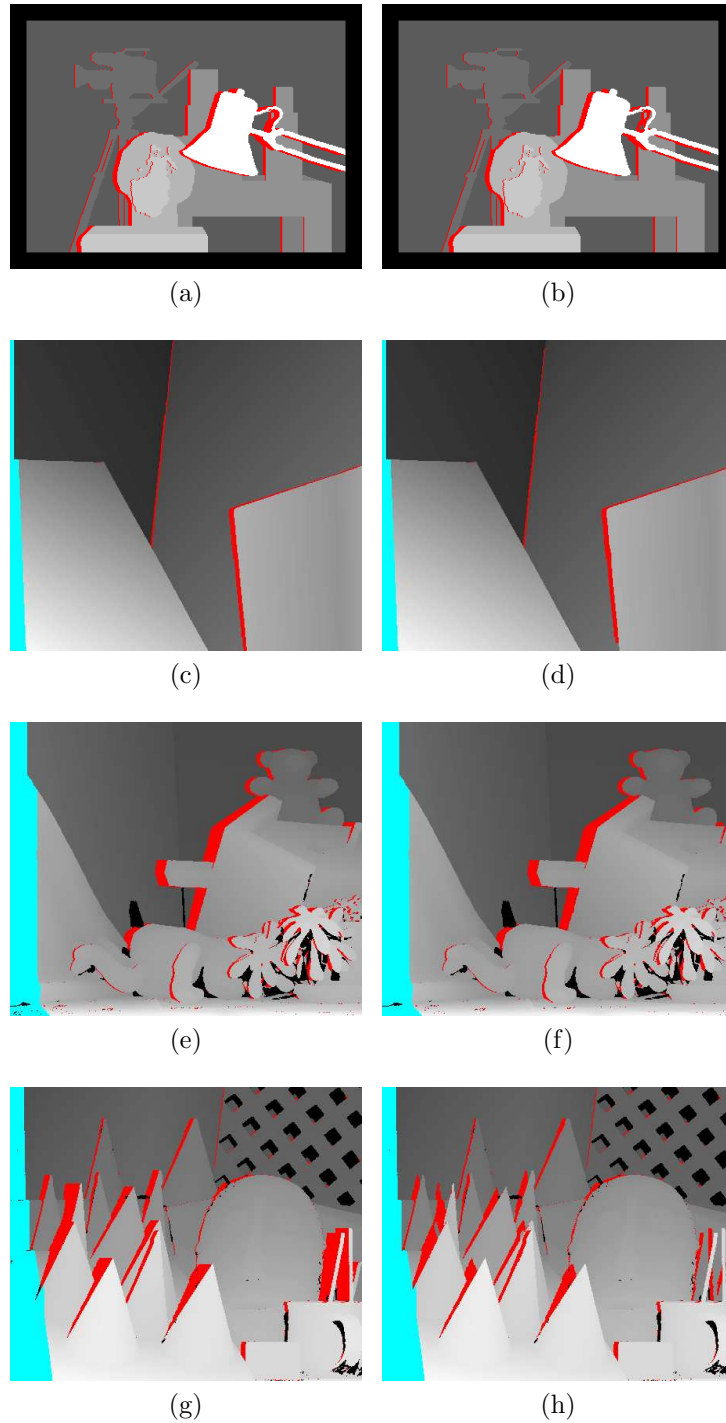


FIGURE 3.2 – Modification des cartes d'occultation. Colonne de gauche : premières cartes obtenues à partir des discontinuités de la vérité-terrain. Colonne de droite : raffinement calculé grâce à la consistance gauche-droite de la vérité-terrain. En rouge : les pixels occultés, en cyan les occultations hors-champ. De haut en bas : Tsukuba, Venus, Teddy et Cones.

3.5.2 Récapitulatif de l'algorithme

On rappelle que la méthode décrite plus haut se décompose en deux étapes :

1. résolution par algorithme primal-dual sur-relaxé du problème convexe (3.13) (désigné sous le nom de **algorithme sur-relaxé**), puis seuillage à $s = 0,9$; on obtient alors une première carte de disparité u^h ;
- 1bis ou résolution par algorithme primal-dual du problème fortement convexe (3.14) (désigné sous le nom de **algorithme accéléré**), puis seuillage à $s = 0$; on obtient également une première carte de disparité u^h ;
2. détection des zones occultées par saturation de la pente et densification par diffusion (vers la droite) de la disparité.

Dans ce qui suit, les paramètres de la méthode sont fixés comme suit :

- paramètre de pondération des termes d'attache aux données et de régularisation $\mu = 50/255$ [12] ;
- paramètres du terme d'attache aux données : $a = 100$ dans la définition du coefficient variable α^h , paramètre du lissage ROF $\lambda = 1/50$, convolution avec une gaussienne d'écart-type 8 et de support de taille 9×9 ;
- pas de discrétisation de la disparité $h_t \in \{1,0,5\}$.

Les cartes de disparités générées seront en particulier précises au pixel ou au demi-pixel près, suivant le choix de h_t .

3.5.3 Première carte de disparité

On présente dans ce paragraphe les résultats obtenus à l'issue de la première étape de l'algorithme, qui est l'estimation dense de la disparité obtenue par algorithme primal-dual.

Volume de coût Pour calculer le volume de coût g^h , il faut calculer pour chaque triplet d'indices (i,j,k) la quantité

$$g_{i,j,k}^h = (1 - \alpha_{i,j}^h) \|(I_L^h)_{i,j} - (I_R^{\text{interp}})_{i,j-kh_t}\| + \alpha_{i,j}^h \|(\tilde{\nabla}^h I_L^h)_{i,j} - (\tilde{\nabla}^h I_R^{\text{interp}})_{i,j-kh_t}\|$$

où l'opérateur $\tilde{\nabla}^h$ est défini par

$$\forall (i,j) \in \Omega^h, \quad (\tilde{\nabla}^h I^h)_{i,j} = \begin{pmatrix} (\delta_x^h I^h)_{i,j} \\ (\delta_y^h I^h)_{i,j} \end{pmatrix}$$

Les variations horizontales $\delta_x^h I^h$ sont données par la formule (3.18) et les variations verticales δ_y^h sont définies de manière analogue par les différences finies

$$(\delta_y^h I^h)_{i,j} = \begin{cases} I_{i,j+1}^h - I_{i,j}^h & \text{si } j < N_y - 1 \\ 0 & \text{si } j = N_y - 1. \end{cases}$$

L'image interpolée I_R^{interp} est obtenue par interpolation B-spline d'ordre 5 [16, 15] de l'image échantillonnée I_R^h .

La pondération variable α^h est définie par

$$\forall (i,j) \in \Omega^h, \quad \alpha_{i,j}^h = \frac{1}{1 + G \star \|(\tilde{\nabla}^h I_L^{\text{ROF}})\|_{i,j}^2 / a}$$

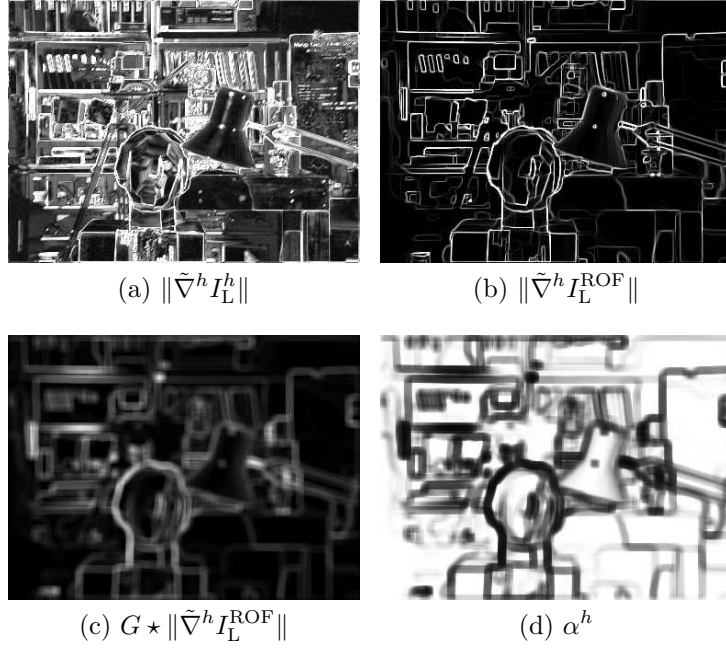


FIGURE 3.3 – Amplitude du gradient dans la construction de α^h , exemple de la paire Tsukuba. (a) Amplitude du gradient de l’image initiale I_L^h : trop de détails sont apparents, qui sont principalement dus à la texture des objets. (b) Amplitude du gradient de l’image lissée I_L^{ROF} : le lissage ROF permet de supprimer les variations non significatives et de conserver celles qui correspondent plus vraisemblablement à des discontinuités de scène. (c) Filtrage de $\|\tilde{\nabla}^h I_L^{\text{ROF}}\|$ par une gaussienne : ce lissage permet d’étaler spatialement les forts gradients. Il est en effet essentiel que les voisinages de discontinuités soient traités comme des discontinuités. (d) Pondération α^h résultant : les discontinuités et leur voisinage correspondent à des valeurs faibles de α^h . La comparaison y sera donc majoritairement basée sur l’intensité, et non le gradient.

où l’image lissée I_L^{ROF} est obtenue en résolvant le problème

$$\min_{I^h \in \mathbb{R}^{3N_x N_y}} \frac{\lambda}{2} \|I_L^h - I^h\|_2^2 + \text{TV}(I^h) \quad (3.21)$$

(voir chapitre 6 pour plus d’informations au sujet du débruitage ROF). On choisit $\lambda = 1/50$. L’intérêt d’utiliser une version lissée de l’image initiale est d’éliminer les variations non significatives. On choisit pour cela la régularisation ROF car elle préserve les discontinuités importantes. La gaussienne utilisée pour la convolution est quant à elle choisie centrée d’écart-type 8. Elle permet d’obtenir des valeurs de α^h faibles au voisinage des discontinuités. On pourra visualiser l’influence du lissage ROF et de la convolution par la gaussienne sur la construction de α^h à la figure 3.3.

On a ainsi construit le volume de coût g^h , de dimension $N_x \times N_y \times N_t$, dont on peut visualiser une tranche à la figure 3.4(a). Sans régularisation, la minimisation revient à effectuer un WTA, c’est-à-dire de prendre pour chaque pixel la disparité impliquant un coût de corrélation minimal :

$$\forall (i,j) \in \Omega^h, \quad u_{i,j}^{\text{WTA}} = h_t \underset{k \in \llbracket 0; N_t \rrbracket}{\operatorname{argmin}} g_{i,j,k}^h$$

dont le résultat est présenté à la figure 3.4(b).

