

Spécification de l'intervalle de disparité par pixel dans la méthode des graph cuts

Introduction	116
4.1 Fonctionnelle d'énergie	116
4.1.1 Terme d'attache aux données et d'occultation	117
4.1.2 Terme de régularisation	118
4.1.3 Terme d'injectivité	119
4.2 Représentation d'une énergie par un graphe	119
4.2.1 Coupure minimale et flot maximal	119
4.2.2 Représentabilité d'une énergie	121
4.3 Décroissance de l'énergie par <i>expansion move</i> optimal	123
4.3.1 <i>Expansion moves</i> et décroissance de l'énergie	123
4.3.2 Assignements et configurations	124
4.3.3 Énergie d'une configuration	126
4.3.4 Énergie d'un <i>expansion move</i>	128
4.3.5 Représentabilité de l'énergie des <i>expansion moves</i>	130
4.4 Résolution numérique par coupure de graphes	132
4.4.1 Construction du graphe	132
4.4.2 Recherche du flot maximal par chemins augmentants	134
4.4.3 α - <i>Expansion move</i> optimal	134
4.4.4 Paramètres	135
4.5 Adapter l'intervalle de disparité au pixel	136
4.5.1 Intervalle adaptatif	136
4.5.2 Densification de cartes de disparité	138
4.5.3 Raffinement subpixellique	140
4.6 Résultats expérimentaux	141
4.6.1 Algorithme original	141
4.6.2 Densification de cartes éparses	148
4.6.3 Précision subpixellique	148
4.6.4 Raffinement subpixellique	150
4.6.5 Discussion	150
Conclusion	161

Introduction

En 2001, Vladimir KOLMOGOROV et Ramin ZABIH proposent une méthode globale basée sur l'utilisation de graphes pour minimiser de manière approchée une fonctionnelle d'énergie. L'intérêt de cette méthode réside en deux principales caractéristiques. En premier lieu, elle ne découle pas d'une formulation variationnelle. Autrement dit, la disparité discrète n'est pas la discrétisation d'une fonction continue, elle peut donc en particulier prendre des valeurs non réelles (sans être infinie). Cette propriété permet de définir une étiquette qui marque les pixels comme *occultés*, sans devoir les mettre artificiellement en correspondance avec d'autres pixels (section 4.1). En second lieu, la stratégie de minimisation, qui repose sur l'algorithme des *expansion moves*, peut être gérée de manière efficace en la reformulant comme des recherches de coupures minimales dans un graphe (section 4.2). Ce dernier problème est bien connu depuis FORD et FULKERSON et de nombreux algorithmes existent qui le résolvent efficacement.

Cependant, on verra que les performances de la méthode reposent grandement sur la valeur de certains paramètres, dont le choix peut s'avérer délicat (section 4.6). En outre, l'utilisation des *expansion moves* ne permet pas de minimiser exactement la fonctionnelle, mais seulement de proposer un schéma de décroissance de l'énergie (section 4.3). Si la fonctionnelle ne possède pas de minima locaux, on peut raisonnablement espérer en trouver le minimum global de cette manière. Malheureusement, cette propriété n'est pas assurée car la fonctionnelle n'est pas convexe. Par ailleurs, la représentabilité de l'énergie (section 4.2) introduit des contraintes quant au choix de la fonctionnelle d'énergie. Néanmoins, l'algorithme donne des résultats très satisfaisants, notamment en terme de détection des occultations (voir le chapitre précédent), et dans un temps raisonnable.

L'objet principal de ce chapitre est de proposer une modification de la méthode de KOLMOGOROV et ZABIH qui permette de définir pour chaque pixel un intervalle de disparité adapté (section 4.5). Une telle possibilité permet d'utiliser l'efficacité des coupures de graphes pour densifier des cartes éparses ou de raffiner des cartes à des précisions subpixelles. On se propose donc dans un premier temps de décrire la méthode originale, dont les différents éléments ont été détaillés dans plusieurs articles [8, 9, 3], ainsi que dans la thèse de KOLMOGOROV [6]. Cette étude reprend en partie la publication IPOL [7] consacrée à cette méthode. L'objectif est d'offrir la compréhension nécessaire pour envisager les modifications souhaitées, dont les différents résultats expérimentaux sont proposés à la section 4.6.

4.1 Fonctionnelle d'énergie

Dans tout ce qui suit, on reprend l'approche de la méthode KZ2 de [8], déjà détaillée dans l'article IPOL [7]. Toutefois, nous reformulons de manière différente la fonctionnelle, pour faciliter la comparaison avec celle de la méthode proposée au chapitre précédent (dont la version discrète est donnée dans la section 3.3.1).

Plaçons-nous directement dans la formulation discrète. Les pas de discrétisation des images et de l'intervalle de disparité valent 1, ce qui conduit, en conservant les notations du chapitre précédent (paragraphe 3.3.1) à considérer le vecteur $h = (1, 1, 1)$. Contrairement à la méthode étudiée dans le chapitre précédent, la disparité u^h est à valeurs dans $I_{\text{disp}}^h \cup \text{occ}$, où *occ* est une étiquette particulière destinée à déclarer un

pixel occulté. Le problème de mise en correspondance s'écrit donc

$$\min_{u^h \in (I_{\text{disp}}^h \times \{\text{occ}\})^{N_x N_y}} E(u^h)$$

où les images I_L et I_R sont de tailles $N_x \times N_y$. On rappelle que $I_{\text{disp}}^h = \llbracket d_{\min} ; d_{\max} \rrbracket \subset \mathbb{Z}$.

La fonctionnelle considérée possède trois termes :

$$E(u^h) = E_{\text{data+occ}}(u^h) + E_{\text{reg}}(u^h) + E_{\text{inj}}(u^h).$$

Le premier terme correspond au terme d'attache aux données classique, auquel on ajoute un terme pour prendre en compte les pixels occultés; le second terme mesure la régularité de la disparité; enfin, le dernier terme assure l'injectivité de la mise en correspondance.

4.1.1 Terme d'attache aux données et d'occultation

Ce terme attribue un coût à tout pixel, qui est soit le coût de corrélation si le pixel n'est pas occulté, soit un *coût d'occultation* sinon. Il se décompose donc en deux termes distincts :

$$E_{\text{data+occ}}(u^h) = E_{\text{data}}(u^h) + E_{\text{occ}}(u^h).$$

Terme d'attache aux données Ce terme somme les coûts de corrélation pour chaque pixel non occulté :

$$E_{\text{data}}(u^h) = \sum_{i=0}^{N_x-1} \sum_{j=0}^{N_y-1} g((i,j), u_{i,j}^h) (1 - \mathbb{1}_{\{\text{occ}\}}(u_{i,j}^h)). \quad (4.1)$$

La fonction de corrélation g est choisie de manière suivante pour la corrélation AD :

$$g_{\text{AD}}(x,t) = N_{I_L, I_R}(x, x-t),$$

tandis que pour la corrélation SD, on a

$$g_{\text{SD}}(x,t) = N_{I_L, I_R}(x, x-t)^2.$$

Ces deux corrélations sont définies à partir de la norme N , qui vaut dans le cas pixellique

$$N_{I_L, I_R}^{\text{pix}}(x, x-t) = \frac{1}{3} \sum_{c \in \{r, g, b\}} \min \left\{ |I_L^c(x) - I_R^c(x-t)|, 30 \right\}$$

ce qui revient à calculer dans chaque canal couleur une différence en valeur absolue seuillée, puis de fusionner les trois coûts obtenus à l'aide d'une moyenne. L'utilisation d'un seuil est fréquent (c'est le cas par exemple dans [10]) : lorsque la dissimilarité est importante pour tout couple de pixels, elle évite qu'une mise en correspondance soit privilégiée par rapport à une autre alors qu'aucune n'est pertinente. Dans le cas subpixellique, c'est une variante de la mesure de BIRCHFIELD et TOMASI [1] qui est utilisée :

$$N_{I_L, I_R}^{\text{subpix}}(x, x-t) = \frac{1}{3} \sum_{c \in \{r, g, b\}} \min \left\{ (\|I_L^c(x) - I_R^c(x-t)\|_{\text{BT}}, 30) \right\}.$$

La dissimilarité dans chaque canal est mesurée par la fonction

$$\|I_L^c(x) - I_R^c(y)\|_{\text{BT}} = \max \left\{ 0, I_L^c(x) - (I_R^c)^{\max}(y), (I_R^c)^{\min}(y) - I_L^c(x) \right\}$$

qui calcule la distance de l'intensité $I_L^c(x)$ au segment $\left[(I_R^c)^{\min}(y); (I_R^c)^{\max}(y) \right]$, où les images $(I_R^c)^{\max}$ et $(I_R^c)^{\min}$ sont définies respectivement par

$$(I_R^c)^{\max}(y) = \max_{r \in \{(0,0), (\pm 1,0), (0,\pm 1)\}} \left\{ \tilde{I}_R^c(y + r/2) \right\}$$

et

$$(I_R^c)^{\min}(y) = \min_{r \in \{(0,0), (\pm 1,0), (0,\pm 1)\}} \left\{ \tilde{I}_R^c(y + r/2) \right\}$$

avec

$$\tilde{I}_R^c(y + r/2) = \frac{1}{2} \left(I_R^c(y) + I_R^c(y + r) \right)$$

l'interpolation bilinéaire du canal couleur c de l'image I_R . Autrement dit, $(I_R^c)^{\max}(y)$ (respectivement $(I_R^c)^{\min}(y)$) correspond à la valeur maximale (resp. minimale) prise par l'interpolée \tilde{I}_R^c sur le carré de côté 1 centré en y . Dans [1], les auteurs montrent que l'utilisation de cette mesure de corrélation réduit le biais dû à l'échantillonnage (eux se contentant de considérer $r \in \{(0,0), (0, \pm 1)\}$). En effet, si le pixel homologue se trouve dans ce carré mais pas sur la grille d'échantillonnage, alors le coût de corrélation restera malgré tout faible.

REMARQUE : Le coût de corrélation présenté ici est celui proposé par les auteurs de l'article original [8]. Il a été repris et implanté dans l'article IPOL consacré à cette méthode [7]. C'est la raison pour laquelle il est utilisé dans ce chapitre. Toutefois, rien dans la méthode n'empêche l'utilisation d'une autre mesure de dissimilarité.

Terme d'occultation Le terme d'occultation associe à chaque pixel occulté un coût d'occultation fixe $K \geq 0$

$$E_{\text{occ}}(u^h) = \sum_{i=0}^{N_x-1} \sum_{j=0}^{N_y-1} K \mathbb{1}_{\{\text{occ}\}}(u_{i,j}^h). \quad (4.2)$$

Le choix de la pénalité K sera explicité ultérieurement.

4.1.2 Terme de régularisation

Le terme de régularisation pénalise toute variation de la disparité. Cette pénalité ne dépend pas de l'amplitude de cette variation, mais de la variation d'intensité au niveau de la variation de disparité :

$$\begin{aligned} E_{\text{reg}}(u^h) = & \sum_{i=1}^{N_x-1} \sum_{j=0}^{N_y-1} R_{\lambda_1, \lambda_2}((i,j), (i,j-1), u_{i,j}^h) + R_{\lambda_1, \lambda_2}((i,j), (i,j-1), u_{i,j-1}^h) \\ & + \sum_{i=0}^{N_x-1} \sum_{j=1}^{N_y-1} R_{\lambda_1, \lambda_2}(u_{i,j}^h, u_{i,j-1}^h) \mathbb{1}_{\{0\}}(\mathcal{N}(u_{i,j}^h, u_{i,j-1}^h)) \end{aligned} \quad (4.3)$$

où la pénalité R_{λ_1, λ_2} est définie par

$$R_{\lambda_1, \lambda_2}(p, p', u) = \begin{cases} 0 & \text{si } u = \text{occ} \\ \lambda_1 & \text{si } u \neq \text{occ} \text{ et } N_{I_L, I_L}^{\text{pix}}(p, p') \leq 8 \\ & \text{et } N_{I_R, I_R}^{\text{pix}}(p + u, p' + u) \leq 8 \\ \lambda_2 & \text{sinon} \end{cases}$$

avec $0 \leq \lambda_2 \leq \lambda_1$. Autrement dit, la variation de disparité est moins pénalisée si elle s'accompagne d'une variation d'intensité importante dans la vue de gauche ou dans la vue de droite.

4.1.3 Terme d'injectivité

Enfin, le terme d'injectivité impose que deux pixels de l'image de référence possèdent deux pixels homologues différents. Cette contrainte se traduit sous la forme suivante :

$$E_{\text{inj}}(u^h) = \sum_{i=0}^{N_x-1} \sum_{j=0}^{N_y-1} \sum_{j'=0}^{N_y-1} \chi_{\mathbb{R} \setminus \{j-u_{i,j}^h\}}(j' - u_{i,j'}^h). \quad (4.4)$$

REMARQUE : La contrainte d'injectivité tient partiellement compte de l'analyse de l'occultation proposée dans le chapitre 2 en ce sens que si la contrainte de visibilité (sur la pente horizontale) est respectée, alors tous les pixels saturant la pente doivent être déclarés occultés à cause de ce terme. En effet, ces pixels sont tous mis en correspondance avec le même pixel, ce qui est ici interdit.

4.2 Représentation d'une énergie par un graphe

On va montrer dans cette section qu'il est possible d'utiliser les graphes pour résoudre de manière efficace certains problèmes d'optimisation d'énergies. Pour cela, on introduit le concept de représentation d'une énergie par un graphe.

4.2.1 Coupure minimale et flot maximal

Graphe orienté et pondéré Commençons par introduire quelques définitions utiles pour la suite. On appelle *graphe* \mathcal{G} le couple $(\mathcal{V}, \mathcal{E})$ où \mathcal{V} est un ensemble dont les éléments sont appelés *sommets* (*vertices* en anglais) et $\mathcal{E} \subset \mathcal{V}^2$ est l'ensemble des arcs *orientés* et *pondérés* du graphe (*edges* en anglais). L'orientation des arcs implique de distinguer les deux arcs (a_1, a_2) et (a_2, a_1) si $a_1 \neq a_2 \in \mathcal{V}$ et la pondération consiste à leur attribuer une valeur $c_{\mathcal{G}}(a_1, a_2) \in]0; +\infty]$, appelé *capacité* de l'arc. On ignore les arcs de capacité nulle, ce qui revient à les supprimer de l'ensemble \mathcal{E} . Notons qu'on accepte les arcs de capacité infinie.

Pour plus d'informations sur la théorie des graphes, on pourra se reporter à [2].

Coupure d'un graphe Désormais, on suppose que le graphe \mathcal{G} possède deux sommets distincts s et t , appelés respectivement *source* et *puits* du graphe. Une *coupure*¹ du graphe désigne alors toute partition $(\mathcal{V}^s, \mathcal{V}^t)$ des sommets telle que la source s (resp. le puits t) appartienne au sous-ensemble \mathcal{V}^s (resp. \mathcal{V}^t). Le *coût* d'une coupure $(\mathcal{V}^s, \mathcal{V}^t)$ est donnée par la somme des capacités des arcs $(a_1, a_2) \in \mathcal{E}$ partant de $a_1 \in \mathcal{V}^s$ et aboutissant à $a_2 \in \mathcal{V}^t$. Elle est notée $C(\mathcal{V}^s, \mathcal{V}^t)$ et donnée par la formule

$$C_{\mathcal{G}}(\mathcal{V}^s, \mathcal{V}^t) = \sum_{\substack{(a_1, a_2) \in \mathcal{E} \\ a_1 \in \mathcal{V}^s, a_2 \in \mathcal{V}^t}} c_{\mathcal{G}}(a_1, a_2).$$

On appelle *coupure minimale* d'un graphe la coupure de coût minimal.

Flot d'un graphe On appelle *flot*² du graphe \mathcal{G} toute fonction $\Phi : \mathcal{E} \rightarrow [0; +\infty]$ vérifiant les deux conditions suivantes, la première sur les arcs, appelée *contrainte de capacité* :

$$\forall (a_1, a_2) \in \mathcal{E}, \quad 0 \leq \Phi(a_1, a_2) \leq c_{\mathcal{G}}(a_1, a_2)$$

1. *Cut* en anglais, si bien qu'on rencontre parfois le terme *coupe* également.
2. *Flow* en anglais.

et la seconde sur les sommets

$$\forall a \in \mathcal{V} \setminus \{s, t\}, \quad \sum_{(a, a') \in \mathcal{E}} \Phi(a, a') = \sum_{(a', a) \in \mathcal{E}} \Phi(a', a).$$

Cette dernière propriété est appelée *loi de conservation du flot* (aussi connue sous le nom de loi de KIRCHHOFF), car elle assure que le flot aboutissant en a est égal au flot qui en est issu. La *valeur* d'un flot est alors défini par la valeur commune

$$V(\Phi) = \sum_{(s, a) \in \mathcal{E}} \Phi(s, a) = \sum_{(a, t) \in \mathcal{E}} \Phi(a, t).$$

On appelle *flot maximal* d'un graphe le flot de valeur maximale.

Max-Flow/Min-Cut Le problème de recherche de la coupure minimale est en réalité dual à celui de recherche du flot maximal. Le théorème de FORD-FULKERSON [4] assure que la valeur du flot maximal d'un graphe est égale au coût de sa coupure minimale. Par ailleurs, le flot maximal peut être obtenu grâce à l'algorithme de FORD-FULKERSON. Celui-ci est basé sur le principe des *chemins augmentants*. On commence par rechercher un chemin orienté dans le graphe \mathcal{G} liant la source au puits, c'est-à-dire $m \in \mathbb{N}$ sommets $(a_{i_m}) \in \mathcal{V}^m$ tels que

$$(s, a_{i_1}) \in \mathcal{E}, \quad \forall k \in \llbracket 1; m-1 \rrbracket, \quad (a_{i_k}, a_{i_{k+1}}) \in \mathcal{E} \quad \text{et} \quad (a_{i_m}, t) \in \mathcal{E}$$

et que ces arcs soient de capacité strictement positive. Dorénavant, on dira qu'un chemin reliant un sommet à un autre est de capacité strictement positive si chacun des arcs qui le composent sont de capacité strictement positive. En choisissant le minimum de ces capacités c_{\min} , on peut alors définir un flot en posant

$$\forall (a_1, a_2) \in \mathcal{E}, \quad \begin{cases} \Phi(a_1, a_2) = c_{\min} & \text{si } (a_1, a_2) = (s, a_{i_1}) \\ \Phi(a_1, a_2) = c_{\min} & \text{si } (a_1, a_2) = (a_{i_k}, a_{i_{k+1}}), k \in \llbracket 1; m-1 \rrbracket \\ \Phi(a_1, a_2) = c_{\min} & \text{si } (a_1, a_2) = (a_{i_m}, t) \\ \Phi(a_1, a_2) = 0 & \text{sinon.} \end{cases}$$

On construit ensuite un nouveau graphe, appelé *graphe résiduel* de \mathcal{G} et noté \mathcal{G}_{Φ} . Ce graphe possède les mêmes sommets et arcs que le graphe initial \mathcal{G} , mais ses arcs sont pondérés de la manière suivante :

$$\forall (a_1, a_2) \in \mathcal{V}^2, \quad c_{\mathcal{G}_{\Phi}}(a_1, a_2) = \begin{cases} c_{\mathcal{G}}(a_1, a_2) - \Phi(a_1, a_2) & \text{si } (a_1, a_2) \in \mathcal{E} \\ \Phi(a_1, a_2) & \text{si } (a_2, a_1) \in \mathcal{E} \\ 0 & \text{sinon.} \end{cases}$$

Si la capacité d'un arc est définie deux fois par cette définition (si les deux arcs (a_1, a_2) et (a_2, a_1) existent), alors on somme les deux capacités. La contrainte de capacité assure que les capacités ainsi définies sont positives ou nulles. On recommence alors le procédé, en recherchant un nouveau chemin liant la source et le puits suivant des arcs de capacité $c_{\mathcal{G}_{\Phi}}$ strictement positive (chemin augmentant). Si un tel chemin existe, alors on définit le flot correspondant et on construit le graphe résiduel associé; sinon, alors la somme des flots définis précédemment donne le flot maximal et la coupure minimale est donnée par $(\mathcal{V}^s, \mathcal{V}^t)$, où \mathcal{V}^s est défini comme l'ensemble des sommets a de \mathcal{G} pour lesquels il existe un chemin de capacité strictement positive $c_{\mathcal{G}_{\Phi}}$ reliant la source s au sommet a . L'ensemble \mathcal{V}^t est alors donné par $\mathcal{V} \setminus \mathcal{V}^s$.

Les algorithmes basés sur le principe des flots augmentants sont généralement désignés sous le nom d'algorithmes de type FORD-FULKERSON [5]. Leur complexité dans le pire des cas est en $O(mn^2)$, où n est le nombre de sommets dans le graphe et m le nombre d'arcs [3].

4.2.2 Représentabilité d'une énergie

On reprend ici une partie de la théorie de la représentabilité des fonctions par un graphe, détaillée dans [9].

Représentation d'une fonction Soit E une fonction de n variables binaires $x = \{x_i\}_{i \in \llbracket 1; n \rrbracket} \in \{0,1\}^n$. On dit que G est représentable par un graphe s'il existe un graphe $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ comportant une source s et un puits t , ainsi que n sommets $\{a_i\}_{i \in \llbracket 1; n \rrbracket} \subset \mathcal{V}$ tels que, pour tout $x \in \{0,1\}^n$, la quantité $E(x)$ soit égale à une constante C plus le coût minimal des coupures $(\mathcal{V}^s, \mathcal{V}^t)$ vérifiant $a_i \in \mathcal{V}^s$ si $x_i = 0$ et $a_i \in \mathcal{V}^t$ si $x_i = 1$. On dit alors que le graphe \mathcal{G} représente la fonction E . Notons qu'il n'y a pas unicité de la représentation.

Si le nombre de sommets dans \mathcal{G} est minimal, exactement égal à $n + 2$, c'est-à-dire si $\mathcal{V} = \{s, t, \{a_i\}_{i \in \llbracket 1; n \rrbracket}\}$, alors $E(x)$ est égal à C plus le coût de la coupure $(\mathcal{V}^s, \mathcal{V}^t)$ vérifiant $a_i \in \mathcal{V}^s$ si $x_i = 0$ et $a_i \in \mathcal{V}^t$ si $x_i = 1$.

La classe \mathcal{F}^2 Soit n un entier naturel non nul. On définit la classe \mathcal{F}^2 comme étant l'ensemble des fonctions à n variables binaires s'écrivant comme la somme de fonctions à au plus deux variables binaires :

$$\forall x \in \{0,1\}^n, \quad E(x) = \sum_{i=1}^n E^i(x_i) + \sum_{i=1}^n \sum_{j=1}^{i-1} E^{i,j}(x_i, x_j)$$

Les fonctions à une variable binaire étant toujours représentables par un graphe, et la représentabilité étant une propriété additive, le théorème suivant donne une condition suffisante sur les termes à deux variables pour assurer la représentabilité de toute fonction E :

Théorème 12 (Théorème \mathcal{F}^2 , [8, 9]) Soit E une fonction de la classe \mathcal{F}^2 , écrit sous la forme

$$\forall x \in \{0,1\}^n, \quad E(x) = \sum_{i=1}^n E^i(x_i) + \sum_{i=1}^n \sum_{j=1}^{i-1} E^{i,j}(x_i, x_j)$$

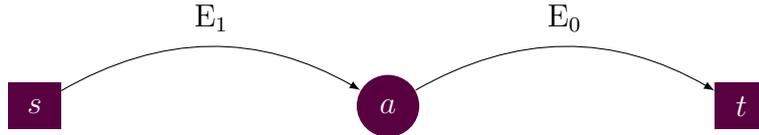
Si pour tout couple $(i, j) \in \llbracket 1; n \rrbracket^2$ tel que $i < j$, le terme $E^{i,j}$ vérifie la condition de sous-modularité

$$E^{i,j}(0,0) + E^{i,j}(1,1) \leq E^{i,j}(0,1) + E^{i,j}(1,0),$$

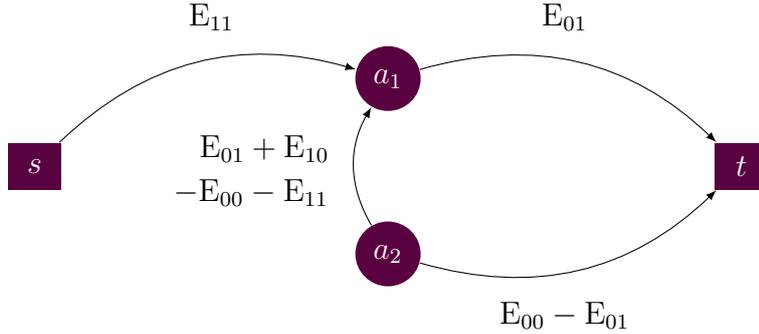
alors la fonction E est représentable par un graphe.

Pour démontrer ce théorème, on construit pour chaque terme un graphe le représentant. Dans [6], on montre que cette condition est en réalité nécessaire.