Nombre d’itérations Pour stopper les itérations, on choisit

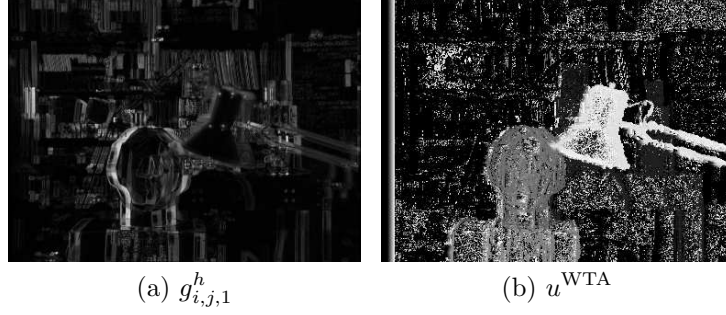


FIGURE 3.4 – Volume de coût g^h associé à la paire Tsukuba. (a) Tranche 1 du volume g^h . Plus le pixel est clair, plus le coût de corrélation est important. (B) WTA (*Winner-Takes-All*) : pour chaque pixel de l'image de référence, on retient la disparité qui minimise le coût de corrélation.

| Paire | Tsukuba | Venus | Teddy | Cones |
|---------------------------------------|---------|---------|---------|---------|
| Algorithme sur-relaxé ($h_t = 1$) | 700 | 1 390 | 3 770 | 4 020 |
| Algorithme accéléré ($h_t = 1$) | 1 490 | 1 520 | 3 350 | 2 370 |
| Algorithme classique ($h_t = 1$) | 1 350 | 2 710 | 7 350 | 7 830 |
| Algorithme sur-relaxé ($h_t = 0,5$) | 2 470 | 5 780 | 10 000+ | 10 000+ |
| Algorithme accéléré ($h_t = 0,5$) | 1 650 | 2 050 | 5 920 | 4 510 |
| Algorithme classique ($h_t = 0,5$) | 4 800 | 10 000+ | 10 000+ | 10 000+ |

FIGURE 3.5 – Nombre d'itérations obtenu grâce au critère d'arrêt. L'*algorithme classique* désigne une variante de l'algorithme sur-relaxé, sans la sur-relaxation globale ($\rho = 1$). Le nombre d'itérations maximal est limité à 10 000 pour chaque expérience.

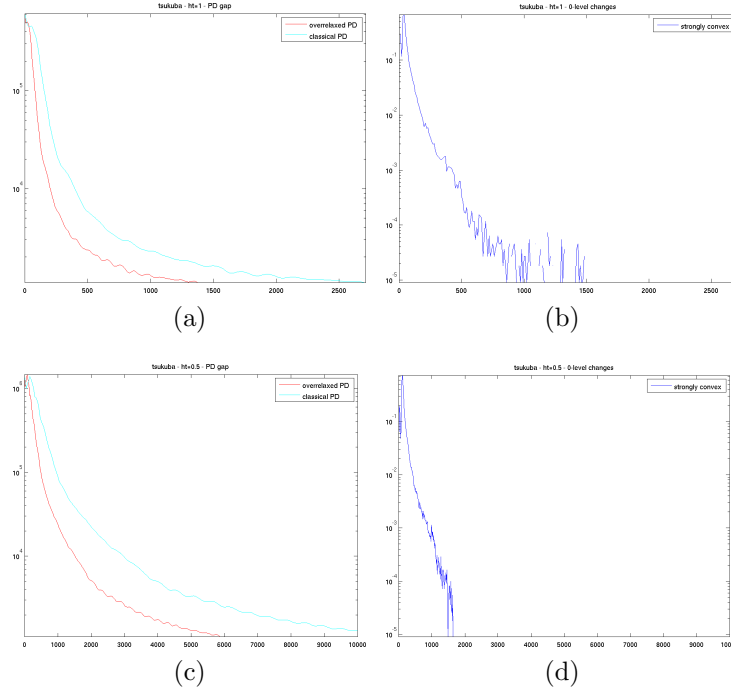


FIGURE 3.6 – Comparaison des *gaps* pour les algorithmes primaux-duaux présentés dans ce chapitre (l'échelle est logarithmique pour l'ordonnée), exemple de la paire Tsukuba. (a) et (c) Algorithmes sur-relaxé et classique. (b) et (d) Algorithme accéléré. Ligne du haut : précision pixellique. Ligne du bas : précision sous-pixellique.

| Paire | Tsukuba | | | | Venus | |
|---|---------|-------|--------|-------|-------|--------|
| Précision pixellique ($h_t = 1$) | | | | | | |
| Algorithme sur-relaxé | 9,92% | 2,44% | 9,92% | 4,08% | 3,34% | 22,44% |
| Algorithme accéléré | 11,73% | 3,40% | 11,73% | 5,01% | 3,52% | 23,01% |
| Précision sous-pixellique ($h_t = 0,5$) | | | | | | |
| Algorithme sur-relaxé | 6,26% | 2,51% | 23,33% | 3,64% | 2,80% | 12,72% |
| Algorithme accéléré | 7,18% | 3,05% | 23,21% | 3,62% | 2,83% | 12,94% |

| Paire | Teddy | | | | Cones | |
|---|--------|--------|--------|--------|-------|--------|
| Précision pixellique ($h_t = 1$) | | | | | | |
| Algorithme sur-relaxé | 14,38% | 11,57% | 42,66% | 11,18% | 8,91% | 35,22% |
| Algorithme accéléré | 14,80% | 11,87% | 42,73% | 10,65% | 8,38% | 34,10% |
| Précision sous-pixellique ($h_t = 0,5$) | | | | | | |
| Algorithme sur-relaxé | 11,97% | 9,60% | 26,58% | 8,01% | 6,48% | 19,14% |
| Algorithme accéléré | 12,00% | 9,51% | 25,80% | 8,18% | 6,67% | 18,15% |

FIGURE 3.7 – Erreur d’estimation dans la première estimation (dans les zones non occultées). Pour chaque paire, on considère l’erreur pixellique, l’erreur Middlebury et l’erreur sous-pixellique.

1. pour l’algorithme sur-relaxé : d’arrêter l’algorithme lorsque le *primal-dual gap* tombe en-dessous de $N_x \times N_y \times N_t / 1\,000$ (ce qui revient à tolérer un *gap* moyen de l’ordre du millièrme) ;
2. pour l’algorithme accéléré : de mesurer la stabilité du niveau zéro de la variable primale et d’arrêter l’algorithme lorsque le nombre de coefficients changeant de signe tombe en dessous de $N_x \times N_y / 10\,000$.

Le tableau 3.5 affiche le nombre d’itérations nécessaires pour activer le critère d’arrêt pour chaque algorithme et pour deux précisions différentes ($h_t = 1$ pour la précision pixellique et $h_t = 0,5$ pour la précision sous-pixellique). La figure 6.6 présente quant à elle l’évolution du critère mesuré pour l’arrêt (*primal-dual gap* et nombre de coefficients changeant de signe).

Cartes obtenues On présente les résultats obtenus sur les paires de Middlebury aux figures 3.8 et 3.9. Grâce à la vérité terrain, on peut déterminer précisément les estimations erronées, qui sont répertoriés dans le tableau 3.7. On s’intéresse aux erreurs pixelliques (supérieures ou égales au pixel) localisées en dehors des zones occultées ou non renseignées de la scène, ainsi qu’aux erreurs strictement supérieure à 1 (erreur Middlebury) et aux erreurs sous-pixelliques (supérieure ou égale à 0,5 px). L’erreur Middlebury est naturellement la plus faible, et l’erreur sous-pixellique la plus grande.

3.5.4 Traitement des occultations

Une fois la première estimation de la disparité obtenue, on procède au traitement des occultations en suivant la méthode présentée dans la section 3.4.

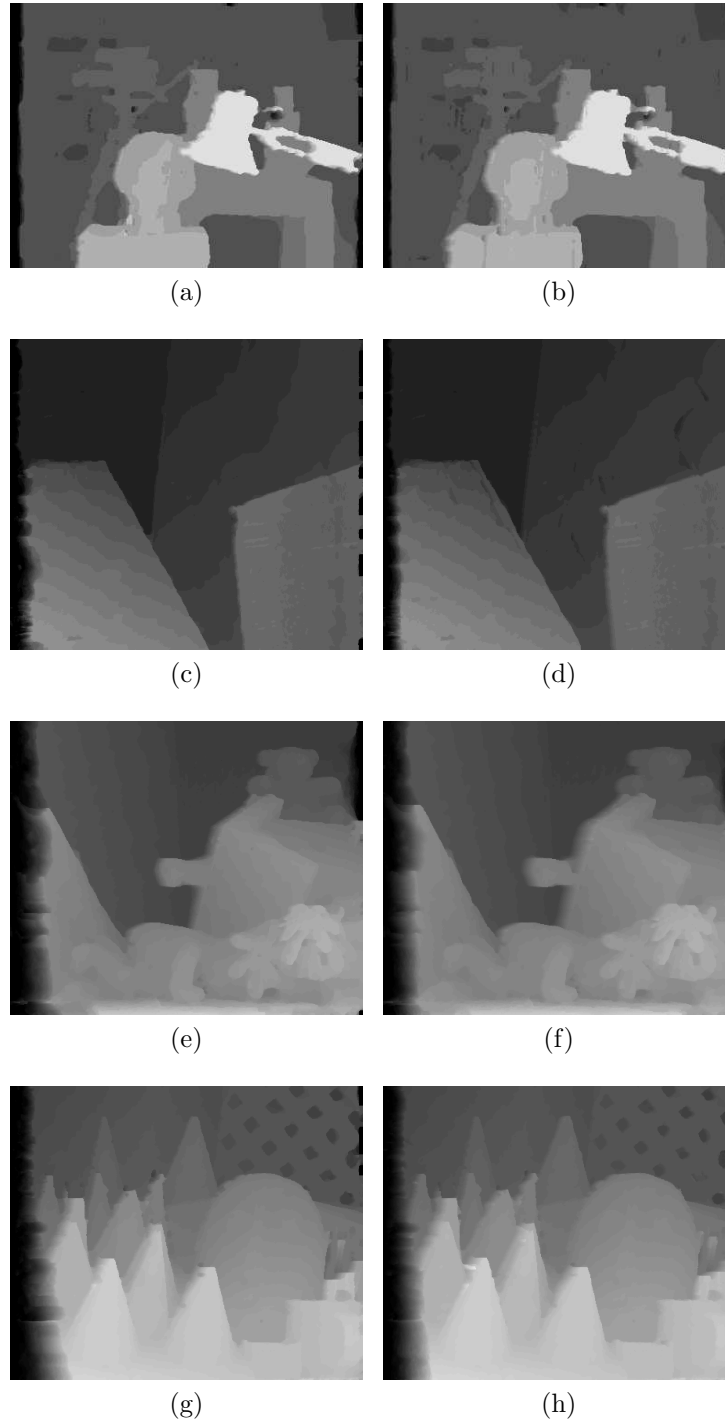


FIGURE 3.8 – Première estimation de la carte de disparité, précision **pixellique**. Colonne de gauche : algorithme sur-relaxé. Colonne de droite : algorithme accéléré. De haut en bas : Tsukuba, Venus, Teddy et Cones.

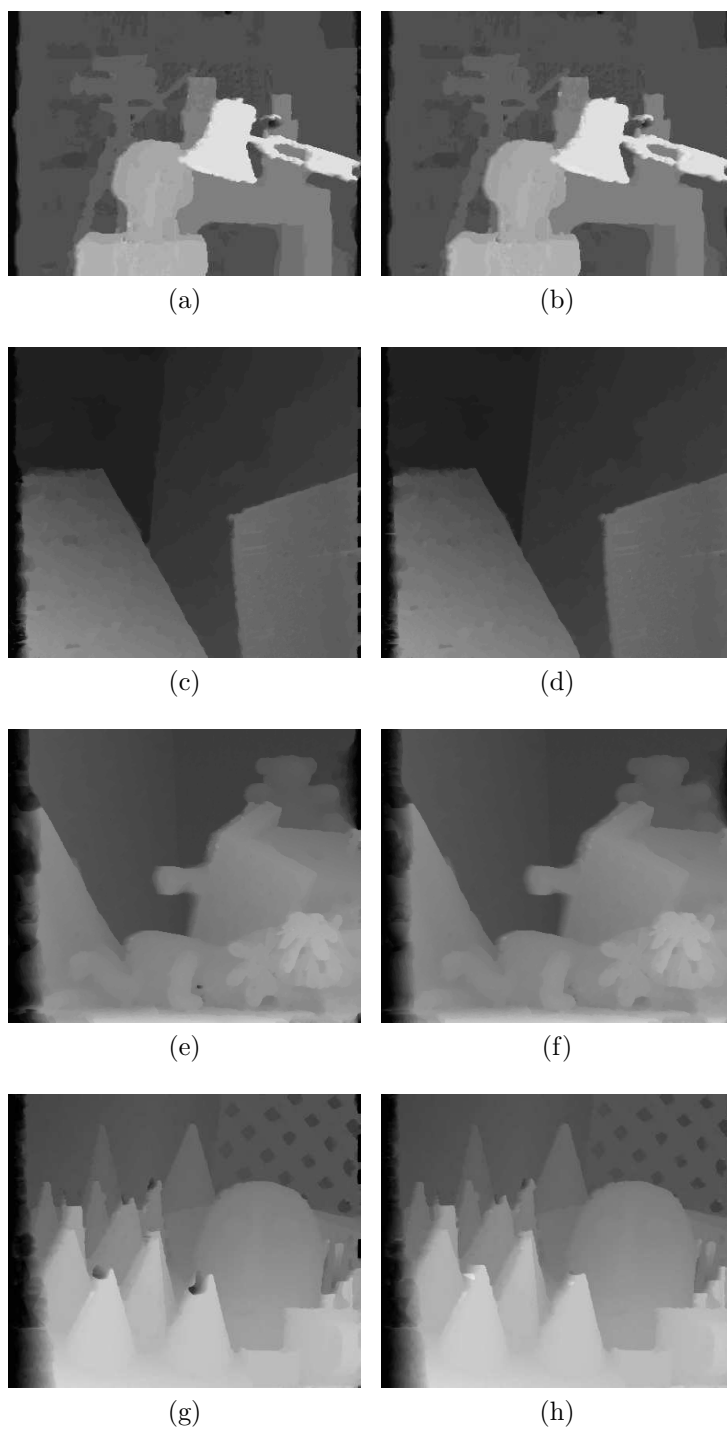


FIGURE 3.9 – Première estimation de la carte de disparité, précision **sous-pixellique**. Colonne de gauche : algorithme sur-relaxé. Colonne de droite : algorithme accéléré. De haut en bas : Tsukuba, Venus, Teddy et Cones.

Précision et rappel Pour évaluer la détection des occultations par notre méthode, on commence par mesurer le nombre de

1. vrais positifs (TP) : occultations correctement détectées comme telles ;
2. vrais négatifs (TN) : pixels non occultés d’après la vérité-terrain qui n’ont pas été détectés comme occultés ;
3. faux positifs (FP) : pixels non occultés selon la vérité-terrain mais détectés comme occultés par notre méthode ;
4. faux négatifs (FN) : pixels occultés selon la vérité-terrain mais non détectés comme tels.

On rappelle qu’on utilise comme vérités-terrains les cartes d’occultation générées selon le paragraphe 3.5.1. On remarquera que la somme TP + FN donne le nombre de pixels occultés d’après la vérité-terrain, tandis que la somme TP + FP donne le nombre de pixels occultés d’après le détecteur.

Pour mesurer les performances de la détection, on utilise deux mesures appelées *précision* et *rappel*. La précision est définie par le rapport $TP/(TP + FP)$ tandis que le rappel est défini par le rapport $TP/(TP + FN)$. Plus ces deux taux sont proches de 1, meilleure est la détection. La précision mesure à quel point la détection est concentrée dans les zones à détecter : en effet, elle est élevée s’il y a peu de faux positifs. Ainsi, même si peu de détections sont faites, ce score peut être grand si, parmi elles, un grand nombre correspond à des détections correctes. Au contraire, si la méthode a tendance à sur-détecter les occultations, ce score peut être faible même si toutes les occultations ont été détectées. Le rappel ne s’intéresse quant à lui qu’à l’efficacité de la détection dans les zones occultées selon la vérité-terrain. Peu importe la détection en-dehors de ces zones, le rappel sera élevé si un grand nombre de détections sont correctes.

Détection par saturation de la pente On présente dans les figures 3.10 – 3.11 (colonne de gauche) les régions détectées par le masque (3.19), c’est-à-dire les pixels saturant la pente horizontale de la disparité. Chaque détection est marquée en rouge quand il s’agit d’un vrai positif et en cyan quand il s’agit d’un faux positif. Les points jaunes correspondent aux points non détectés, c’est-à-dire aux faux négatifs.

Densification Une fois les occultations détectées, on interpole la carte de disparité en diffusant (sur la ligne) la disparité des zones non occultées vers les zones occultées. Cette diffusion se fait vers la droite, en suivant la formule (3.20) (figures 3.12 – 3.13, colonne de gauche).

Amélioration de la détection La saturation de la pente peut s’avérer insuffisante pour détecter correctement les occultations. Parmi les causes des détections incorrectes, on peut citer le fait que, lorsque le bord de l’objet n’est pas vertical, mais incliné, localiser exactement les occultations implique d’introduire dans la carte de disparité des discontinuités qui suivent le bord de l’objet, ici incliné. Ce genre de discontinuité conduit à une variation totale plus importante, puisqu’il faut tenir compte à la fois des variations horizontales et verticales. Une manière de réduire le coût de régularisation consiste à déplacer localement les zones occultées, de sorte de les aligner d’une ligne à l’autre. On observe alors des motifs en escalier qui paraissent artificiels. Une autre cause d’erreur peut survenir dans le cas de la précision sous-pixellique. En effet, dans ce cas-là, l’occultation est détectée lorsqu’il y a un saut de disparité d’un pixel, alors que la carte peut présenter des sauts d’un demi-pixel. Un coût de mise en correspondance plus

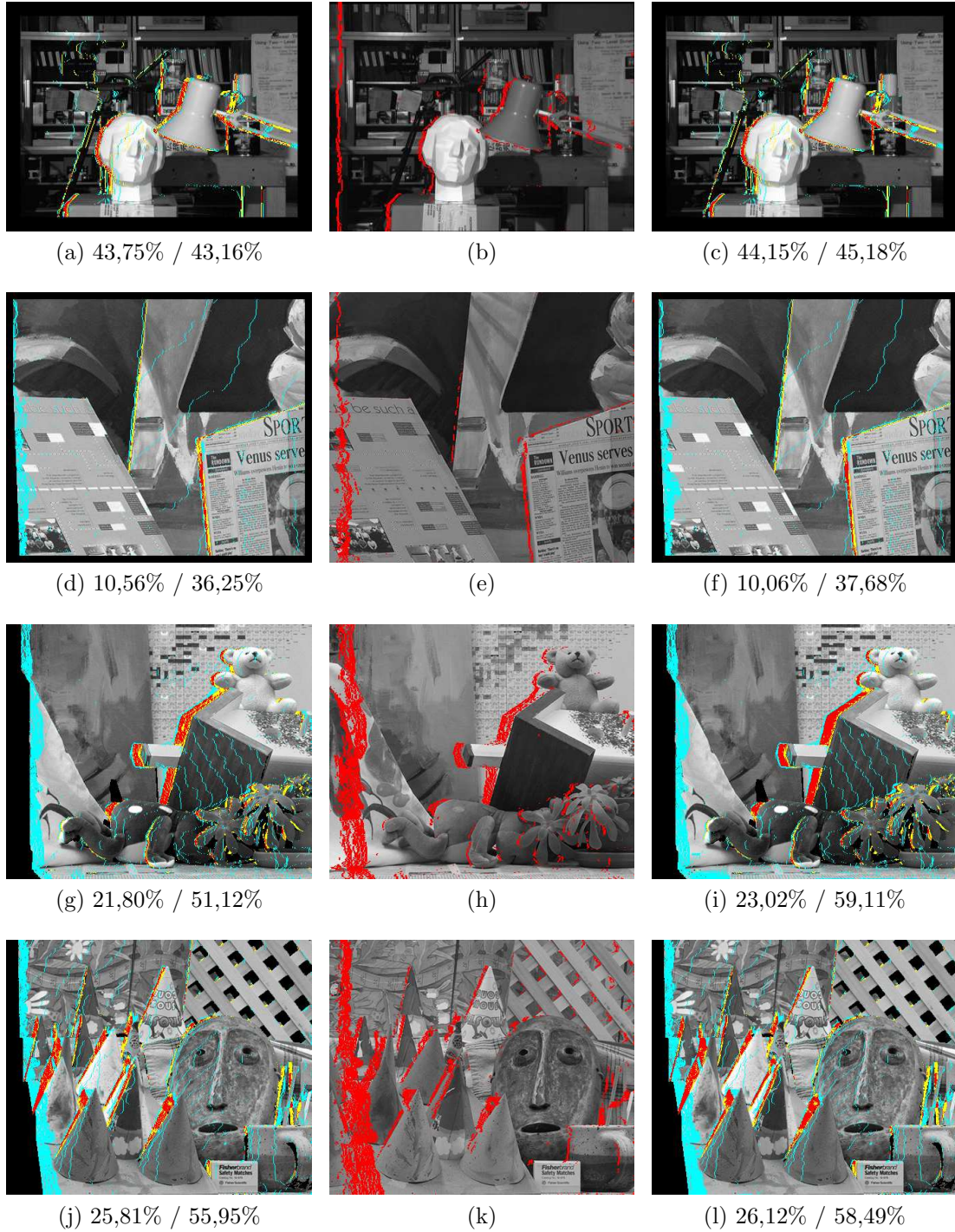


FIGURE 3.10 – Traitement des occultations, précision **pixelique**. Cas de l’algorithme **sur-relaxé**. Colonne de gauche : détection des occultations par saturation de la pente. Colonne du milieu : suppression des détections de largeur 1. Colonne de droite : amélioration des détections. En rouge, les détections correctes (vrais positifs, TP), en jaune, les détections manquantes (faux négatifs, FN) et en cyan, les détections incorrectes (faux positifs, FP). En légende : les taux de précision et de rappel. De haut en bas : Tsukuba, Venus, Teddy et Cones.