Représentation d'une fonction d'une variable Pour représenter la fonction E à une variable binaire, il suffit que de poser $E_0 = E(0)$ et $E_1 = E(1)$. Si ces deux valeurs sont positives, alors il suffit de considérer un sommet a (en plus des deux sommets s et t) et de construire les arcs (s, a) et (a, t) , en leur attribuant respectivement les capacités positives E_0 et E_1 (voir figure 4.1(a)). On vérifie que ce graphe représente bien la fonction E , car si $x = 0$, alors $E(x) = E_0$ et la coupure associée est la coupure $(\{s, a\}, \{t\})$, dont le coût vaut E_0 ; si $x = 1$, alors $E(x) = E_1$ et la coupure associée



(a) Fonction d'une variable



(b) Fonction de deux variables

FIGURE 4.1 – Représentation des fonctions à une et deux variables binaires. On suppose que, dans (b), la fonction vérifie la condition de sous-modularité.

est la coupure $(\{s\}, \{t, a\})$, dont le coût vaut E_1 (voir figure 4.2(a)). La constante C de la définition est ici nulle.

Dans le cas général, si une des deux quantités E_0 ou E_1 est négative par exemple (mais cela reste valable si elles sont positives toutes les deux), il suffit de considérer la fonction positive $E - \min\{E_0, E_1\}$, qui est représentable par un graphe d'après le paragraphe précédent. Les différentes coupures sont alors de coûts respectifs $E(x) - \min\{E_0, E_1\}$, ce qui montre que la fonction E est représentable par ce même graphe, avec cette fois $C = -\min\{E_0, E_1\}$. On notera que, dans ce cas, l'un des arcs construits est de capacité nulle, ce qui permet de l'ignorer.

Représentation d'une fonction de deux variables sous-modulaire On suppose que la fonction à deux variables E vérifie la condition de sous-modularité. Pour tout $(a, b) \in \{0, 1\}^2$, notons $E_{ab} = E(a, b)$. On a alors

$$E_{00} + E_{11} \leq E_{01} + E_{10}.$$

Remarquons ensuite que E peut se décomposer de la manière suivante :

$$\begin{aligned}
 E(x_1, x_2) = & \begin{cases} E_{01} & \text{si } x_1 = 0 \\ E_{11} & \text{si } x_1 = 1 \end{cases} \\
 & + \begin{cases} E_{00} - E_{01} & \text{si } x_2 = 0 \\ 0 & \text{si } x_2 = 1 \end{cases} \\
 & + \begin{cases} E_{01} + E_{10} - E_{00} - E_{11} & \text{si } (x_1, x_2) = (1, 0) \\ 0 & \text{sinon.} \end{cases}
 \end{aligned} \tag{4.5}$$

Le premier terme définit une fonction ne dépendant que de la variable binaire x_1 , tandis que le second ne dépend que de la variable x_2 . Aussi, d'après le point précédent,

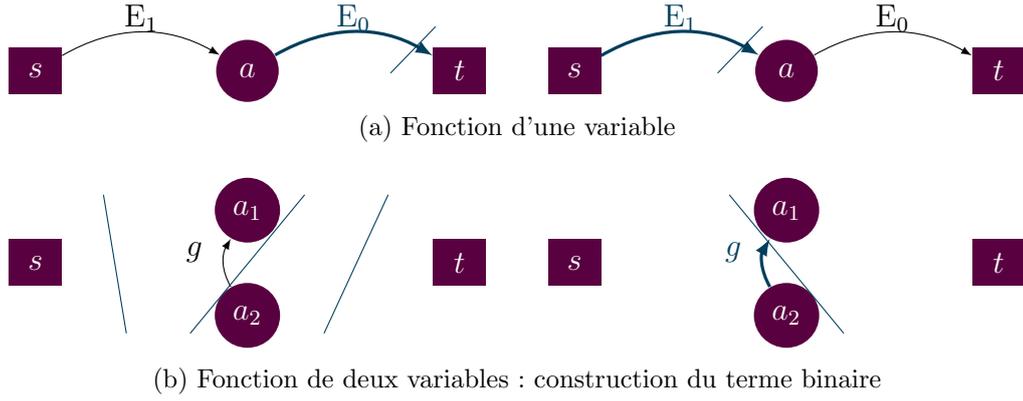


FIGURE 4.2 – Les différentes coupures possibles dans les graphes construits à la figure 4.1. Dans (b), on ne considère que la représentation du troisième terme dans 4.5. Les coupures sont symbolisées par les lignes bleues, tandis que les arcs bleus sont ceux contribuant au coût de la coupure.

ces deux termes sont représentables par un graphe. Le dernier terme est une fonction à deux variables binaires à valeurs positives d'après la condition de sous-modularité. Pour la représenter par un graphe, il suffit de construire l'arc (a_2, a_1) en la pondérant par la quantité positive $g = E_{01} + E_{10} - E_{00} - E_{11}$, où les sommets a_1 et a_2 sont respectivement associés aux variables x_1 et x_2 . La construction d'un tel graphe est proposée dans la figure 4.1(b). On peut alors vérifier que la seule coupure de coût non nul est la coupure $(\{s, a_2\}, \{t, a_1\})$, et qu'elle est de coût g (voir figure 4.2(b)).

4.3 Décroissance de l'énergie par *expansion move* optimal

La méthode de KOLMOGOROV et ZABIH ne cherche pas à minimiser la fonctionnelle d'énergie (non convexe) E car aucun algorithme efficace n'est connu pour cette tâche, le problème étant NP-difficile. À la place, elle essaie de la faire décroître, en introduisant des *expansion moves*. On montrera qu'une décroissance optimale peut être obtenue grâce à la représentation de l'énergie par un graphe.

4.3.1 *Expansion moves* et décroissance de l'énergie

Expansion moves Soit $u^h \in (I_{\text{disp}} \times \{\text{occ}\})^{N_x N_y}$ une carte de disparité et $\alpha \in I_{\text{disp}}^h$ une valeur de disparité admissible. Alors on dit que $(u^h)' \in (I_{\text{disp}} \times \{\text{occ}\})^{N_x N_y}$ est un α -*expansion move* de u^h si

$$\forall (i, j) \in \Omega^h, \quad \begin{cases} (u^h)'_{i,j} = \alpha & \text{si } u^h_{i,j} = \alpha \\ (u^h)'_{i,j} \in \{u^h_{i,j}, \alpha, \text{occ}\} & \text{si } u^h_{i,j} \neq \alpha. \end{cases}$$

En particulier, u^h est un α -*expansion move* de lui-même quel que soit α . Pour un α donné, tout α -*expansion move* de u^h possède davantage de pixels de disparité α que la carte u^h . Chaque pixel de disparité différente de α peut soit devenir occulté, soit adopter la disparité α , soit conserver sa disparité. Enfin, chaque pixel occulté peut soit rester occulté, soit adopter la disparité α . Par conséquent, l'ensemble des pixels occultés peut croître, mais pas les ensembles des pixels de disparité différente de α .

Décroissance de l'énergie Soit u^h une carte de disparité donnée, d'énergie $E(u^h)$ supposée finie. Pour tout $\alpha \in I_{\text{disp}}^h$ donné, on considère l'ensemble des α -*expansion move* de u^h . Puisque cet ensemble est de cardinal fini, il existe un élément d'énergie minimale, noté u_α^h . Par ailleurs, cet ensemble contient u^h , donc $E(u_\alpha^h) \leq E(u^h)$.

Le principe de la décroissance de l'énergie par *expansion moves* est de partir d'une carte de disparité u^h , puis, pour une valeur de $\alpha \in I_{\text{disp}}^h$, de la mettre à jour en la remplaçant par u_α^h l' α -*expansion move* de u^h d'énergie minimale

$$u_\alpha^h = \underset{(u^h)' \text{ } \alpha\text{-expansion move de } u^h}{\operatorname{argmin}} E((u^h)').$$

Une *itération* de l'algorithme d'*expansion moves* consiste à répéter cette opération pour toutes les valeurs de I_{disp}^h , en mettant à jour la carte de départ entre chaque minimisation. Cela conduit à l'algorithme 1. Le critère d'arrêt est choisi de la manière suivante : lorsque l'énergie n'a pas décréu sur une itération complète de tous les α (*expansion moves*) appliqués successivement, alors l'algorithme s'arrête.

Algorithme 1: Décroissance de l'énergie par *expansion moves* : une itération

Entrée : u_0^h une carte de disparité initiale, I_{disp}^h l'intervalle de disparité

Sortie : u^h une carte de disparité d'énergie plus petite

```

1 begin
2    $u^h \leftarrow u_0^h$ 
3   foreach valeur de disparité  $\alpha \in I_{\text{disp}}^h$  (dans un ordre arbitraire) do
4      $u^h \leftarrow \underset{(u^h)' \text{ } \alpha\text{-expansion move de } u^h}{\operatorname{argmin}} E((u^h)').$ 

```

L'algorithme proposé ne minimise donc pas la fonctionnelle E , mais chaque pas de minimisation considéré est choisi de manière optimale. Si la fonctionnelle d'énergie ne possède pas de minima locaux, alors cette approche permet de retrouver un minimiseur global. Sinon, il est possible de trouver un minimiseur local. Néanmoins, ce risque est plus faible comparé à une approche de type descente de gradient, alors la décroissance de l'énergie se fait en considérant un sous-ensemble de cartes de disparité qui ne se situe pas nécessairement dans un voisinage de la carte de départ.

On se propose dans ce qui suit d'utiliser la représentabilité des énergies par un graphe pour résoudre de manière optimale la recherche de l'*expansion move* d'énergie minimale. Pour cela, on montre que l'on peut réécrire ce problème à l'aide d'une énergie représentable par un graphe, en effectuant un changement de variable. Cette section reprend l'étude proposée dans [7].

4.3.2 Assignements et configurations

On introduit ici un premier changement de variable qui permettra de rendre le problème de la décroissance d'énergie par *expansion moves* représentable par des graphes.

Assignements Au lieu de considérer les couples $((i,j), u_{i,j}^h)$ formés par les pixels de l'image de référence et leur disparité, on choisit une représentation symétrique sur les deux images de la paire en considérant les couples (p,q) de pixels, avec p appartenant à l'image de référence (dont l'ensemble des pixels est noté \mathcal{I}_L) et q appartenant à

l'image de droite (dont l'ensemble des pixels est noté \mathcal{I}_R). Parmi ces couples, on appelle *assignement* tout couple vérifiant $q - p \in I_{\text{disp}}^h \times \{0\}$. Un assignement est donc un couple de deux pixels situés sur la même ligne, dont la différence sur la ligne appartient à l'intervalle de disparité. Autrement dit, tout couple de pixels homologues est un assignement. L'ensemble des assignements est noté \mathcal{A} :

$$\mathcal{A} = \left\{ a = (p, q) \in \mathcal{I}_L \times \mathcal{I}_R \mid q - p \in I_{\text{disp}}^h \times \{0\} \right\}.$$

Notons en particulier que si $(p, p + \alpha)$ appartient à l'ensemble des assignements, alors α est une valeur de disparité admissible et $p + \alpha$ est un pixel de l'image de droite.

Configuration Pour toute carte de disparité u^h , si $p = (i, j) \in \mathcal{I}_L$ n'est pas occulté, alors il est mis en correspondance avec le pixel $q = (i, j - u_{i,j}^h) \in \mathcal{I}_R$. On dit alors que l'assignement $a = (p, q)$ est *actif*. Pour tout assignement (p', q') , si p' n'est pas mis en correspondance avec q' , alors l'assignement est dit *inactif*.

On peut alors définir la *configuration* associée à la carte de disparité u^h , qui est une fonction f_{u^h} qui à tout assignement actif associe la valeur 1 et à tout assignement inactif associe la valeur 0 :

$$\forall a = (p, q) \in \mathcal{A}, \quad f_{u^h}(a) = \begin{cases} 1 & \text{si } p = (i, j) \quad \text{et} \quad q = (i, j + u_{i,j}^h) \\ 0 & \text{sinon.} \end{cases}$$

Toute fonction $f : \mathcal{A} \rightarrow \{0, 1\}$ définit alors une configuration, et on appelle *état* d'un assignement son image par la configuration f . On note $\mathcal{A}^\circ(f)$ l'ensemble des assignements actifs sous la configuration f :

$$\mathcal{A}^\circ(f) = \left\{ a \in \mathcal{A} \mid f(a) = 1 \right\}.$$

Disparité Enfin, pour tout assignement $a = (p, q) \in \mathcal{A}$ (actif ou inactif), on définit sa disparité $d(a) = q - p$, que l'on confondra avec son abscisse qui appartient par définition à I_{disp}^h (son ordonnée est nulle). Si $a = (p, q)$ est un assignement actif, alors p et q sont des pixels homologues et la disparité de $p = (i, j)$ dans l'image de gauche vaut $u_{i,j}^h = u_{I_L}^h(p) = d(a)$. Celle de q dans l'image de droite vaut $u_{I_R}^h(q) = -d(a)$.