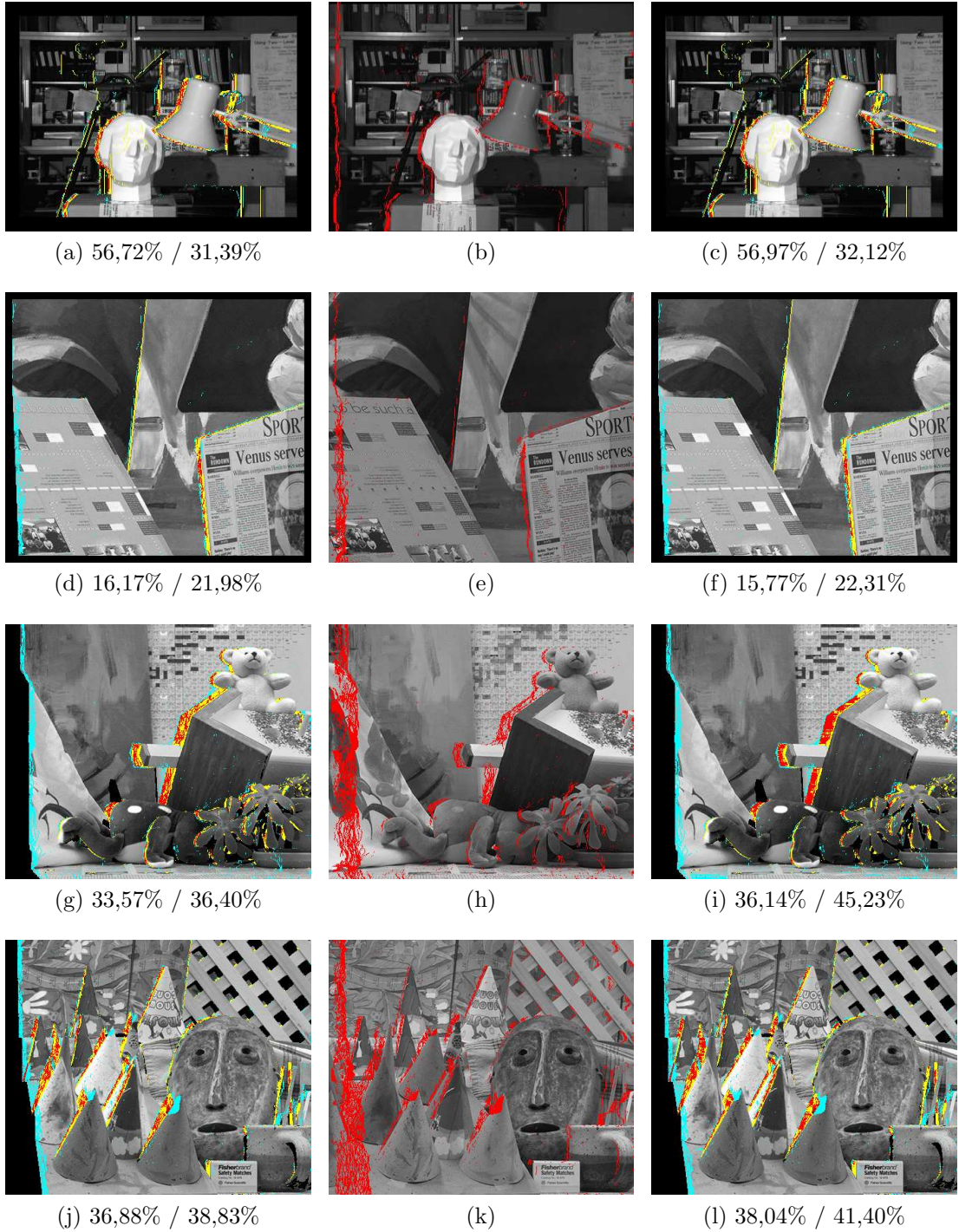


FIGURE 3.11 – Traitement des occultations, précision **sous-pixellique**. Cas de l’algorithme **sur-relaxé**. Colonne de gauche : détection des occultations par saturation de la pente. Colonne du milieu : suppression des détections de largeur 1. Colonne de droite : amélioration des détections. En rouge, les détections correctes (vrais positifs, TP), en jaune, les détections manquantes (faux négatifs, FN) et en cyan, les détections incorrectes (faux positifs, FP). En légende : les taux de précision et de rappel. De haut en bas : Tsukuba, Venus, Teddy et Cones.

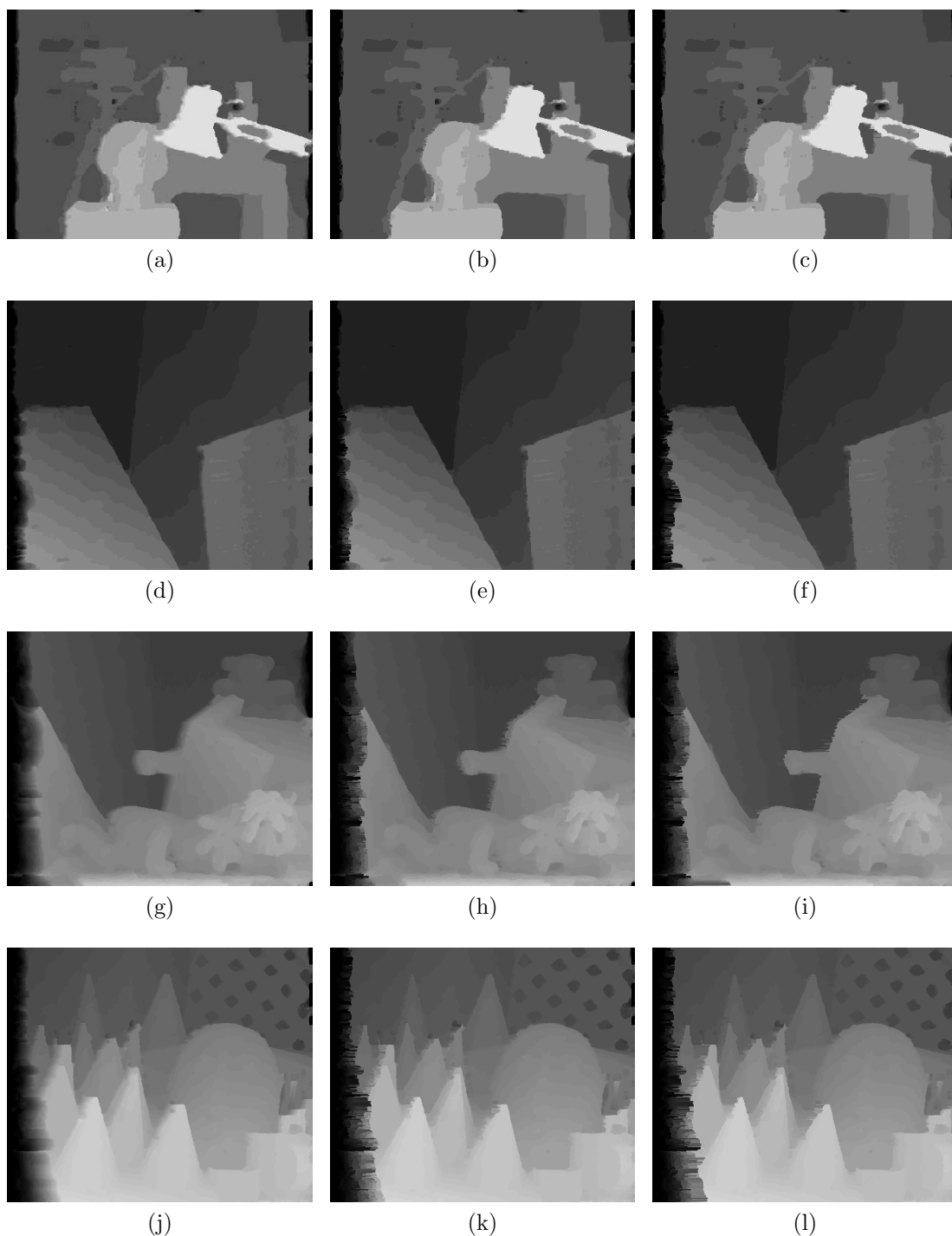


FIGURE 3.12 – Densification des occultations, précision **pixelique**. Cas de l’algorithme **sur-relaxé**. Colonne de gauche : carte dense initiale. Colonne du milieu : densification à partir de la première détection des occultations par saturation de pente. Colonne de droite : amélioration des détections. De haut en bas : Tsukuba, Venus, Teddy et Cones.

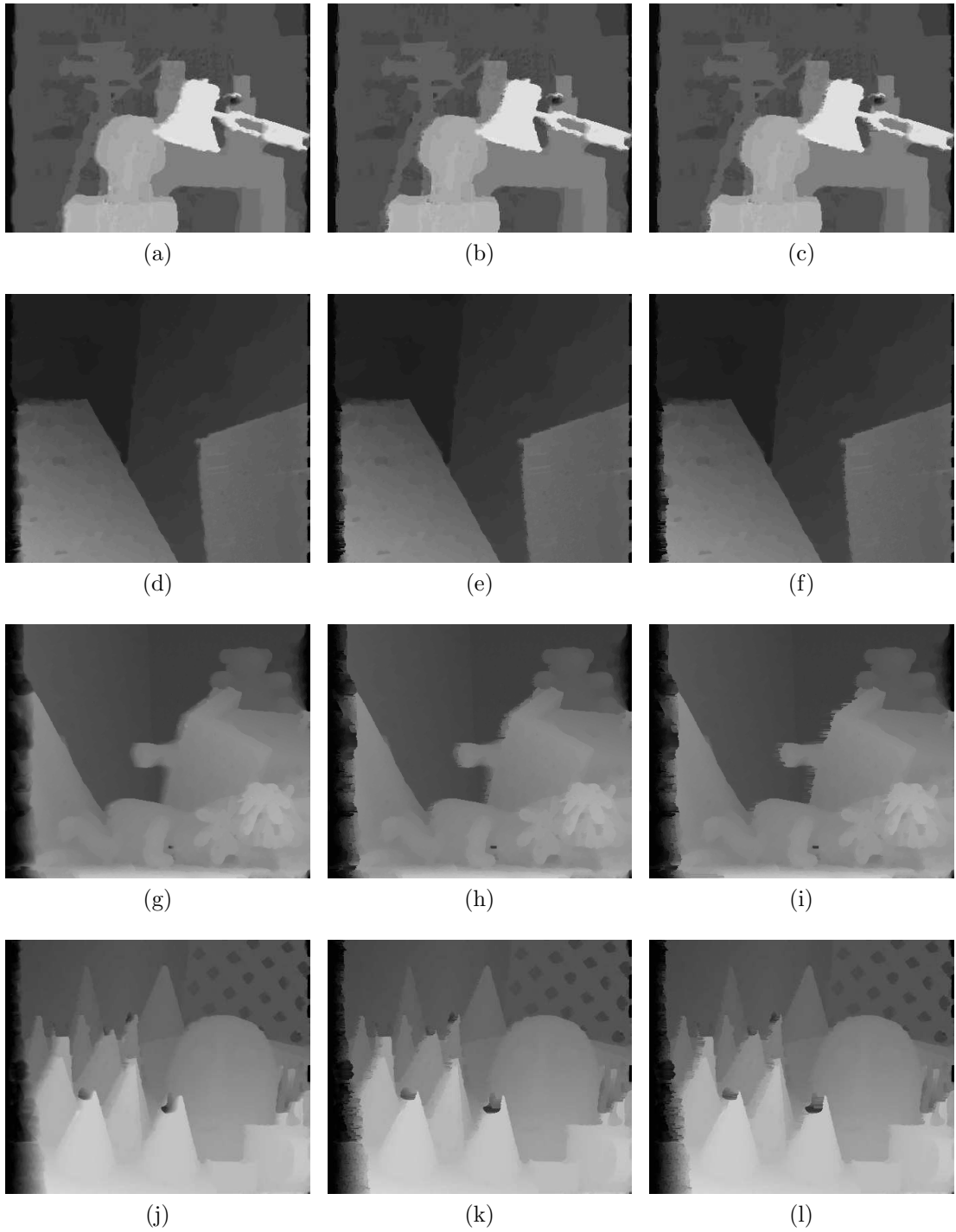


FIGURE 3.13 – Densification des occultations, précision **sous-pixelique**. Cas de l'algorithme **sur-relaxé**. Colonne de gauche : carte dense initiale. Colonne du milieu : densification à partir de la première détection des occultations par saturation de pente. Colonne de droite : amélioration des détections. De haut en bas : Tsukuba, Venus, Teddy et Cones.

favorable peut alors conduire à préférer dans les zones occultées deux sauts disjoints d'un demi-pixel à un saut plus franc d'un pixel, qui ne sont alors pas alors détectés comme occultation. Dans ces deux cas, ces artéfacts conduisent à créer, à l'intérieur des régions occultées, les zones où la disparité est constante. Les pixels correspondants ne saturant pas la contrainte de pente, ils ne sont pas détectés comme pixels occultés. Ainsi, les zones détectées comme occultées présentent parfois des trous : elles ne sont pas denses.

Pour densifier ces régions, on propose un post-traitement supplémentaire. Il consiste à déclarer occulté tout pixel qui serait situé dans un tel trou. Pour détecter ces pixels, on suppose que si un pixel est situé entre deux pixels occultés du même objet, alors il est vraisemblablement lui-même occulté. La difficulté est principalement de décider si deux points appartiennent au même objet, sans connaître au préalable la disparité de la scène. Nous proposons ici d'utiliser la version I_L^{ROF} lissée par le modèle ROF de l'image de référence. Si la distance entre deux intensités $I_L^{\text{ROF}}(p)$ et $I_L^{\text{ROF}}(p')$ (pour la norme euclidienne de \mathbb{R}^3) est inférieure à un seuil $T = 8$, alors les deux pixels sont déclarés appartenant au même objet.

Le post-traitement proposé se décrit donc ainsi : pour tout pixel non occulté p , on recherche dans l'intervalle $[p - h_t(N_t - 1); p + h_t(N_t - 1)]$ les voisins à gauche et à droite (sur la même ligne) occultés les plus proches, s'ils existent, de p :

$$p_1 = \underset{\substack{p' \in p - h_t \llbracket 1; N_t - 1 \rrbracket \\ M_{p'}^{\text{occ}} = 1}}{\text{argmin}} \|p - p'\| \quad \text{et} \quad p_2 = \underset{\substack{p' \in p + h_t \llbracket 1; N_t - 1 \rrbracket \\ M_{p'}^{\text{occ}} = 1}}{\text{argmin}} \|p - p'\|$$

étant entendu que la largeur d'une occultation ne peut excéder $h_t N_t$ pixels. On vérifie ensuite que p , p_1 et p_2 appartiennent au même objet, en définissant

$$M_{p,p'}^{\text{objet}} = \begin{cases} 1 & \text{si } \|I_L^{\text{ROF}}(p) - I_L^{\text{ROF}}(p')\| \leq T \\ 0 & \text{sinon.} \end{cases}$$

Si c'est le cas, alors p est déclaré occulté :

$$\tilde{M}_p^{\text{occ}} = \begin{cases} 1 & \text{si } M_p^{\text{occ}} = 1 \\ 1 & \text{si } M_p^{\text{occ}} = 0 \text{ et } (p_1, p_2) \text{ existent, avec } M_{p,p_1}^{\text{objet}} M_{p,p_2}^{\text{objet}} = 1 \\ 0 & \text{sinon.} \end{cases}$$

Dans le cas de la précision pixelique, on travaille sur une carte d'occultation où les occultations de largeur 1 ont été supprimées au préalable, car celles-ci sont généralement le fruit de la quantification des cartes de disparité (figures 3.10 – 3.11, colonne du milieu).

Le résultat de ce traitement est présenté aux figures 3.10 – 3.11 (colonne de droite). Une fois la carte des occultations améliorée, on peut à nouveau densifier la carte de disparité en diffusant la disparité vers la droite. Le résultat de cette densification est présentée dans la colonne de droite de la figure 4.12. Le tableau 3.14 réunit les taux de précision et de rappel pour chaque expérience, avant et après le traitement des détections des zones occultées. Le tableau 3.15 présente pour les deux détections les erreurs d'estimation dans les zones occultées d'après la vérité-terrain, obtenues à l'issue de la densification. Elles sont comparées aux scores initiaux.

| Paire | | Tsukuba | |
|--|-----------------|-----------------|--|
| Précision pixelique ($h_t = 1$) | | | |
| Algorithme sur-relaxé | 43,75% / 43,16% | 44,15% / 45,18% | |
| Algorithme accéléré | 29,11% / 33,90% | 29,49% / 35,41% | |
| Précision sous-pixelique ($h_t = 0,5$) | | | |
| Algorithme sur-relaxé | 56,72% / 31,39% | 56,97% / 32,12% | |
| Algorithme accéléré | 44,42% / 23,22% | 44,58% / 23,71% | |
| Paire | | Venus | |
| Précision pixelique ($h_t = 1$) | | | |
| Algorithme sur-relaxé | 10,56% / 36,25% | 10,06% / 37,68% | |
| Algorithme accéléré | 6,68% / 23,48% | 6,96% / 24,97% | |
| Précision sous-pixelique ($h_t = 0,5$) | | | |
| Algorithme sur-relaxé | 16,17% / 21,98% | 15,77% / 22,31% | |
| Algorithme accéléré | 14,16% / 17,44% | 14,41% / 18,29% | |
| Paire | | Teddy | |
| Précision pixelique ($h_t = 1$) | | | |
| Algorithme sur-relaxé | 21,80% / 51,12% | 23,02% / 59,11% | |
| Algorithme accéléré | 16,43% / 37,77% | 17,95% / 46,08% | |
| Précision sous-pixelique ($h_t = 0,5$) | | | |
| Algorithme sur-relaxé | 33,57% / 36,40% | 36,14% / 45,23% | |
| Algorithme accéléré | 29,02% / 30,46% | 32,01% / 38,31% | |
| Paire | | Cones | |
| Précision pixelique ($h_t = 1$) | | | |
| Algorithme sur-relaxé | 25,81% / 55,95% | 26,12% / 58,49% | |
| Algorithme accéléré | 23,69% / 47,70% | 24,30% / 51,00% | |
| Précision sous-pixelique ($h_t = 0,5$) | | | |
| Algorithme sur-relaxé | 36,88% / 38,83% | 38,04% / 41,40% | |
| Algorithme accéléré | 37,54% / 38,67% | 38,64% / 41,65% | |

FIGURE 3.14 – Taux de précision et de rappel, avant (colonne de gauche) et après le traitement des occultations (colonne de droite).

| Paire | Tsukuba | | | | Venus | |
|---|---------|--------|--------|--------|--------|--------|
| Précision pixellique ($h_t = 1$) | | | | | | |
| Algorithme sur-relaxé | 86,52% | 65,33% | 61,66% | 90,47% | 68,55% | 63,29% |
| Algorithme accéléré | 91,83% | 83,59% | 79,96% | 93,71% | 87,48% | 80,67% |
| Précision sous-pixellique ($h_t = 0,5$) | | | | | | |
| Algorithme sur-relaxé | 83,41% | 71,72% | 70,46% | 93,71% | 82,94% | 81,97% |
| Algorithme accéléré | 87,05% | 79,54% | 78,28% | 95,53% | 91,89% | 89,56% |

| Paire | Teddy | | | | Cones | |
|---|--------|--------|--------|--------|--------|--------|
| Précision pixellique ($h_t = 1$) | | | | | | |
| Algorithme sur-relaxé | 96,08% | 79,92% | 55,79% | 95,08% | 74,83% | 70,24% |
| Algorithme accéléré | 96,76% | 94,59% | 78,79% | 96,08% | 90,03% | 84,12% |
| Précision sous-pixellique ($h_t = 0,5$) | | | | | | |
| Algorithme sur-relaxé | 96,44% | 89,01% | 76,87% | 95,83% | 85,68% | 82,77% |
| Algorithme accéléré | 96,88% | 90,81% | 79,40% | 96,29% | 87,54% | 84,05% |

FIGURE 3.15 – Erreur d’estimation dans les zones occultées, avant et après le traitement des occultations. Pour chaque expérience, on mesure l’erreur pixelique dans les zones occultées dans la première estimation de la disparité (colonne de gauche), après la détection des occultations par saturation de la pente (colonne du milieu) et après l’amélioration de ces détections (colonne de droite).

3.6 Discussion

3.6.1 Résultats

Estimation de la disparité En dehors des zones occultées, les cartes de disparité obtenues présentent les mêmes caractéristiques que celles obtenues grâce à une régularisation TV, c’est-à-dire des cartes constantes par morceaux, avec des contours simples et nets. Ainsi, certains détails de la scène ne peuvent être restitués, comme le bras de la lampe dans Tsukuba, dont les deux branches ont fusionné du fait de la régularité.

Avant le traitement des occultations, les bords gauches des objets présentent une rampe de disparité, qui est contrainte par la contrainte de visibilité. Cette contrainte produit également des artéfacts qui sont dus à la non-préservation de l’ordre dans la scène, qui est visible au niveau des pinceaux dans Cones ou le fil de la lampe de Tsukuba.

On notera également des effets de bords importants sur la partie gauche de la scène. Dans cette partie de l’image, l’occultation hors-champ conduit à des coûts de corrélation non significatifs (on les a choisis constants égaux à $g_{\max} = 100$). Néanmoins, on constate dans les figures 3.10 – 3.11 que ces erreurs sont diffusées en-dehors de cette région.

Détection des occultations Qualitativement, les occultations détectées présentent la même forme que celles de la vérité-terrain, ce qui tend à montrer que le modèle considéré est correct. En particulier, la carte dense produite induit des régions où la pente horizontale vaut 1, et celles-ci sont corrélées avec les zones occultées.

Néanmoins, les régions extraites par la saturation de la pente ne sont pas toujours correctement placées. C'est le cas en particulier de la paire Venus, où l'occultation du fond de la scène est légèrement décalée sur la gauche. On observe également des détections non significatives, car, les cartes étant quantifiées, un saut de disparité se produit même en cas de disparité variant de manière douce (comme les plans inclinés dans Venus ou le toit dans Cones) et ne correspondent pas à de réelles occultations. Or, dans le cas de la précision pixellique, celles-ci sont systématiquement détectées comme telles car toute discontinuité de la scène correspond à un saut d'au moins un pixel. Dans le cas de la précision sous-pixellique, il peut y avoir dans ces cas des sauts d'un demi-pixel, qui ne sont, eux, pas considérés comme des occultations.

Post-traitement des zones occultées On observe pour Teddy un autre phénomène déjà évoqué, qui est celui des alignements verticaux des lignes de niveaux. Ainsi, pour l'occultation induite par le toit de la maison, qui présente un bord oblique dans le repère de l'image, la régularité verticale tend à déplacer les discontinuités, et donc en particulier les lignes de saturation de la pente. La région affectée par ce phénomène est donc globalement plus large que la région effectivement occultée, tandis que la carte de saturation présente des trous. Dans ce cas, le post-traitement proposé permet d'améliorer de manière satisfaisante les détections. Plus précisément, le taux de précision gagne en moyenne +0,77% et le taux de rappel +3,33%. Il n'est pas étonnant de constater un gain plus important du rappel que de la précision, car le post-traitement vise à améliorer la densité de la détection dans les zones occultées, ce qui conduit en théorie (si la première détection est de bonne qualité) à rajouter des vrais positifs. On notera que, mis à part la paire Venus, les trois autres paires bénéficient, quelle que soit l'expérience considérée, de ce post-traitement.

Densification On peut vérifier les gains en terme d'estimation de la disparité de chacune de ces étapes de traitement dans le tableau 3.15 : pour la précision pixellique, on gagne en moyenne +12,74% dans les taux d'erreur pixellique après la première détection des occultations, puis +8,73% une fois ces détections améliorées, ce qui équivaut à un gain total de +21,47%. Pour la précision sous-pixellique, le gain est moins important (respectivement +9,46%, +4,47% et +13,94%). L'amélioration la plus importante concerne la paire Teddy (précision pixellique, carte obtenue par l'algorithme sur-relaxé), où l'on passe de 96,08% d'erreur pixellique à 55,79%, soit une erreur pratiquement divisée par deux.

3.6.2 Différences entre les deux algorithmes proposés

Équivalence des problèmes relaxé (3.13) et fortement convexe (3.17) Nous avons démontré l'équivalence entre le problème primal-dual continu (3.11)

$$\min_{v \in \mathcal{C}} \sup_{\phi \in \mathcal{K}} \left\{ \int_{\Omega \times \mathbb{R}} \phi Dv + \int_{\Omega \times \mathbb{R}} h^{\text{vis}}(Dv) \right\}$$

et le problème fortement convexe (3.14)

$$\min_{w \in \mathcal{D}} F(w) + \frac{1}{2} \|w\|_{L^2(\Omega \times I_{\text{disp}})}^2.$$

La discrétisation de ces deux problèmes n'en préserve visiblement pas l'équivalence. Dans la figure 3.8, on observe ainsi des différences dans le fond de la scène Venus.