Notons que la carte de disparité est entièrement donnée par la configuration associée, grâce à la formule de reconstruction

$$\forall p = (i, j) \in \mathcal{I}_L \quad u_{i,j}^h = \begin{cases} d(a) & \text{si } \exists a = (p, q) \in \mathcal{A}^\circ(f_{u^h}) \\ \text{occ} & \text{sinon.} \end{cases}$$

Cette formule de reconstruction est bien définie si la configuration satisfait la contrainte d'injectivité³, c'est-à-dire si, pour tout pixel $p \in \mathcal{I}_L$ (respectivement $q \in \mathcal{I}_R$), il existe au plus un assignement actif de premier élément p (resp. de second élément q).

Pour toute valeur de disparité $\alpha \in I_{\text{disp}}^h$, on note \mathcal{A}^α l'ensemble des assignements de disparité α :

$$\mathcal{A}^\alpha = \left\{ a \in \mathcal{A} \mid d(a) = \alpha \right\}.$$

On notera que cet ensemble ne dépend pas de la configuration choisie. Par ailleurs, une remarque préliminaire assure que si $(p, p + \alpha) \in \mathcal{A}^\alpha$, alors $p + \alpha$ est un pixel de l'image de droite.

3. *Uniqueness constraint* dans [8, 7].

4.3.3 Énergie d'une configuration

On va à présent réaliser un changement de variables dans la définition de l'énergie E pour définir l'énergie d'une configuration. Pour cela, on va construire pour chaque terme de l'énergie E un terme correspondant défini sur l'ensemble des configurations, de sorte que $E(f_{u^h}) = E(u^h)$.

Terme d'attache aux données et d'occultation Le terme d'attache aux données (4.1) n'est défini que sur les pixels p non occultés, c'est-à-dire ceux pour lesquels il existe un assignement (p,q) actif. On verra par la suite que la contrainte d'injectivité assure que cet assignement est unique, aussi on peut définir le nouveau terme d'attache aux données sur les assignements actifs sous la configuration f :

$$E_{\text{data}}(f) = \sum_{a=(p,q) \in \mathcal{A}^\circ(f)} g(p, d(a)).$$

Pour redéfinir le terme d'occultation (4.2), il faut remarquer qu'un pixel p est occulté si tous les assignements de la forme (p,q) sont inactifs. Or, le terme d'injectivité assure qu'il existe au plus un tel assignement actif. Ainsi, un pixel est occulté si

$$\sum_{a=(p,q) \in \mathcal{A}} \mathbb{1}_{\{1\}}(f(a)) = 0$$

et sinon, cette quantité vaut 1. On en déduit le terme d'occultation suivant

$$E_{\text{occ}}(f) = \sum_{p \in \mathcal{I}_L} K \left(1 - \sum_{a=(p,q) \in \mathcal{A}} \mathbb{1}_{\{1\}}(f(a)) \right) = K N_x N_y - \sum_{a \in \mathcal{A}^\circ(f)} K.$$

Si on somme ces deux termes, on peut définir un coût $D(a)$ pour chaque assignement a en posant

$$\forall a = (p,q) \in \mathcal{A}, \quad D(a) = g(p, d(a)) - K$$

de sorte que la somme du terme d'attache aux données et d'occultation s'écrive

$$E_{\text{data+occ}}(f) = \sum_{a \in \mathcal{A}^\circ(f)} D(a) + \text{constante}$$

où la constante vaut $K N_x N_y$. On remarque que ce terme est défini sur les assignements actifs seulement. On vérifie que

$$E_{\text{data+occ}}(f_{u^h}) = E_{\text{data+occ}}(u^h).$$

Terme de régularisation Définissons à présent le nouveau terme de régularisation. Dans (4.3), une pénalité est ajoutée à chaque fois que deux pixels voisins (tous deux non occultés) p et p' ont une disparité différente. Commençons par introduire la notation suivante pour tous assignements $a = (p,q)$ et $a' = (p',q')$

$$a \sim a' \quad \text{si} \quad d(a) = d(a') \quad \text{et} \quad \begin{cases} p' = p + (0, \pm 1) \\ p' = p + (\pm 1, 0) \end{cases}$$

pour des assignements qualifiés de *voisins*. Deux assignements sont voisins s'ils ont même disparité et si leurs premiers pixels le sont dans l'image de référence ou que leurs seconds pixels le sont dans l'image de droite. On notera que, puisque $d(a) = d(a')$, on obtient les mêmes conditions sur q et q' .

Soient deux pixels p et p' voisins dans l'image de gauche. Si les deux sont occultés ou qu'ils ont même disparité, alors le couple (p,p') ne contribue pas au terme de régularisation. Or, dans le premier cas, on a

$$\forall a = (p,q) \in \mathcal{A}, \quad \forall a' = (p',q') \in \mathcal{A}, \quad f(a) = f(a') = 0.$$

En particulier, pour tous $a \sim a'$, on a $f(a) = f(a') = 0$. Dans le second cas, le seul assignement actif de la forme (p,q) et le seul assignement actif de la forme (p',q') ont même disparité, puisque p et p' ont même disparité (et les autres assignements de cette forme sont inactifs). On en déduit que $f(a) = f(a')$ pour tous assignements $a = (p,q)$ et $a' = (p',q')$.

Supposons maintenant que les deux pixels ne sont pas occultés, mais de disparité différente. Alors le couple (p,p') contribue deux fois au terme de régularisation, une fois pour la disparité $u_{I_L}^h(p)$ et une seconde fois pour la disparité $u_{I_L}^h(p')$. Or, cette situation impose que l'assignement actif de la forme (p,q) et l'assignement actif de la forme (p',q') soient de disparité différente. Plus précisément, les assignements $(p,p+u_{I_L}^h(p))$ et $(p',p'+u_{I_L}^h(p'))$ sont actifs mais pas les assignements $(p,p+u_{I_L}^h(p'))$ et $(p',p'+u_{I_L}^h(p))$. Autrement dit,

$$(p,p+u_{I_L}^h(p)) \sim (p',p'+u_{I_L}^h(p)) \quad \text{et} \quad f(p,p+u_{I_L}^h(p)) \neq f(p',p'+u_{I_L}^h(p))$$

et

$$(p',p'+u_{I_L}^h(p')) \sim (p,p+u_{I_L}^h(p')) \quad \text{et} \quad f(p',p'+u_{I_L}^h(p')) \neq f(p,p+u_{I_L}^h(p'))$$

Supposons à présent que l'un des deux pixels (p' par exemple) est occulté (et que l'autre ne l'est pas). Alors le couple (p,p') contribue une fois au terme de régularisation (pour la disparité $u_{I_L}^h(p)$ du pixel non occulté) et on a cette fois

$$(p,p+u_{I_L}^h(p)) \sim (p',p'+u_{I_L}^h(p)) \quad \text{et} \quad f(p,p+u_{I_L}^h(p)) \neq f(p',p'+u_{I_L}^h(p))$$

puisque l'assignement $(p',p'+u_{I_L}^h(p))$ est nécessairement inactif.

On en déduit que le couple de pixels voisins (p,p') contribue au terme de régularisation à chaque fois qu'il existe des assignements $a = (p,q)$ et $a' = (p',q')$ voisins qui n'ont pas le même état. Cela nous conduit à proposer le terme de régularisation suivant

$$E_{\text{reg}}(f) = \frac{1}{2} \sum_{\substack{(a,a') \in \mathcal{A}^2 \\ a \sim a'}} \tilde{R}_{\lambda_1, \lambda_2}(a, a', f(a), f(a'))$$

$$\text{avec } \tilde{R}_{\lambda_1, \lambda_2}(a, a', f(a), f(a')) = \begin{cases} 0 & \text{si } f(a) = f(a') \\ R_{\lambda_1, \lambda_2}(p, p', d(a)) & \text{si } f(a) \neq f(a') \text{ et } f(a) = 1 \\ R_{\lambda_1, \lambda_2}(p, p', d(a')) & \text{si } f(a) \neq f(a') \text{ et } f(a') = 1 \end{cases}$$

Le facteur $1/2$ est nécessaire car les couples d'assignements (a,a') et (a',a) contribuent tous deux à ce terme. On a alors

$$E_{\text{reg}}(f_{u^h}) = E_{\text{reg}}(u^h).$$

Terme d'injectivité Le terme d'injectivité (4.4) impose que, si p est mis en correspondance avec le pixel q , alors il est le seul. Autrement dit, si l'assignement (p,q) est actif, alors tous les assignements (p',q) avec $p' \neq p$ doivent être inactifs. Par ailleurs, il est évident que tout pixel de l'image de gauche ne peut être mis en correspondance avec au plus un pixel. En termes d'assignements, cela se traduit de la manière suivante : si

l'assignement (p,q) est actif, alors tous les assignements (p,q') avec $q' \neq q$ doivent être inactifs. On en déduit le terme d'injectivité suivant

$$E_{\text{inj}}(f) = \sum_{a=(p,q) \in \mathcal{A}^\circ(f)} \sum_{\substack{a'=(p',q) \\ p' \neq p}} \chi_{\{0\}}(f(a')) + \sum_{a=(p,q) \in \mathcal{A}^\circ(f)} \sum_{\substack{a'=(p,q') \\ q' \neq q}} \chi_{\{0\}}(f(a')).$$

Ce terme est nul ou prend une valeur infinie et on vérifie à nouveau que

$$E_{\text{inj}}(f_{u^h}) = E_{\text{inj}}(u^h).$$

Finalement, on a construit une énergie sur l'ensemble des configurations

$$E(f) = E_{\text{data+occ}}(f) + E_{\text{reg}}(f) + E_{\text{inj}}(f)$$

vérifiant l'identité $E(u^h) = E(f_{u^h})$ pour toute carte de disparité.

REMARQUE : En réécrivant l'énergie à l'aide d'assignements, on rétablit la symétrie du problème de mise en correspondance, en rendant les deux images (gauche et droite) indifférenciées. En particulier, ce traitement nécessite de manipuler des images échantillonnées sur une même grille, afin de conserver cette symétrie.

4.3.4 Énergie d'un *expansion move*

Expansion move Soit f une configuration d'énergie finie et $\alpha \in I_{\text{disp}}^h$ une valeur de disparité. On dit que f' est un α -*expansion move* de la configuration f si sa carte de disparité associée est un α -*expansion move* de la carte de disparité associée à la configuration f .

D'après la définition des α -*expansion moves* sur les cartes de disparité, on en déduit que

$$\forall a \in \mathcal{A}, \quad f'(a) = \begin{cases} 1 & \text{si } f(a) = 1 \text{ et } d(a) = \alpha \\ 0 & \text{si } f(a) = 0 \text{ et } d(a) \neq \alpha \end{cases}.$$

En d'autres termes, tout assignement actif de disparité α reste actif après un α -*expansion move* (tout pixel de disparité α conserve sa disparité) et tout assignement inactif de disparité différente de α le reste (les pixels occultés ou de disparité différente de α ne peuvent pas adopter une nouvelle disparité, mais peuvent devenir occultés ou adopter la disparité α).

Changement d'état Remarquons qu'il est possible de différencier le comportement d'un assignement pendant un α -*expansion move* suivant l'ensemble auquel il appartient :

1. l'ensemble $\mathcal{A}^\circ(f) \cap \mathcal{A}_\alpha$ des assignements actifs de disparité α ;
2. l'ensemble $\mathcal{A}_\alpha \setminus \mathcal{A}^\circ(f)$ des assignements inactifs de disparité α ;
3. l'ensemble $\mathcal{A}^\circ(f) \setminus \mathcal{A}_\alpha$ des assignements actifs de disparité différente de α ;
4. l'ensemble $\mathcal{A} \setminus (\mathcal{A}^\circ \cup \mathcal{A}_\alpha)$ des assignements inactifs de disparité différente de α .

Les assignements du premier et du dernier ensemble ne changent pas d'état après un α -*expansion move*. Plus précisément, on a

$$\forall a \in \mathcal{A}, \quad f'(a) = \begin{cases} 1 & \text{si } a \in \mathcal{A}^\circ \text{ et } a \in \mathcal{A}_\alpha \\ 0 & \text{si } a \notin \mathcal{A}^\circ \text{ et } a \notin \mathcal{A}_\alpha. \end{cases}$$

Le second ensemble ne peut que croître tandis que le troisième ne peut que décroître.