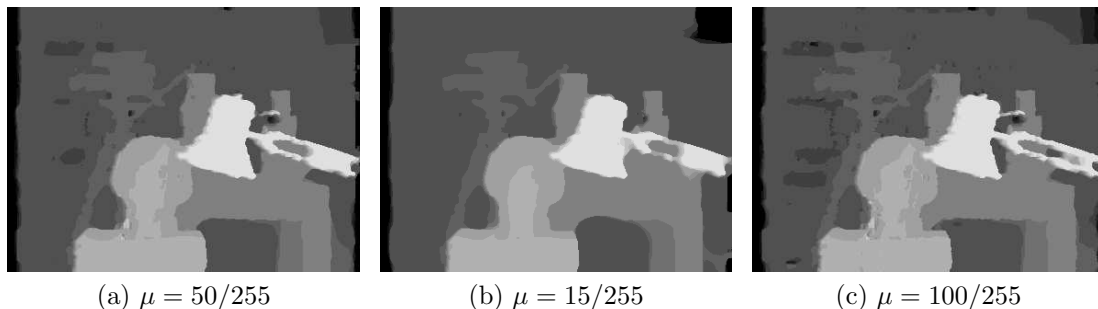


FIGURE 3.16 – Influence du paramètre μ sur la paire Tsukuba. Lorsque μ est trop faible (b), la disparité est trop régulière (voir par exemple sous la table). Lorsque μ est trop grand (c), la disparité paraît bruitée car elle n'est pas suffisamment régularisée. On retrouve en revanche certains détails perdus lorsque la régularisation est trop importante (comme le bras de la lampe).

On observe également des différences dans les scores considérés pour mesurer la performance des algorithmes testés. Ainsi, qu'il s'agit de la carte de la disparité ou de la carte des occultations, les résultats diffèrent selon la méthode employée. Plus précisément, sauf pour quelques cas particuliers (comme la paire Cones dans le cas de la précision sous-pixelique), l'algorithme accéléré conduit à des erreurs d'estimation de la disparité plus importantes (jusqu'à une différence de 1,81%). Les taux de précision et de rappel sont également systématiquement moins élevés, ce qui se traduit par une estimation moins bonne des occultations (à l'unique exception de Cones en précision sous-pixelique, où la différence pour le taux de précision est de 0,66% avant amélioration des détections). Il est en revanche indubitable que l'algorithme accéléré présente généralement une convergence plus rapide que l'algorithme sur-relaxé (à l'exception notable des paires Tsukuba et Venus pour la précision pixelique). Ainsi, le nombre d'itérations est réduit de moitié pour au moins deux expériences (Cones en précision pixelique et Venus en précision sous-pixelique). L'évolution du critère d'arrêt montre une allure globalement décroissante pour l'algorithme accéléré moins importante que celle de l'algorithme sur-relaxé, mais qui est localement oscillante, permettant d'atteindre plus rapidement le seuil de tolérance.

Influence de la sur-relaxation La sur-relaxation introduite dans l'algorithme primal-dual conduit expérimentalement à une convergence plus rapide de l'algorithme. Ainsi, le *primal-dual gap*, qui, rappelons-le, ne dépend pas de l'algorithme mais du problème considéré, montre une convergence vers zéro plus rapide lorsque $\rho = 1,95$ (voir courbe cyan dans la figure 6.6). Le nombre d'itérations est pratiquement réduit de moitié (sauf pour les paires Teddy et Cones en précision sous-pixelique, où nous avons arrêté les itérations au bout de 10 000 itérations). En outre, lorsque le critère d'arrêt est activé, les cartes obtenues sont dans les deux cas les mêmes.

3.6.3 Choix des paramètres

Influence du paramètre de pondération μ Le paramètre μ permet de choisir les poids respectifs du terme d'attache aux données et du terme de régularisation. Dans [12], il est suggéré de le choisir égal à 50/255, ce que nous faisons car les résultats obtenus nous paraissent convenables.

La valeur de ce paramètre est cruciale. Si μ est choisi trop faible, alors c'est la

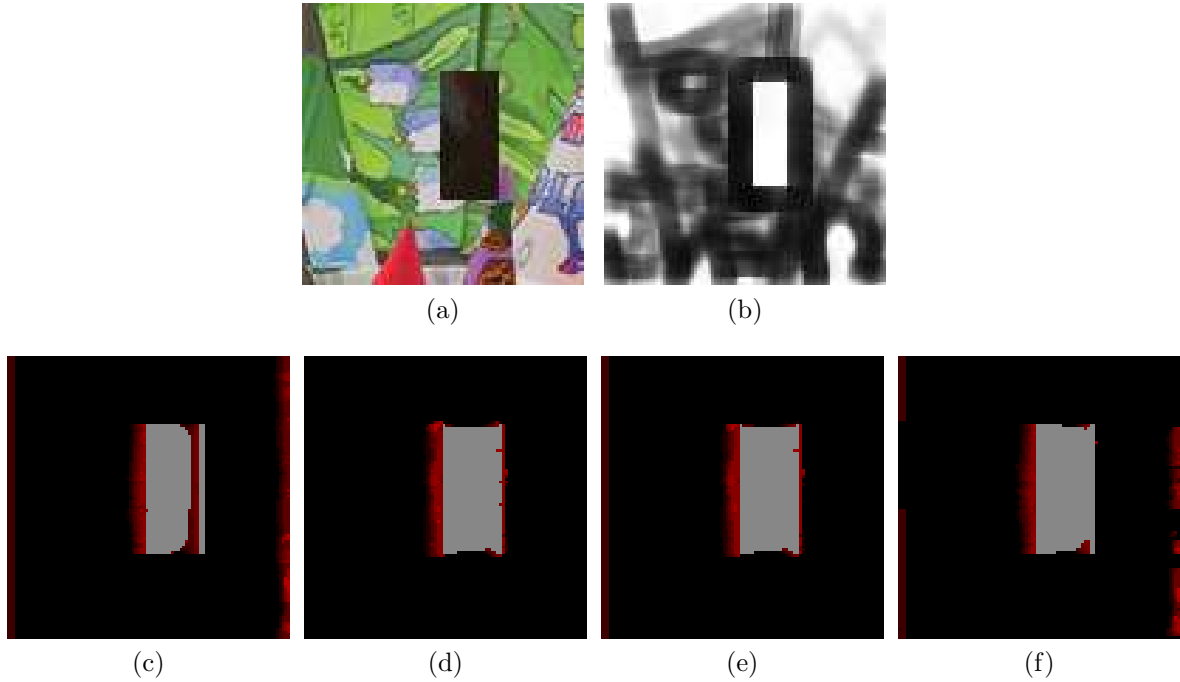


FIGURE 3.17 – Limitation de l’adhérence à l’aide du coefficient variable α . Le masque transparent rouge montre les erreurs. (a) Image de référence. (b) Valeurs prises par le coefficient variable α^h . Plus le pixel est sombre, moins la corrélation de gradient compte. (c) Corrélacion de couleurs. (d) Corrélacion de gradient. (e) Combinaison convexe des deux corrélacion, avec $\alpha = 0.9$ fixe [13, 14]. (f) Combinaison convexe des deux corrélacion, avec α^h variant entre 0 et 1. On voit que lorsque le gradient est utilisé, de l’adhérence (au moins sur le bord droit du rectangle) apparaît.

régularisation qui l’emporte et les cartes obtenues sont trop régulières (figure 3.16(b)). On voit par exemple les effets de cette sur-régularisation sous la table, ou juste au-dessus du bras de la lampe. Le fond de la scène paraît en revanche plus satisfaisant, car la disparité y présente moins de variations (dans la vérité-terrain, elle est constante). Au contraire, si μ est trop grand (figure 3.16(c)), alors la disparité n’est pas suffisamment régulière. Cela conduit à des cartes de disparité qui paraissent bruitées. Ce manque de régularité peut cependant améliorer l’estimation de la disparité dans des cas spécifiques, comme pour le bras de la lampe. Dès que la régularisation est un peu forte, alors l’espace entre deux barres disparaît.

On constate donc que le choix de ce paramètre est difficile et principalement empirique. La valeur optimale dépend de la scène considérée, mais elle ne sera pas toujours adaptée aux différentes parties de la scène. Il faut donc généralement trouver un équilibre entre une bonne régularisation et la perte de certains détails. Néanmoins, des expériences montrent que les résultats sont relativement peu sensibles à la valeur de μ , en ce sens qu’elle doit beaucoup varier avant que les résultats ne se dégradent de manière significative.

Influence de la pondération variable α Le terme d’attache aux données introduit dans le paragraphe 3.1.1 a été choisi pour tirer parti de l’efficacité de la corrélation combinée utilisée dans [13] et pour limiter le phénomène d’adhérence. Pour justifier cette dernière affirmation, on propose de comparer la performance de ces deux corrélacions sur un exemple synthétique, composé d’un rectangle texturé sur un fond texturé. On

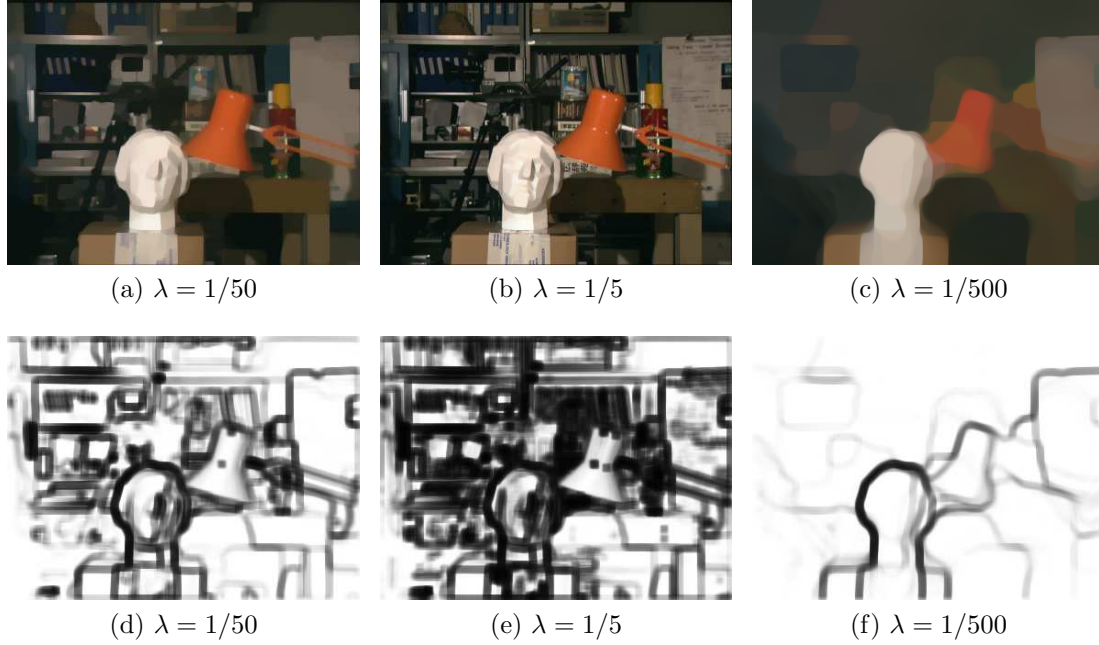


FIGURE 3.18 – Influence de la régularisation ROF de l'image sur la valeur de α^h . Ligne du haut : lissage de l'image pour différentes valeurs du paramètre λ . Ligne du bas : le coefficient α^h correspondant. On voit que quand (b) λ est trop grand, α est petit pratiquement partout (e), la corrélation sera donc principalement basée sur la couleur. Quand (c) λ est trop faible, alors α^h sera plutôt grand (f) et la corrélation reposera principalement sur le gradient, même près des discontinuités de la scène.

voit dans la figure 3.17 que, lorsque la corrélation de gradient est utilisée systématiquement sur toute la scène (3.17(d) et (e)), de l'adhérence apparaît, au moins sur le bord droit du rectangle. Ce n'est pas le cas ni avec la corrélation de couleurs (3.17(c)), ni avec le nouveau terme que nous avons introduit (3.17(f)). Près des bords du rectangle, le coefficient α^h est en effet choisi de sorte que la corrélation se fait principalement sur la couleur (figure 3.17(b)). En comparant la corrélation couleurs et de notre terme d'attache aux données sur cet exemple synthétique, on voit cependant que ce dernier conserve les performances de la corrélation de gradient sur la moitié droite du rectangle, où la disparité très mal estimée par la corrélation de couleurs.

La conception du coefficient variable α^h repose principalement sur deux composantes : la régularisation ROF de l'image de référence et le choix du paramètre a dans la définition de α^h . Nous allons observer l'influence de ces deux composantes. On rappelle que l'objectif est de concevoir un coefficient variable α^h qui soit proche de 0 près des discontinuités de la scène et proche de 1 ailleurs.

La régularisation ROF choisie dépend d'un paramètre, qui est le poids λ du terme d'attache aux données dans le modèle de débruitage ROF (3.21). La valeur de ce dernier détermine la force de la régularisation. Si la régularisation est trop importante (figure 3.18(b) et (e)), alors beaucoup de variations de l'image sont ignorées, dont celles de la scène, et c'est la comparaison de gradient qui est principalement utilisée, qui introduit de l'adhérence. Si elle est trop faible (figure 3.18(c) et (f)), trop de variations sont détectées et dans ce cas, c'est la comparaison d'intensité qui prime, dont on sait qu'elle est moins performante que la comparaison de gradient loin des discontinuités de scène. Le paramètre λ doit donc être choisi de manière à conserver les discontinuités significatives (c'est-à-dire celles qui coïncident avec les discontinuités de la scène) et

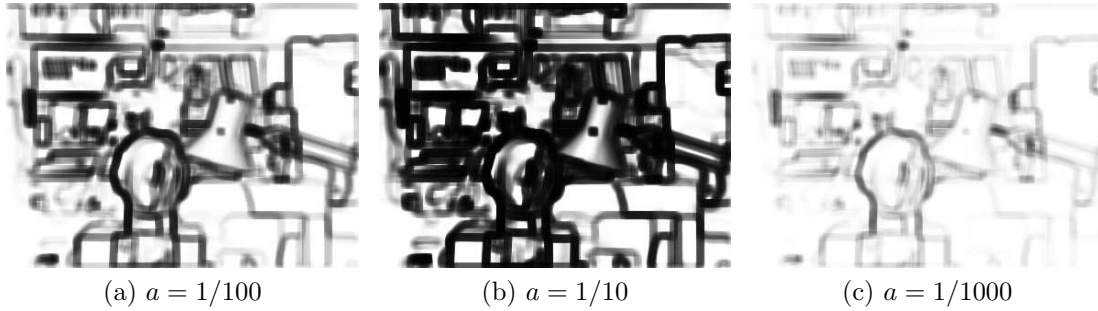


FIGURE 3.19 – Influence du choix du paramètre a sur la valeur de α^h . On voit que quand (b) a est trop grand, α est petit pratiquement partout, la corrélation sera donc principalement basée sur la couleur. Quand (c) a est trop faible, alors α^h sera plutôt grand et la corrélation reposera principalement sur le gradient, même près des discontinuités de la scène.

éliminer les autres variations (qui correspondent plutôt à de la texture). Ce choix est évidemment difficile car certaines textures présentent des variations d'intensité très importantes alors que certaines discontinuités de scène ne sont pas très visibles.

De la même manière, le choix du paramètre a est également important, car il permet de régler la sensibilité du détecteur de discontinuité : à partir de quelles valeurs de l'amplitude du gradient α devient-il négligeable ? On voit dans la figure 3.19 qu'à nouveau, a doit être choisi ni trop petit, ni trop grand, afin que les gradients significatifs correspondent à des valeurs proches de zéro pour α^h et que les autres pixels conduisent à des valeurs plus grandes.

Influence du seuil s Dans la version continue du problème que l'on étudie, le choix du seuillage s pour l'algorithme sur-relaxé est arbitraire. Cependant, le cas discret ne conserve pas cette propriété. En particulier, on observe expérimentalement qu'un seuillage près de zéro introduit de l'adhérence (les objets apparaissent plus gros qu'ils ne le sont réellement). Ainsi, il semble préférable de privilégier un seuillage près de 1.

Précision sous-pixellique Mis à part le cas de la paire Tsukuba, qui présente une vérité-terrain pixellique, l'introduction de la précision sous-pixellique améliore de manière significative l'estimation de la disparité dans les zones non occultées. Ce gain est particulièrement visible lorsque l'on mesure l'erreur sous-pixellique, puisque celle-ci est pratiquement diminuée de moitié.

En revanche, si elle améliore le taux de précision (pour les quatre paires, les meilleurs scores sont obtenus pour l'algorithme sur-relaxé en précision sous-pixellique), la précision sous-pixellique dégrade le taux de rappel. Ces résultats s'expliquent principalement par deux phénomènes. Le premier, qui est le plus visible, concerne les effets de bords que l'on peut observer sur la partie gauche des scènes. Sur le bord gauche des images, la disparité est en effet nulle, puis, au fur et à mesure que l'on parcourt la scène vers la droite, l'attache aux données lui impose de croître horizontalement. La saturation de la pente détecte donc cette région comme occultée. Or, pour la précision pixellique, cette région dépasse celle – non prise en compte – les occultations hors-champ et ce, de manière plus importante qu'avec la précision sous-pixellique. Le taux de précision pour cette dernière est de fait plus élevée lorsque l'on retire ces détections incorrectes. Le taux de rappel plus faible indique quant à lui une sous-détection dans les zones occultées. Celle-ci s'explique par le fait qu'introduire une disparité sous-

pixelique permet de remplacer un saut (horizontal) d'un pixel par deux sauts d'un demi-pixel chacun. Or, le premier est détecté comme occultation, ce qui n'est pas le cas des seconds. C'est le terme d'attache aux données qui détermine le choix de l'un de ces deux scénarios. Ainsi, si l'utilisation d'un saut sous-pixelique permet d'éviter que des discontinuités de scènes dues à la quantification pixelique de la disparité ne soit interprétées comme des occultations, elle permet au contraire à des occultations de passer inaperçues en ne saturant pas la pente horizontale.

3.6.4 Comparaison avec d'autres algorithmes

Comparaison avec KZ2 Comparons la détection d'occultation avec celle de [10], qui, comme on le verra au chapitre suivant, propose une gestion différente de l'occultation. La figure 3.20 présente les occultations détectées par [10] (implémentation de [9]). Commençons par souligner que cette méthode est capable de détecter les occultations hors-champ. Cette bonne performance ne se lira pas sur les différentes statistiques présentées.

On voit que les occultations détectées par [10] sont globalement mieux localisées que celles que nous obtenons, en particulier sur la paire Venus. Les occultations dans la paire Teddy (notamment celles induites par la maison) sont également remarquablement détectées. Néanmoins, de nombreux faux négatifs peuvent être observés pour Cones. Ainsi, hormis pour la paire Cones, la méthode de KOLMOGOROV et de ZABIH affiche des scores plus élevés.

Ainsi qu'on le verra au chapitre 4, la détection des occultations dans [10] repose sur le choix d'un certain paramètre, noté K . Celui-ci, s'il est choisi trop petit, conduit à détecter trop de pixels occultés, et au contraire, la méthode n'en détecte pas suffisamment s'il est trop grand. Il s'agit en effet du coût de « corrélation » attribué aux pixels occultés.

On voit donc que le choix de ce paramètre est crucial, alors que notre méthode est principalement basée sur la saturation de la pente horizontale, qui ne dépend elle d'aucun paramètre. Elle dépend en revanche du seuil T qui permet, dans l'étape d'amélioration des détections d'occultation, de décider si deux pixels appartiennent au même objet. Ce seuil est néanmoins plus facile à choisir, puisqu'il s'agit de déterminer en moyenne à partir de quel seuil deux couleurs sont significativement différentes. En particulier, il ne dépend pas de la scène, mais de la perception (ou plutôt, de la différenciation) des couleurs. Par ailleurs, cette étape ne corrige qu'une petite partie des occultations, ce qui signifie que notre méthode est moins sensible au choix de ce paramètre (tant qu'il reste raisonnable).

Il faut également noter que, dans la méthode de KOLMOGOROV et de ZABIH, les pixels occultés ne paient qu'un coût de « corrélation » K fixe, alors, dans notre méthode, ceux-ci paient le coût de corrélation associé à la mise en correspondance avec un pixel qui ne peut être leur pixel homologue. Ce coût est donc potentiellement important et peut entraîner des solutions non satisfaisantes mais légèrement moins coûteuses (revoir par exemple l'analyse associée à la figure 3.1). Ainsi, on observe pour la paire Venus de nombreuses détections ponctuelles, qui n'ont aucune signification physique.

Comparaison avec la régularisation TV seule [12] On peut également comparer notre méthode à celle de [12], dans le cas de la régularisation TV. On choisit d'utiliser le même terme d'attache aux données. La fonctionnelle d'énergie est donc la même que celle considérée dans ce chapitre, à l'exception du terme de visibilité qui n'est pas pris

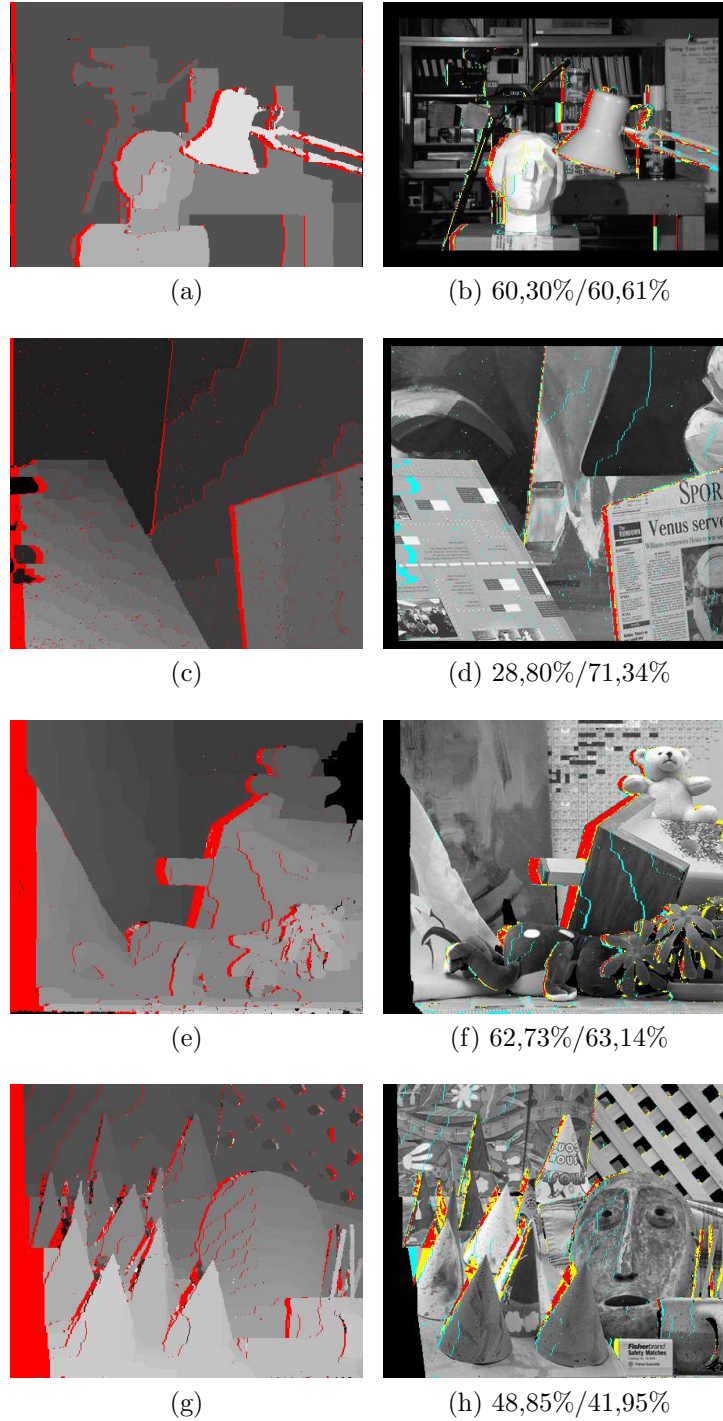


FIGURE 3.20 – Détections des occultations par la méthode [10]. Colonnes de gauche : carte des occultations générées par [9] (en rouge). Colonnes de droite : en rouge, les détections correctes (vrais positifs, TP), en jaune, les détections manquantes (faux négatifs, FN) et en cyan, les détections incorrectes (faux positifs, FP). En légende : les taux de précision et de rappel. De haut en bas : Tsukuba, Venus, Teddy et Cones.

en compte. En l'absence de ce terme, les occultations peuvent être détectées par un filtre LRRL qui vérifie la consistance des cartes de disparité calculées sur la vue de gauche et sur la vue de droite. Plus précisément, deux pixels mis en correspondance dans une des deux cartes devraient le rester dans l'autre carte. Si ce n'est pas le cas, ces pixels sont rejetés (et considérés généralement comme occultés).