Énergie Nous allons à présent exprimer l'énergie d'un α -*expansion move* f' en fonction de l'énergie de la configuration initiale f . Pour cela, on utilise les remarques précédentes pour distinguer les termes variables de ceux qui restent constants, car impliquant des assignements dont l'état reste invariable après α -*expansion move*. Commençons par le terme d'attache aux données et d'occultation. Celui-ci est défini sur les assignements actifs de f' . Or, les seuls assignements f' qui peuvent être actifs sont ceux qui l'étaient déjà dans f , et les assignements de disparité α , en gardant en tête que, parmi ces derniers, ceux déjà actifs dans la configuration initiale f le restent (leur contribution à ce terme ne change donc pas). On peut donc le réécrire en utilisant les trois premiers ensembles introduits plus haut :

$$E_{\text{data+occ}}(f') = \sum_{\substack{a \notin \mathcal{A}^\circ(f) \\ a \in \mathcal{A}_\alpha}} D(a) \mathbb{1}_{\{1\}}(f'(a)) + \sum_{\substack{a \in \mathcal{A}^\circ(f) \\ a \notin \mathcal{A}_\alpha}} D(a) \mathbb{1}_{\{1\}}(f'(a)) + \text{constante}.$$

Pour le terme de régularisation, on commence par décomposer la somme sur les assignements voisins de disparité α et les autres, puisqu'on rappelle que deux assignements voisins ont par définition même disparité :

$$E_{\text{reg}}(f') = \frac{1}{2} \sum_{\substack{(a,a') \in \mathcal{A}^2 \\ a \sim a' \\ d(a) = \alpha}} \tilde{R}_{\lambda_1, \lambda_2}(a, a', f'(a), f'(a')) + \frac{1}{2} \sum_{\substack{(a,a') \in \mathcal{A}^2 \\ a \sim a' \\ d(a) \neq \alpha}} \tilde{R}_{\lambda_1, \lambda_2}(a, a', f'(a), f'(a')).$$

Ensuite, pour chaque terme, on distingue selon l'état initial des assignements a et a' . On rappelle qu'un assignement a' actif de disparité α le reste, de même qu'un assignement a' inactif de disparité différente de α . Par ailleurs, un couple (a, a') ne contribue à ce terme que si les deux états associés sont différents. En particulier, si a et a' sont actifs sous la configuration f et de disparité α , alors le couple (a, a') ne contribue pas plus au terme de régularité pour la configuration f que pour la configuration f' (*idem* pour deux assignements voisins inactifs sous f de disparité différente de α). On peut donc les ignorer, et on obtient alors

$$\begin{aligned} E_{\text{reg}}(f') &= \sum_{\substack{a \in \mathcal{A}^\circ(f) \\ a' \notin \mathcal{A}^\circ(f) \\ a \sim a' \\ d(a) = \alpha}} \tilde{R}_{\lambda_1, \lambda_2}(a, a', 1, f'(a')) + \frac{1}{2} \sum_{\substack{a \notin \mathcal{A}^\circ(f) \\ a' \notin \mathcal{A}^\circ(f) \\ a \sim a' \\ d(a) = \alpha}} \tilde{R}_{\lambda_1, \lambda_2}(a, a', f'(a), f'(a')) \\ &+ \frac{1}{2} \sum_{\substack{a \in \mathcal{A}^\circ(f) \\ a' \in \mathcal{A}^\circ(f) \\ a \sim a' \\ d(a) \neq \alpha}} \tilde{R}_{\lambda_1, \lambda_2}(a, a', f'(a), f'(a')) + \sum_{\substack{a \notin \mathcal{A}^\circ(f) \\ a' \in \mathcal{A}^\circ(f) \\ a \sim a' \\ d(a) \neq \alpha}} \tilde{R}_{\lambda_1, \lambda_2}(a, a', 0, f'(a')). \end{aligned}$$

Notons que, dès que l'on supprime la symétrie dans le rôle des assignements a et a' , le facteur $1/2$ disparaît. Enfin, le terme d'injectivité étant nul pour f car celle-ci est supposée d'énergie finie, il est soit nul pour f' si la contrainte d'injectivité est respectée, soit infini si l'activation d'un assignement viole cette contrainte. Or, les seuls assignements à pouvoir être activés sont les assignements (p, q) de disparité α inactifs pour f , et cette activation viole la contrainte d'injectivité que s'il existait auparavant un assignement actif (p, q') ou (p', q) de disparité différente de α , et que celui-ci n'est

pas désactivé dans f' :

$$\begin{aligned} E_{\text{inj}}(f') &= \sum_{\substack{a=(p,q) \in \mathcal{A}^\circ(f) \\ a \notin \mathcal{A}_\alpha \\ (p,p+\alpha) \in \mathcal{A}_\alpha}} \chi_{\{(0,0),(0,1),(1,0)\}}(f'(p,p+\alpha), f'(a)) \\ &+ \sum_{\substack{a=(p,q) \in \mathcal{A}^\circ(f) \\ a \notin \mathcal{A}_\alpha \\ (q-\alpha,q) \in \mathcal{A}_\alpha}} \chi_{\{(0,0),(0,1),(1,0)\}}(f'(q-\alpha,q), f'(a)). \end{aligned}$$

4.3.5 Représentabilité de l'énergie des *expansion moves*

Soit f une configuration et $\alpha \in \mathbb{I}_{\text{disp}}^h$ une valeur de disparité. On rappelle qu'on cherche à trouver l' α -*expansion move* de f d'énergie minimale, c'est-à-dire de résoudre le problème

$$f_\alpha = \underset{f' \text{ } \alpha\text{-expansion move de } f}{\text{argmin}} \quad E(f'). \quad (4.6)$$

Malheureusement, cette énergie n'est pas représentable par un graphe. On va donc introduire un nouveau changement de variable pour la rendre représentable. Ce changement de variable permettra en outre de supprimer la contrainte dans le problème (4.6), en encodant de manière naturelle les α -*expansion moves* de f .

Vecteur de changement d'état On choisit d'introduire le changement de variable suivant, qui encode tout changement d'état entre la configuration initiale f et l' α -*expansion move* f' :

$$\forall a \in (\mathcal{A}^\circ(f) \cup \mathcal{A}_\alpha) \setminus (\mathcal{A}^\circ(f) \cap \mathcal{A}_\alpha), \quad g'(a) = \begin{cases} 1 & \text{si } f(a) \neq f'(a) \\ 0 & \text{sinon.} \end{cases} \quad (4.7)$$

On choisit d'ignorer les assignements dont l'état reste invariable. Tout α -*expansion move* f' définit un vecteur $g' \in \{0,1\}^N$, avec $N = \text{card}((\mathcal{A}^\circ(f) \cup \mathcal{A}_\alpha) \setminus (\mathcal{A}^\circ(f) \cap \mathcal{A}_\alpha))$, tandis que tout tel vecteur g' définit de manière unique un α -*expansion move* f' , grâce à la formule d'inversion suivante :

$$\forall a \in \mathcal{A}, \quad f'(a) = \begin{cases} 1 - g'(a) & \text{si } a \in \mathcal{A}^\circ(f) \text{ et } a \notin \mathcal{A}_\alpha \\ g'(a) & \text{si } a \notin \mathcal{A}^\circ(f) \text{ et } a \in \mathcal{A}_\alpha \\ f(a) & \text{sinon.} \end{cases} \quad (4.8)$$

Une nouvelle énergie On va à présent proposer une nouvelle énergie $E^{\alpha,f}$ en appliquant le changement de variable proposé plus haut. Ainsi, l'identité $E^{\alpha,f}(g') = E(f')$ est vérifiée (à une constante près) pour tout α -*expansion move* f' . Le terme d'attache aux données et d'occultation est donné par :

$$E_{\text{data+occ}}^{\alpha,f}(g') = \sum_{\substack{a \notin \mathcal{A}^\circ(f) \\ a \in \mathcal{A}_\alpha}} D(a) \mathbb{1}_{\{1\}}(g'(a)) + \sum_{\substack{a \in \mathcal{A}^\circ(f) \\ a \notin \mathcal{A}_\alpha}} D(a) \mathbb{1}_{\{1\}}(g'(a)).$$

On obtient ensuite pour le terme de régularisation

$$\begin{aligned} E_{\text{reg}}^{\alpha,f}(g') &= \sum_{\substack{a \in \mathcal{A}^\circ(f) \\ a' \notin \mathcal{A}^\circ(f) \\ a \sim a' \\ d(a) = \alpha}} \tilde{R}_{\lambda_1, \lambda_2}(a, a', 1, g'(a')) + \sum_{\substack{a \notin \mathcal{A}^\circ(f) \\ a' \in \mathcal{A}^\circ(f) \\ a \sim a' \\ d(a) \neq \alpha}} \tilde{R}_{\lambda_1, \lambda_2}(a, a', 0, 1 - g'(a')) \\ &+ \frac{1}{2} \sum_{\substack{a \notin \mathcal{A}^\circ(f) \\ a' \notin \mathcal{A}^\circ(f) \\ a \sim a' \\ d(a) = \alpha}} \tilde{R}_{\lambda_1, \lambda_2}(a, a', g'(a), g'(a')) + \frac{1}{2} \sum_{\substack{a \in \mathcal{A}^\circ(f) \\ a' \in \mathcal{A}^\circ(f) \\ a \sim a' \\ d(a) \neq \alpha}} \tilde{R}_{\lambda_1, \lambda_2}(a, a', 1 - g'(a), 1 - g'(a')) \end{aligned}$$

et enfin pour le terme d'injectivité :

$$\begin{aligned} E_{\text{inf}}^{\alpha,f}(g') &= \sum_{\substack{a=(p,q) \in \mathcal{A}^\circ(f) \\ a \notin \mathcal{A}_\alpha \\ (p, p+\alpha) \in \mathcal{A}_\alpha}} \chi_{\{(0,0), (1,0), (1,1)\}}(g'(p, p+\alpha), g'(a)) \\ &+ \sum_{\substack{a=(p,q) \in \mathcal{A}^\circ(f) \\ a \notin \mathcal{A}_\alpha \\ (q-\alpha, q) \in \mathcal{A}_\alpha}} \chi_{\{(0,0), (1,0), (1,1)\}}(g'(q-\alpha, q), g'(a)). \end{aligned}$$

On vérifie alors que, si on pose

$$E^{\alpha,f}(g') = E_{\text{data+occ}}^{\alpha,f}(g') + E_{\text{reg}}^{\alpha,f}(g') + E_{\text{inf}}^{\alpha,f}(g')$$

alors on a l'identité $E^{\alpha,f}(g') = E(f') + \text{constante}$ pour tout α -*expansion move* de f , où la constante ne dépend pas de f . Ainsi, le problème (4.6) est équivalent au problème *non constraint* suivant

$$g_\alpha = \underset{g' \in \{0,1\}^N}{\text{argmin}} E^{\alpha,f}(g'). \quad (4.9)$$

Représentabilité de l'énergie Vérifions que l'énergie $E^{\alpha,f}$ est représentable par un graphe. Pour cela, on utilise le théorème 12, qui stipule qu'il suffit de vérifier si chaque terme dépendant de deux variables satisfait la contrainte de sous-modularité.

Le terme d'attache aux données ne possède que des termes dépendant d'une variable binaire, donc il est représentable. Passons au terme de régularisation. Les seuls termes dépendant de deux variables binaires sont ceux de la forme

$$E_1(x, x') = \tilde{R}_{\lambda_1, \lambda_2}(a, a', x, x')$$

lorsque a et a' sont inactifs sous la configuration initiale f , mais de disparité α et ceux de la forme

$$E_2(x, x') = \tilde{R}_{\lambda_1, \lambda_2}(a, a', 1 - x, 1 - x')$$

où a et a' sont actifs sous la configuration initiale f , mais de disparité différente de α . La définition de $\tilde{R}_{\lambda_1, \lambda_2}$ assure que $E_1(0,0) = E_1(1,1) = E_2(0,0) = E_2(1,1) = 0$ et que G_1 et G_2 sont positives, ce qui permet de vérifier la contrainte de sous-modularité. Enfin, les termes dans le terme d'injectivité sont de la forme

$$E_3(x, x') = \chi_{\{0\}}(x(1-x'))$$

qui est nul si $x = 0$ ou $x' = 1$, et infini si $(x, x') = (1,0)$, ce qui assure à nouveau la sous-modularité de ce terme.

Résolution du problème (4.9) On vient donc de démontrer que l'énergie $E^{\alpha,f}$ est représentable par un graphe $\mathcal{G}^{\alpha,f} = (\mathcal{V}, \mathcal{E})$. Par définition, si on choisit le graphe avec le nombre de sommets minimal, alors, si on ordonne les assignements de $(\mathcal{A}^\circ(f) \cup \mathcal{A}_\alpha) \setminus (\mathcal{A}^\circ(f) \cap \mathcal{A}_\alpha) = \{a_i\}_{i \in \llbracket 1; N \rrbracket}$, on a

$$\mathcal{V} = \left\{ s, t, \{a_i\}_{i \in \llbracket 1; N \rrbracket} \right\}$$

et $E^{\alpha,f}(g')$ est égal à C plus le coût de la coupure $(\mathcal{V}^s, \mathcal{V}^t)$ vérifiant $a_i \in \mathcal{V}^s$ si $g'(a_i) = 0$ et $a_i \in \mathcal{V}^t$ si $g'(a_i) = 1$. On en déduit en particulier que la valeur minimale de $E^{\alpha,f}$, qui vaut $E^{\alpha,f}(g_\alpha)$ est donnée par la coupure minimale du graphe $\mathcal{G}^{\alpha,f}$. Résoudre le problème (4.9) revient donc à trouver la coupure minimale du graphe $\mathcal{G}^{\alpha,f}$, et d'après le théorème de FORD-FULKERSON, cela revient à trouver le flot maximal dans ce même graphe.

Une fois la coupure minimale obtenue, les assignements associés aux sommets qui sont reliés à la source ne changent pas d'état, tandis que les autres changent d'état.

4.4 Résolution numérique par coupure de graphes

On présente ici les détails de l'implémentation de l'algorithme décrit dans les sections précédentes, dont le code initial a été écrit par KOLMOGOROV (et largement aménagé pour [7]). Il sera désigné par la suite sous le nom d'algorithme KZ2.

On rappelle que l'idée est de faire décroître l'énergie en calculant pour chaque α le meilleur α -*expansion move* de la configuration f obtenue à l'itération précédente (algorithme 1). Pour cela, on a montré qu'il suffisait de construire le graphe représentant l'énergie $E^{\alpha,f}$ puis d'en trouver la coupure minimale en en déterminant le flot maximal. Cela conduit à l'algorithme 2.

Algorithme 2: Recherche de l' α -*expansion move* optimal de f

Entrée : f une configuration initiale, $\alpha \in I_{\text{disp}}^h$ une valeur de disparité

Sortie : f_α l' α -*expansion move* de f d'énergie la plus faible

$$f_\alpha = \underset{f' \text{ } \alpha\text{-expansion move de } f}{\operatorname{argmin}} E(f')$$

1 **begin**

- | | |
|---|---|
| 2 | Construire le graphe $\mathcal{G}^{\alpha,f}$ |
| 3 | Calculer le flot maximal du graphe $\mathcal{G}^{\alpha,f}$ |
| 4 | En déduire la coupure maximale du graphe $\mathcal{G}^{\alpha,f}$ |
| 5 | Construire le vecteur g_α à l'aide de la formule (4.4.3) |
| 6 | Obtenir f_α à partir de g_α , avec (4.8) |
-

4.4.1 Construction du graphe

Soit $\alpha \in I_{\text{disp}}^h$ et f une configuration initiale donnée (associée à la carte de disparité u^h). Commençons par détailler la construction du graphe $\mathcal{G}^{\alpha,f}$ représentant l'énergie $E^{\alpha,f}$. Pour des précisions plus techniques quant à cette construction, nous invitons le lecteur à se reporter à [7].

Sommets Les sommets du graphe $\mathcal{G}^{\alpha,f}$ doivent comprendre une source s et un puits t , ainsi qu'un sommet pour chaque assignement actif sous f de disparité différente de α et un sommet par assignement inactif sous f de disparité α . Ces derniers sont indexés par le premier pixel p , pour lequel on peut construire jusqu'à deux sommets (celui associé à l'assignement $(p,p+u^h)$ et celui associé à l'assignement, s'il est différent, $(p,p+\alpha)$). Ainsi, pour tout pixel de l'image de référence, on construit 0, 1 ou 2 sommets. On confondra dans ce qui suit la notation des assignements et celle des sommets associés.

En pratique, la construction des sommets du graphe se fait à l'aide de deux tableaux, nommés `vars0` et `varsA`, qui sont indexés par les pixels p de l'image de référence. S'ils existent, les coefficients `vars0(p)` et `varsA(p)` représentent respectivement l'assignement actif $(p,p+u^h(p))$ de premier élément p et l'assignement $(p,p+\alpha)$ de disparité α . Les éléments de ces deux tableaux sont soit invariables (`VAR_ACTIVE` ou `VAR_ABSENT`), soit variables (`o` ou `a`).

Les valeurs invariables permettent de localiser les assignements qui ne changeront par d'état (et donc pour lesquels le vecteur g de changement d'état n'est pas défini). La valeur `VAR_ACTIVE` est réservée aux assignements qui restent actifs, c'est-à-dire les assignements $(p,p+\alpha)$ initialement actifs. Ainsi, si l'assignement $a = (p,p+\alpha)$ est initialement actif, on affecte `VAR_ACTIVE` aux coefficients `vars0(p)` et `varsA(p)`. La valeur `VAR_ABSENT` est quant à elle destinée aux sommets qui ne sont pas construits car les assignements associés n'existent pas, ou qui resteront inactifs (car inactifs et de disparité différente de α). C'est le cas lorsque $p+\alpha$ n'appartient pas à l'ensemble des pixels de l'image de droite (car il sort du domaine), auquel cas il existe aucun assignement de la forme $(p,p+\alpha)$; on attribue alors la valeur `VAR_ABSENT` au coefficient `varsA(p)`. C'est également le cas s'il n'existe aucun assignement actif de la forme (p,q) , c'est-à-dire si p est occulté; dans ce cas, on affecte `VAR_ABSENT` au coefficient `vars0(p)`.

Lorsqu'un sommet est construit pour un assignement $a = (p,q)$, alors on crée une variable `o` dans le tableau `vars0` si a est un assignement actif de disparité différente de α , et on crée une variable `a` pour le coefficient `varsA(p)` si l'assignement a est de disparité α . On désignera désormais ces sommets comme des sommets *variables*.

Arcs La construction des arcs suit celle proposée dans l'étude du paragraphe 4.2.2, en particulier les graphes représentés à la figure 4.1.

Le terme d'attache aux données ne comporte que des fonctions dépendant d'une seule variable binaire. On doit donc construire pour chaque sommet variable a un arc, relié à la source ou au puits, suivant le signe du coût $D(a)$. Si $D(a)$ est positif, alors on construit l'arc (s,a) , en lui attribuant la capacité $D(a)$. Sinon, on construit l'arc (a,t) , de capacité $-D(a)$.

Le terme de régularité est composé à la fois de fonctions dépendant d'une seule variable et de fonctions dépendant de deux variables. Dans les deux cas, les capacités des arcs construits dépendent des assignements voisins, qui sont localisés en comparant la disparité des assignements des pixels voisins dans les tableaux `vars0` et `varsA`. Soit p le pixel courant et p' un de ses voisins. Commençons par considérer le cas où les assignements voisins $a = (p,q)$ et $a' = (p',q')$ sont de disparité α . Si le sommet $a = (p,p+\alpha)$ associé à p n'est pas variable, alors il n'y a aucun terme variable associé à l'assignement a dans l'énergie $E^{\alpha,f}$. On suppose a est un sommet variable. Si a' n'est pas variable, alors deux cas sont possibles. Si a' n'est pas variable mais actif sous la configuration initiale, alors il faut représenter le terme $\tilde{R}_{\lambda_1,\lambda_2}(a,a',x,1)$ qui ne dépend que de la seule variable a , avec

$$\tilde{R}_{\lambda_1,\lambda_2}(a,a',1,1) = 0 \quad \text{et} \quad \tilde{R}_{\lambda_1,\lambda_2}(a,a',0,1) = R_{\lambda_1,\lambda_2}(p,p',d(a')) > 0$$

avec ici $d(a) = \alpha$. Il faut donc construire l'arc (a', t) avec la capacité $R_{\lambda_1, \lambda_2}(p, p', d(a))$. Si l'arc a déjà été construit à l'étape précédent, il suffit de mettre à jour sa capacité en lui ajoutant cette quantité. Si a' n'est pas variable car l'assignement associé n'existe pas ($p' + \alpha$ n'est pas dans l'image de droite), alors il n'y a rien à construire car le couple (a, a') ne contribue pas au terme de régularisation. Si les deux sommets sont variables, alors il faut représenter le terme $\tilde{R}_{\lambda_1, \lambda_2}(a, a', x, x')$. On utilise pour cela la décomposition (4.5), avec

$$\begin{aligned} \tilde{R}_{\lambda_1, \lambda_2}(a, a', 0, 0) = 0 \quad \text{et} \quad \tilde{R}_{\lambda_1, \lambda_2}(a, a', 0, 1) = R_{\lambda_1, \lambda_2}(p, p', d(a')) > 0 \\ \tilde{R}_{\lambda_1, \lambda_2}(a, a', 1, 1) = 0 \quad \text{et} \quad \tilde{R}_{\lambda_1, \lambda_2}(a, a', 1, 0) = R_{\lambda_1, \lambda_2}(p, p', d(a)) > 0. \end{aligned}$$

On procède de la même manière pour les assignements voisins de disparité différente de α (il faut alors distinguer le cas où $a' = (p', p' + u^h(p))$ est variable et le cas où il n'est pas actif). Enfin, pour représenter le terme d'injectivité, on construit pour les assignements $a = (p, p + u^h(p))$ et $a = (p, p + \alpha)$ (si les deux existent et sont variables) l'arc (a, a') en lui attribuant une capacité infinie (une valeur arbitrairement grande en pratique).

4.4.2 Recherche du flot maximal par chemins augmentants

La recherche du flot maximal est réalisée grâce à l'algorithme proposé dans [3].

Cet algorithme améliore ceux basés sur les chemins augmentants de type FORD-FULKERSON [5]. Les auteurs affirment en effet que la recherche de chemins augmentants classique est trop coûteuse en calculs, car, dans les graphes utilisés en traitement d'images, elle nécessite généralement de parcourir une grande partie des sommets (dont le nombre est de l'ordre de celui des pixels des images utilisées). Ils proposent donc de construire deux arbres, dont les racines sont la source s et le puits t , dans une étape de *croissance* des arbres. Lorsque ces deux arbres se touchent, alors un chemin augmentant est trouvé et le graphe résiduel construit. Ensuite, au lieu de repartir de la source ou du puits pour trouver un nouveau chemin augmentant, on utilise les arbres précédemment construits, en les mettant à jour : on retire les branches saturées par le flot précédent, puis on essaie de reconstituer chaque arbre en reliant les sous-arbres obtenus à l'arbre principal *via* d'autres arcs. Enfin, on poursuit la croissance des nouveaux arbres. L'intérêt majeur de cette approche est que, dans la plupart des cas, la construction des arbres reste suffisamment avancée pour gagner du temps dans la recherche d'un chemin augmentant.

4.4.3 α -Expansion move optimal

Une fois le flot maximal calculé, on en déduit la coupure minimale $(\mathcal{V}^s, \mathcal{V}^t)$ en construisant le graphe résiduel, puis en recherchant l'ensemble des sommets pour lesquels il existe un chemin les reliant à s . On en déduit la valeur des coefficients $\text{vars0}(p)$ et $\text{varsA}(p)$ pour chaque pixel p :

$$\text{Si } \text{vars0}(p) \notin \{\text{VAR_ACTIVE}, \text{VAR_ABSENT}\}, \quad \text{vars0}(p) = \begin{cases} 1 & \text{si } (p, p + u^h(p)) \in \mathcal{V}^t \\ 0 & \text{si } (p, p + u^h(p)) \in \mathcal{V}^s \end{cases}$$

$$\text{Si } \text{varsA}(p) \notin \{\text{VAR_ACTIVE}, \text{VAR_ABSENT}\}, \quad \text{varsA}(p) = \begin{cases} 1 & \text{si } (p, p + \alpha) \in \mathcal{V}^t \\ 0 & \text{si } (p, p + \alpha) \in \mathcal{V}^s. \end{cases}$$

On retrouve alors la valeur du vecteur de changement d'état g_α :

$$g(p, p + u^h(p)) = \text{vars0}(p) \quad \text{et} \quad g(p, p + \alpha) = \text{varsA}(p).$$

Ensuite, la formule d'inversion (4.8) permet d'obtenir la configuration optimale f_α . On peut alors comparer son énergie avec celle de la configuration initiale f .

L'énergie décroît (au sens large) ; si elle n'a pas décréu en une itération complète (c'est-à-dire lorsque chaque disparité $\alpha \in I_{\text{disp}}^h$ a été testée), alors on arrête l'algorithme. Par défaut, le programme limite le nombre maximal d'itérations à 4. Autrement dit, chaque disparité est testée au maximum 4 fois.

4.4.4 Paramètres

La méthode de KOLMOGOROV et ZABIH repose essentiellement sur le choix de trois paramètres : K pour le terme d'occultation et λ_1 et λ_2 pour le terme de régularisation. Une heuristique pour choisir de manière automatique la valeur de ces paramètres est proposée.

Mis à part le terme d'injectivité, les termes de l'énergie E doivent être équilibrés. Sinon, l'un des trois critères risque de l'emporter sur les deux autres. Tout comme dans la méthode présentée au chapitre précédent, le choix des paramètres doit préserver cet équilibre.