La figure 3.21 propose les cartes de telles détectations, ainsi que, en légende, les taux de précision et de rappel associés. Dans l'ensemble, ces taux sont meilleurs que ceux obtenus dans notre méthode. Visuellement, on observe des détectations mal situées, comme celles localisées sur le buste de Tsukuba. Il faut noter que l'obtention de telles cartes requiert le calcul de deux cartes de disparité, ce qui revient à doubler le temps de calculs. Néanmoins, il est possible de les calculer en parallèle.

Conclusion

Nous avons proposé dans ce chapitre une méthode variationnelle pour la stéréovision binoculaire dont la fonctionnelle d'énergie, qui traduit le modèle de scène considéré, est relativement peu biaisée par la méthode d'optimisation. La régularisation choisie est en effet la régularisation TV, qui est classiquement choisie en traitement d'images car elle lisse tout préservant les discontinuités de la scène. Elle est donc généralement adaptée aux images rencontrées. L'attache aux données est quant à elle une version adaptative d'une corrélation qui a démontré son efficacité [13, 14], modifiée de sorte de limiter l'adhérence qu'elle introduit aux bords des objets. L'apport principal réside toutefois dans le terme de visibilité, que nous avons intégré au terme de régularité. Il permet d'une part de correctement gérer le phénomène d'occultation et d'autre part d'utiliser la méthode de relaxation convexe proposée par [12]. À l'issue d'une première étape d'optimisation convexe, on obtient une carte de disparité dans laquelle on peut détecter les occultations. Un post-traitement permet d'abord d'améliorer la détection de ces zones, puis de densifier la carte de disparité résultante. Par ailleurs, bien que la détection des occultations est légèrement moins performante que celle proposée par KOLMOGOROV et ZABIH dans [10], elle est moins sensible au choix de son paramètre (qui ne dépend pas de la scène, mais du système de couleur choisi), alors que la méthode [10] dépend du choix d'un paramètre, qu'il faut savoir correctement estimer. Enfin, malgré l'introduction d'un algorithme fortement convexe, l'efficacité reste médiocre. La relaxation convexe ajoute en effet une dimension au problème, et la contrainte de pente empêche de mesurer de manière précise la convergence de l'algorithme, ce qui nécessite de recourir à des critères d'arrêt plus heuristiques. Cette complexité est un frein pour la précision sous-pixelique, alors que cette méthode présente l'avantage d'être, en théorie, capable de produire des cartes à cette précision.

Le modèle de scène considéré dans notre méthode est donc plus satisfaisant que la plupart des méthodes globales, car la régularisation est naturelle et les occultations sont traitées comme découlant de manière structurelle de la disparité. Néanmoins, les détectations obtenues sont moins bonnes que celles que fournit [10]. Un travail futur consisterait à développer un moyen de faire coïncider les zones occultées et les discontinuités de la scène, en considérant par exemple une pondération spatiale de la régularisation (TV non locale). Les performances de [10] nous conduisent par ailleurs à l'utiliser dans le chapitre suivant pour améliorer la détection des occultations dans des méthodes qui n'en proposent pas mais qui fournissent des estimations (denses ou non) fiables par ailleurs, en adaptant l'algorithme pour la densification de cartes non denses et le raffinement sous-pixelique de cartes pixeliques, avec dans chaque cas une

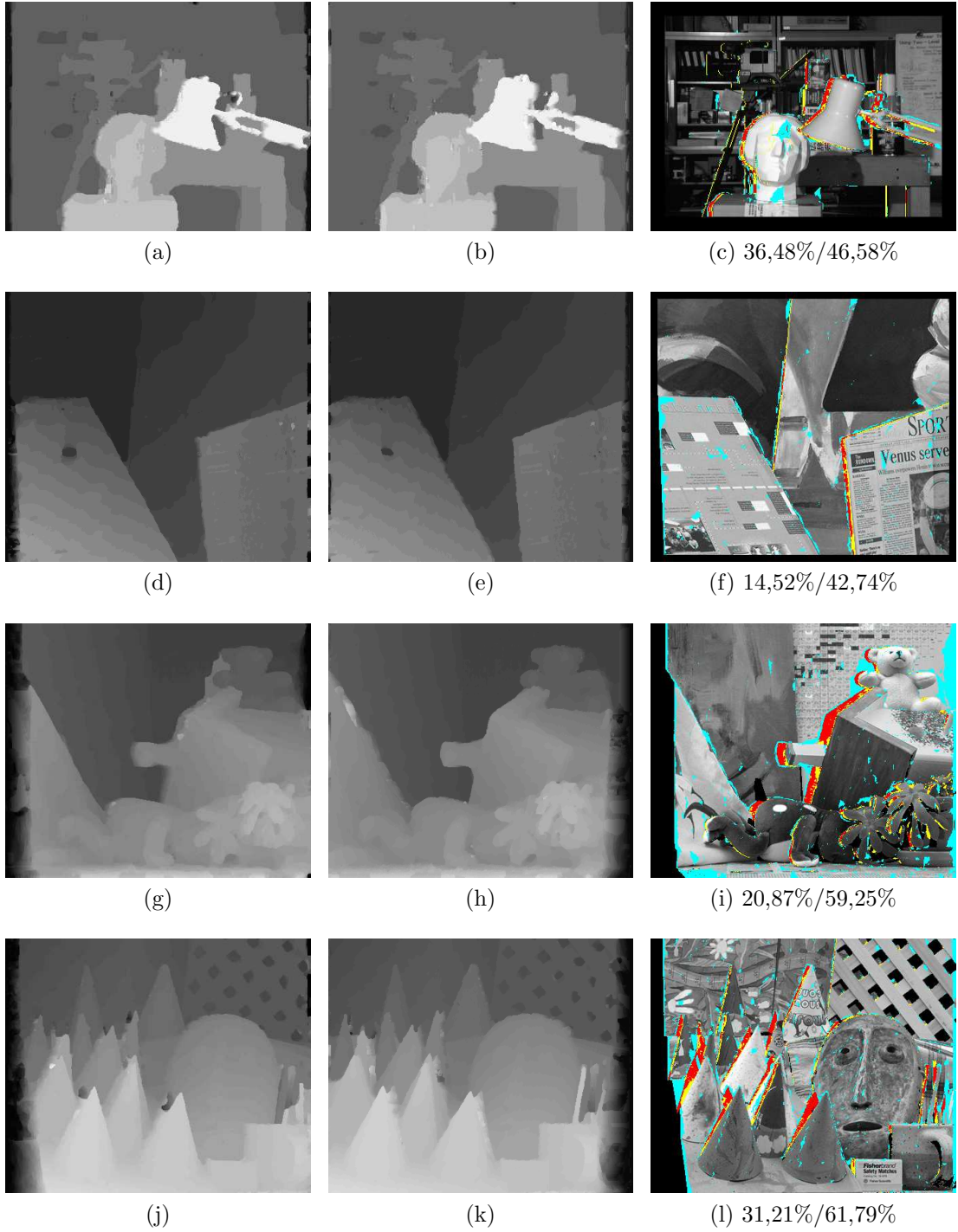


FIGURE 3.21 – Detections des occultations par le filtre LRRL dans le cas d’une carte obtenue par une régularisation TV du volume de coût g . Colonne de gauche : carte de disparité de la vue de gauche. Colonne du milieu : carte de disparité de la vue de droite. Colonne de droite : en rouge, les detections correctes (vrais positifs, TP), en jaune, les detections manquantes (faux négatifs, FN) et en cyan, les detections incorrectes (faux positifs, FP). En légende : les taux de précision et de rappel. De haut en bas : Tsukuba, Venus, Teddy et Cones.

Références

- [1] Aaron F. BOBICK and Stephen S. INTILLE. Large occlusion stereo. *International Journal of Computer Vision*, 33(3) :181–200, 1999.
- [2] Guy BOUCHITTÉ and Michel VALADIER. Integral representation of convex functionals on a space of measures. *Journal of Functional Analysis*, 80(2) :398–420, 1988.
- [3] Antonin CHAMBOLLE, Vicent CASELLES, Daniel CREMERS, Matteo NOVAGA, and Thomas POCK. An introduction to total variation for image analysis. *Theoretical Foundations and Numerical Methods for Sparse Recovery*, 9(263-340) :227, 2010.
- [4] Antonin CHAMBOLLE and Thomas POCK. On the ergodic convergence rates of a first-order primal–dual algorithm. *Mathematical Programming*, pages 1–35, 2015.
- [5] Laurent CONDAT. A primal–dual splitting method for convex optimization involving lipschitzian, proximable and linear composite terms. *Journal of Optimization Theory and Applications*, 158(2) :460–479, 2013.
- [6] Laurent CONDAT and Nelly PUSTELNIK. Segmentation d'image par optimisation proximale. In *XXVème colloque GRETSI (GRETSI 2015)*, 2015.
- [7] Lawrence Craig EVANS and Ronald F. GARIEPY. *Measure theory and fine properties of functions*. CRC press, 2015.
- [8] 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.
- [9] Vladimir KOLMOGOROV, Pascal MONASSE, and Pauline TAN. KOLMOGOROV and ZABIH's graph cuts stereo matching algorithm. *Image Processing On Line*, 4 :220–251, 2014.
- [10] 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.
- [11] Thomas POCK, Daniel CREMERS, Horst BISCHOF, and Antonin CHAMBOLLE. An algorithm for minimizing the MUMFORD-SHAH functional. In *IEEE International Conference on Computer Vision*, pages 1133–1140. IEEE, 2009.
- [12] 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.
- [13] 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.

-
- [14] Pauline TAN and Pascal MONASSE. Stereo disparity through cost aggregation with guided filter. *Image Processing On Line*, pages 252–275, 2014.
 - [15] Philippe THÉVENAZ, Thierry BLU, and Michael UNSER. Image interpolation and resampling. *Handbook of medical imaging, processing and analysis*, pages 393–420, 2000.
 - [16] Michael UNSER. Sampling – 50 years after SHANNON. *Proceedings of the IEEE*, 88(4) :569–587, 2000.
 - [17] Cédric VILLANI. Intégration et analyse de fourier. *Notes de cours pour l'ENS Lyon*, 6, 2005.