Choix du paramètre d'occultation K Ce paramètre correspond au coût d'occultation payé par un pixel occulté. Il participe ainsi au terme d'attache aux données et d'occultation. Sa valeur permet donc principalement de pondérer l'influence relative de ces deux critères. Si K est choisi trop faible, alors il sera globalement plus faible que le coût de corrélation. Par conséquent, il sera globalement moins coûteux de déclarer un pixel occulté que de le mettre en correspondance avec un autre pixel. S'il est au contraire choisi trop grand, alors le phénomène inverse se produit : il est trop coûteux de rendre un pixel occulté, si bien que n'importe quelle mise en correspondance (même erronée, donc engendrant un coût de corrélation important) est préférable.

On voit donc qu'il est essentiel de régler correctement ce paramètre. Dans [8], les auteurs suggèrent de choisir K de sorte qu'en moyenne, pour un pixel, un quart des mises en correspondance seulement soit plus avantageuse que l'occultation.

Choix des paramètres de régularisation λ_1 et λ_2 Une fois que K est choisi, la valeur des deux paramètres λ_1 et λ_2 détermine l'influence relative du terme de régularité par rapport au terme d'attache aux données et d'occultation. Ils ne doivent donc pas être choisis trop grands, au risque d'augmenter de manière significative l'influence de l'attache aux données et de l'occultation, ni trop petits, car sinon les cartes obtenues seront trop régularisées.

KOLMOGOROV et ZABIH proposent dans un premier temps de fixer $\lambda_1 = 3\lambda$ et $\lambda_2 = \lambda$, de sorte que la pénalité pour la régularisation soit trois fois plus importante lorsque le saut de disparité ne coïncide pas avec une discontinuité d'intensité dans l'une des deux vues. Reste alors à choisir la valeur de λ . Il est proposé dans [8] de le choisir égal à $K/5$.

4.5 Adapter l'intervalle de disparité au pixel

Nous proposons dans cette section une modification de l'algorithme de KOLMOGOROV et de ZABIH qui permet d'adapter l'intervalle de disparité à chaque pixel. Cette modification permettra entre autre de proposer un algorithme de densification et une approche de raffinement subpixelique.

4.5.1 Intervalle adaptatif

Principe L'idée est de spécifier pour chaque pixel p de l'image de référence l'ensemble $I_{\text{disp}}^h(p) \subset \mathbb{R}$ des valeurs de disparité qu'il peut prendre, ainsi que de l'autoriser ou non à être déclaré occulté. Cela revient pour tout pixel $p = (i,j) \in \mathcal{I}_L$ à imposer pour la carte de disparité de satisfaire

$$u_{i,j}^h \in I_{\text{disp}}^h(p) \cup \{\text{occ}\} \quad \text{ou} \quad u_{i,j}^h \in I_{\text{disp}}^h(p).$$

On notera désormais $\tilde{I}_{\text{disp}}^h(p)$ l'ensemble des valeurs (incluant éventuellement l'étiquette occ) que peut prendre la disparité $u_{i,j}^h$. On le supposera non vide pour tout pixel p .

On se propose alors de considérer le problème

$$\min_{u_{i,j}^h \in \tilde{I}_{\text{disp}}^h(p)} E(u^h)$$

où la fonctionnelle d'énergie E est celle introduite dans la section 4.1. Au lieu de chercher à la minimiser, on la fait à nouveau décroître par *expansion moves*, en sélectionnant pour chaque $\alpha \in \mathbb{R}$ l' α -*expansion move* de plus faible énergie. Montrons que la recherche de l' α -*expansion move* optimal reste représentable par un graphe, à condition d'utiliser à nouveau le vecteur de changement d'état.

Avant cela, il est nécessaire de modifier la définition des assignements :

$$\mathcal{A} = \left\{ a = (p,q) \in \mathcal{I}_L \times \mathcal{I}_R \mid q - p \in I_{\text{disp}}^h(p) \times \{0\} \right\}.$$

Occultation permise Commençons par étudier le cas où seules les valeurs de disparité sont spécifiées pour chaque pixel (tout pixel restant autorisé à être occulté). Il s'agit donc de considérer le problème

$$\min_{u_{i,j}^h \in I_{\text{disp}}^h(p) \times \{\text{occ}\}} E(u^h).$$

Soit $\alpha \in \mathbb{R}$. Explicitons les α -*expansion moves* $(u^h)'$ d'une carte de disparité u^h . Il est tout d'abord clair que si α n'appartient à aucun des intervalles de disparité $I_{\text{disp}}^h(p)$, alors les seules modifications autorisées consistent à déclarer occultés certains pixels :

$$\forall (i,j) \in \Omega^h, \quad (u^h)'_{i,j} \in \{u_{i,j}^h, \text{occ}\}.$$

Cette opération pouvant être réalisée pour n'importe quelle valeur de α , on va dorénavant ignorer ces α -*expansion moves*. Supposons donc qu'il existe au moins un intervalle $I_{\text{disp}}^h(p)$ contenant α . Les α -*expansion moves* $(u^h)'$ considérées doivent vérifier

$$\forall (i,j) \in \Omega^h, \quad \begin{cases} (u^h)'_{i,j} = \alpha & \text{si } u_{i,j}^h = \alpha \\ (u^h)'_{i,j} \in \{u_{i,j}^h, \text{occ}\} & \text{si } \alpha \notin I_{\text{disp}}^h(p) \text{ et } u_{i,j}^h \neq \alpha \\ (u^h)'_{i,j} \in \{u_{i,j}^h, \alpha, \text{occ}\} & \text{si } \alpha \in I_{\text{disp}}^h(p) \text{ et } u_{i,j}^h \neq \alpha. \end{cases}$$

En termes de configurations et d'assignements, les configurations f' associées sont celles qui vérifient

$$\forall a \in \mathcal{A}, \quad f'(a) = \begin{cases} 1 & \text{si } f(a) = 1 \text{ et } d(a) = \alpha \\ 0 & \text{si } f(a) = 0 \text{ et } d(a) \neq \alpha \end{cases}$$

où f est la configuration associée à la carte de disparité initiale u^h . Ainsi, on voit que la définition des *expansion moves* reste inchangée, seul l'ensemble des assignements \mathcal{A} sur lequel les configurations sont définies varie. Toute l'étude des sections précédentes reste donc valable (puisque aucune condition n'est requise sur l'ensemble \mathcal{A}). La recherche de l' α -*expansion move* de f d'énergie minimale est donc un problème représentable par un graphe, en effectuant le changement de variable (4.7).

Si $\tilde{\mathbb{I}}_{\text{disp}}^h(p) = \mathbb{I}_{\text{disp}}^h \cup \{\text{occ}\}$ pour tout p , alors on retrouve bien la méthode originale.

Occultation interdite Supposons à présent que certains pixels ne sont pas autorisés à être occultés. Les α -*expansion moves* s'écrivent alors

$$\forall (i,j) \in \Omega^h, \quad \begin{cases} (u^h)'_{i,j} = \alpha & \text{si } u^h_{i,j} = \alpha \\ (u^h)'_{i,j} = u^h_{i,j} & \text{si } \alpha \notin \mathbb{I}_{\text{disp}}^h(p) \text{ et } u^h_{i,j} \neq \alpha \text{ et } \text{occ} \notin \tilde{\mathbb{I}}_{\text{disp}}(p) \\ (u^h)'_{i,j} \in \{u^h_{i,j}, \text{occ}\} & \text{si } \alpha \notin \mathbb{I}_{\text{disp}}^h(p) \text{ et } u^h_{i,j} \neq \alpha \text{ et } \text{occ} \in \tilde{\mathbb{I}}_{\text{disp}}(p) \\ (u^h)'_{i,j} \in \{u^h_{i,j}, \alpha\} & \text{si } \alpha \in \mathbb{I}_{\text{disp}}^h(p) \text{ et } u^h_{i,j} \neq \alpha \text{ et } \text{occ} \notin \tilde{\mathbb{I}}_{\text{disp}}(p) \\ (u^h)'_{i,j} \in \{u^h_{i,j}, \alpha, \text{occ}\} & \text{si } \alpha \in \mathbb{I}_{\text{disp}}^h(p) \text{ et } u^h_{i,j} \neq \alpha \text{ et } \text{occ} \in \tilde{\mathbb{I}}_{\text{disp}}(p). \end{cases}$$

Les pixels ne changeant pas de disparité sont donc ceux qui sont de disparité α ou ceux qui ne peuvent ni adopter la disparité α , ni être occulté. Traduisons cette condition en termes de configurations :

$$\forall a = (p,q) \in \mathcal{A}, \quad f'(a) = \begin{cases} 1 & \text{si } f(a) = 1 \text{ et } d(a) = \alpha \\ 0 & \text{si } f(a) = 0 \text{ et } d(a) \neq \alpha \\ 1 & \text{si } f(a) = 1 \text{ et } \alpha \notin \mathbb{I}_{\text{disp}}^h(p) \text{ et } \text{occ} \notin \tilde{\mathbb{I}}_{\text{disp}}(p) \\ 1 & \text{si } f(a) = 1 \text{ et } \alpha \in \mathbb{I}_{\text{disp}}^h(p) \text{ et } \text{occ} \notin \tilde{\mathbb{I}}_{\text{disp}}(p) \\ & d(a) \neq \alpha \text{ et } f(p,p+\alpha) = 0. \end{cases}$$

Les deux dernières conditions empêchent pour un pixel p interdit d'occultation que l'assignement $(p,p+u^h(p))$ (qui est nécessairement actif) ne soit désactivé si l'assignement $(p,p+\alpha)$ n'est activé en échange. Ainsi, si le pixel p ne peut être mis en correspondance avec le pixel $p+\alpha$ (soit que celui-ci n'est pas un pixel de l'image de droite, soit que la valeur α n'est pas une disparité admissible pour le pixel p), alors l'assignement $(p,p+u^h(p))$ doit rester actif. Par ailleurs, si les deux assignements $(p,p+u^h(p))$ et $(p,p+\alpha)$ existent, alors l'un des deux doit être actif.

Cette contrainte revient donc à introduire pour tous les assignements actifs, de la forme $a = (p,p+u^h(p))$, avec p non occultable, le terme suivant dans l'énergie $E^{\alpha,f}$:

$$\chi_{\{(0,0),(0,1),(1,1)\}}(g'(p,p+\alpha), g'(a)).$$

En d'autres termes, on peut remplacer dans le terme d'injectivité le terme correspondant par

$$\chi_{\{(0,0),(1,1)\}}(g'(p,p+\alpha), g'(a))$$

qui impose aux deux assignements (s'ils existent) a et $(p,p+\alpha)$ de changer d'état tous les deux ou de rester inchangés tous les deux ; lorsque l'assignement $(p,p+\alpha)$ n'existe

pas, l'état de l'assignement a reste inchangé. Ce nouveau terme est clairement sous-modulaire, puisque qu'il vaut 0 en (0,0) et (1,1) et prend une valeur infinie pour (0,1) et (1,0). L'énergie résultante est donc représentable.

Modification de l'algorithme KZ2 On voit donc que spécifier l'intervalle de disparité pour chaque pixel suppose de modifier l'ensemble des assignements \mathcal{A} , ce qui affecte la construction des sommets du graphe. Supposons dans un premier temps que le pixel p admette α comme valeur de disparité. Si l'occultation est autorisée, alors on ne change rien par rapport à l'algorithme original. Si l'occultation est interdite, il faut ajouter un arc entre les assignements $(p, p + u^h(p))$ et $(p, p + \alpha)$ pour interdire les changements d'état isolés ; plus précisément, on relie ces deux sommets par deux arcs opposés, tous deux de capacités infinies. Cela force les deux sommets à ne pas être séparés par une coupure.

Si $\alpha \notin I_{\text{disp}}^h(p)$, alors il faut distinguer deux cas. Si l'occultation est interdite, alors l'assignement $(p, p + u^h(p))$ reste actif pendant les α -*expansion moves* ; on ne construit donc pas de sommet pour cet assignement. Dans le code, cela revient à affecter les valeurs

$$\text{vars0}(p) = \text{VAR_ACTIVE} \quad \text{et} \quad \text{varsA}(p) = \text{VAR_ABSENT}.$$

Notons que, dans l'algorithme original, cette combinaison n'apparaît jamais. Si l'occultation est autorisée, alors il faut construire le sommet correspondant, mais aucun sommet pour le couple $(p, p + \alpha)$, puisque celui-ci n'est pas un assignement. On construit donc

$$\text{vars0}(p) = 0 \quad \text{et} \quad \text{varsA}(p) = \text{VAR_ABSENT}$$

ce qui revient à considérer que le pixel $p + \alpha$ n'existe pas dans l'image de droite. Il faut noter que l'introduction d'assignements actifs et non-désactivable, mais de disparité différente de α induit quelques modifications dans le code, car cette situation n'apparaît pas dans le code original. Elle conduit en particulier à ne pas construire de sommet pour l'assignement $(p, p + \alpha)$ lorsque $p + \alpha$ est mis en correspondance avec un pixel p' qui n'est pas amené à changer d'état pendant l' α -*expansion move*, sous peine de violer la contrainte d'unicité.

4.5.2 Densification de cartes de disparité

Motivation Les méthodes locales proposent des cartes de disparité obtenues de manière efficace (voir chapitre 2). Malheureusement, elles n'intègrent aucun modèle global de régularité, et génèrent donc beaucoup d'erreurs, notamment dans les zones d'occultation, les zones non texturées ou bruitées, les textures sujettes à l'effet de STROBES (cf. section 2.1.4). Ces erreurs peuvent être détectées par des tests simples comme le filtre LRRL qui vérifie la cohérence des cartes de disparité de la vue de droite et de la vue de gauche [10], ou bien encore par des procédés plus complexes comme ceux basés sur le principe *a contrario* [11]. Après ce traitement, les pixels mal estimés sont rejetés et la carte de disparité résultante devient éparse (la disparité n'est plus connue partout).

Il peut alors être intéressant d'exploiter un modèle de régularité pour densifier (c'est-à-dire interpoler) ces cartes. En d'autres termes, utiliser une méthode globale (et sa fonctionnelle d'énergie) pour interpoler la carte éparse. Si le nombre de pixels à interpoler est relativement faible, ou que la méthode globale utilisée est efficace,

cette étape de densification nécessitera un temps de calcul raisonnable. On se propose d'adapter l'algorithme KZ2 à cette fin.

Choix des intervalles Utiliser la méthode de KOLMOGOROV et ZABIH pour densifier une carte éparsée revient à rechercher une carte de disparité d'énergie minimale E , où seuls les pixels rejetés par le(s) filtre(s) d'erreur sont laissés libres de prendre n'importe quelle valeur de disparité (ou éventuellement d'être occultés). Soit u_0^h une carte de disparité éparsée que l'on cherche à densifier. Les pixels dont la disparité n'est pas connue sont étiquetés par **unknown**. Ici, nous supposons que les pixels non rejetés (dont la disparité est connue) ne sont pas occultés, et nous cherchons à densifier la carte tout en détectant les occultations. Aussi, la densification peut se traduire en termes d'intervalles de disparité adaptatifs. Pour les pixels p de disparité connue $u_0^h(p)$ dans la carte de disparité initiale éparsée, on interdit tout changement, ce qui revient à poser

$$\forall p = (i,j) \in \mathcal{I}_L, \quad (u_0^h)_{i,j} \neq \text{unknown} \quad \Longrightarrow \quad \tilde{I}_{\text{disp}}^h(p) = \{(u_0^h)_{i,j}\}.$$

Le pixel ne peut donc pas être déclaré occulté non plus. Pour les autres pixels, dont la disparité reste à estimer, on pose

$$\forall p = (i,j) \in \mathcal{I}_L, \quad (u_0^h)_{i,j} = \text{unknown} \quad \Longrightarrow \quad \tilde{I}_{\text{disp}}^h(p) = I_{\text{disp}}^h \cup \{\text{occ}\}.$$

Ainsi, on ne contraint pas leur valeur et ils peuvent être occultés.

Algorithme La procédure pour une densification à l'aide de la méthode proposée peut donc être décrite par l'algorithme 3.

Algorithme 3: Densification par *expansion moves*

Entrée : u_0^h une carte de disparité initiale éparsée, I_{disp}^h l'intervalle de disparité
Sortie : u^h une carte de disparité dense (avec occultations)

```

1 begin
2   foreach pixel  $p = (i,j) \in \mathcal{I}_L$  do
3     if  $(u_0^h)_{i,j} \neq \text{unknown}$  then  $\tilde{I}_{\text{disp}}^h(p) = \{(u_0^h)_{i,j}\}$ 
4     else  $\tilde{I}_{\text{disp}}^h(p) = I_{\text{disp}}^h \cup \{\text{occ}\}$ 
5    $u^h \leftarrow u_0^h$ 
6   while l'énergie  $E$  décroît do
7     foreach valeur de disparité  $\alpha \in I_{\text{disp}}^h$  do
8        $u^h \leftarrow \underset{\substack{(u^h)' \text{ } \alpha\text{-expansion move de } u^h \\ (u^h)'_{i,j} \in \tilde{I}_{\text{disp}}^h(p)}}{\text{argmin}} E((u^h)')$ 

```

Utilisation de points fiables Cette méthode permet également d'incorporer des informations que l'on possède *a priori* sur la carte de disparité. Il peut s'agir de points dont une autre méthode (de mise en correspondance par corrélation ou de mesure directe) a permis d'établir de manière fiable la disparité.

Estimation des occultations à partir d'une carte dense Il est également possible d'utiliser cet algorithme pour estimer les occultations dans une estimation dense de la carte de disparité. Pour cela, on affecte à chaque pixel p de disparité $d(p)$ le singleton $\{d(p)\}$ comme intervalle de disparité (avec éventuellement une quantification préalable si l'estimation initiale est subpixelique) et on autorise le pixel à devenir occulté.

4.5.3 Raffinement subpixelique

Motivation La précision des cartes de disparité peut être un sujet important. Malheureusement, augmenter la précision peut être très coûteux. Pour une précision au demi-pixel par exemple, la taille de l'intervalle de disparité est doublée. La mise en correspondance passe ensuite généralement par un sur-échantillonnage (de facteur 2 dans le cas de la précision au demi-pixel) horizontal d'au moins une des images. Le volume de coût double de volume. Pour les méthodes locales, on a pour chaque pixel de l'image de référence deux fois plus de corrélations à calculer. Pour une méthode globale comme celle proposée au chapitre précédent, on doit manipuler 4 (ou 5) volumes de taille deux fois plus grands que pour la précision pixelique (ce qui implique en autres deux fois plus de projections à calculer). La convergence, qui est liée à la taille du problème, est également plus lente. Pour l'algorithme KZ2, il y a deux fois plus de disparités α à tester dans les *expansion moves*. Calculer une carte de disparité à précision subpixelique peut donc très vite devenir techniquement limité.

Une stratégie pour limiter la complexité du problème est de *raffiner* une carte de disparité pixelique. L'idée est de partir d'une carte de précision pixelique, puis de relâcher pour chaque pixel l'intervalle de recherche autour de la disparité pixelique précédemment estimée, en incluant cette fois des valeurs subpixeliques. Dans le cas idéal où la carte pixelique serait correcte, un raffinement subpixelique d'un demi pixel consisterait pour tout pixel non occulté $p = (i, j)$ de disparité pixelique $(u_0^h)_{i,j}$ à tester les disparités $\{(u_0^h)_{i,j}, (u_0^h)_{i,j} + 0,5, (u_0^h)_{i,j} - 0,5\}$ et à retenir la carte d'énergie minimale sous cette contrainte. Ainsi, pour chaque pixel non occulté, seul un nombre faible (généralement $2N + 1$) de disparité α est testé.

Choix des intervalles Supposons que l'on cherche à estimer la disparité avec une précision de $h_t = 1/N$ pixel, où N est un entier non nul (avec $h_t = 1/2$ dans le cas de la précision au demi-pixel, $h_t = 1/4$ pour le quart de pixel par exemple). Si on a une carte de disparité pixelique u_0^h obtenue par une méthode quelconque, alors on définit pour chaque pixel $p = (i, j)$ non occulté

$$(u_0^h)_{i,j} \neq \text{occ} \quad \Longrightarrow \quad \tilde{I}_{\text{disp}}^h(p) = \{(u_0^h)_{i,j} + h_t \llbracket -N + 1 ; N - 1 \rrbracket\} \cup \{\text{occ}\}.$$

Il est possible d'interdire l'occultation (si la carte des occultations est fiable par exemple), ou au contraire, si la carte initiale n'est pas suffisamment fiable, de relâcher légèrement l'intervalle de disparité en remplaçant $N - 1$ par un entier plus grand. Pour les pixels occultés (s'il y en a), on peut soit interdire toute modification (si la carte des occultations est vraiment fiable), ou plus simplement de ne pas contraindre la disparité

$$(u_0^h)_{i,j} = \text{occ} \quad \Longrightarrow \quad \tilde{I}_{\text{disp}}^h(p) = \{I_{\text{disp}}^h + h_t \llbracket -N + 1 ; N - 1 \rrbracket\} \cup \{\text{occ}\}.$$

Il y a donc théoriquement N fois plus d' α -*expansion moves* à tester dans une itération, mais chacun d'entre eux n'affecte qu'une petite partie des assignements (ceux dont le

premier élément $p = (i,j)$ a pour disparité pixellique $(u_0^h)_{i,j} \in \{[\alpha], \lceil \alpha \rceil\}$ et ceux qui sont initialement occultés).

En pratique, on utilisera le raffinement pour doubler la précision. Si la disparité d^N a été estimée à une précision $1/N$, alors chaque pixel p non occulté se voit affecté l'intervalle $d^N(p) + \{-1,0,1\}/(2N)$. Ainsi, pour obtenir une estimation d'une précision au quart de pixel par exemple, il faut commencer par raffiner au demi-pixel la carte pixellique, puis raffiner au quart de pixel la carte raffinée obtenue.

Algorithme Le raffinement subpixellique d'une carte de disparité de précision pixellique est décrit par l'algorithme 4.

Algorithme 4: Raffinement sous-pixellique par *expansion moves*

Entrée : u_0^h une carte de disparité initiale de précision pixellique,
 I_{disp}^h l'intervalle de disparité pixellique, $h_t = 1/K$ la précision souhaitée

Sortie : u^h une carte de disparité sous-pixellique de précision h_t

```

1 begin
2   foreach pixel  $p = (i,j) \in \mathcal{I}_L$  do
3     if  $(u_0^h)_{i,j} \neq \text{occ}$  then  $\tilde{I}_{\text{disp}}^h(p) = \{(u_0^h)_{i,j} + h_t \llbracket -K + 1; K - 1 \rrbracket\} \cup \{\text{occ}\}$ 
4     else  $\tilde{I}_{\text{disp}}^h(p) = \{I_{\text{disp}}^h + h_t \llbracket -K + 1; K - 1 \rrbracket\} \cup \{\text{occ}\}$ 
5    $u^h \leftarrow u_0^h$ 
6   while l'énergie  $E$  décroît do
7     foreach valeur de disparité  $\alpha \in I_{\text{disp}}^h + h_t \llbracket -K + 1; K - 1 \rrbracket$  do
8        $u^h \leftarrow \underset{\substack{(u^h)' \text{ } \alpha\text{-expansion move de } u^h \\ (u^h)'_{i,j} \in \tilde{I}_{\text{disp}}^h(p)}}{\text{argmin}} E((u^h)')$ 

```

4.6 Résultats expérimentaux

On présente dans cette section les résultats obtenus avec la méthode étudiée dans ce chapitre. Le code utilisé est soit celui proposé dans la publication IPOL [7] pour l'algorithme original, soit une version modifiée de celui-ci pour l'intervalle de disparité adaptatif (modifié suivant les remarques de la section 4.5).

4.6.1 Algorithme original

Commençons par présenter les résultats obtenus avec l'algorithme original sur les paires du banc d'essai Middlebury.

Paramètres automatiques La figure 4.3 présente les cartes de disparité (colonne de gauche) obtenues en choisissant les paramètres suivant l'heuristique proposée dans le paragraphe 4.4.4. Leurs valeurs sont indiquées pour chaque paire, ainsi que les taux d'erreur pixellique (supérieure à un pixel), Middlebury (strictement supérieure au pixel) et subpixellique (supérieure au demi-pixel). Le masque affiché en cyan (colonne du milieu) correspond à l'erreur pixellique. Dans tous les cas, seuls les pixels non occultés

(d'après la vérité-terrain) sont pris en compte. Enfin, on présente (colonne de droite) également la carte des détections d'occultations avec les taux de précision et de rappel (voir Chapitre 3).

Nombre d'itérations, temps de calcul Le temps de calcul est donné dans le tableau 4.4, ainsi que le nombre d'itérations nécessaires. Par défaut, le nombre d'itérations est limité à 4. La figure 4.5 permet de comparer les cartes obtenues pour moins de 4 itérations et celles obtenues pour moins de 8 itérations. À titre d'information, on donne pour chaque paire la taille du volume de coût (voir chapitre 3), qui est un volume de taille $N_x \times N_y \times N_t$, où $N_x \times N_y$ correspond aux dimensions de l'image de référence et N_t le nombre de disparité dans l'intervalle de disparité I_{disp}^h .

Choix des paramètres On rappelle que la valeur automatique des paramètres suit les lois suivantes :

$$\lambda = \frac{K}{5}, \quad \lambda_1 = 3\lambda \quad \text{et} \quad \lambda_2 = \lambda$$

où K est le coût d'occultation, λ_1 le coût de régularité lorsque les variations de couleur sont faibles et λ_2 le coût de régularité lorsqu'elles sont importantes. Il est donc possible de modifier manuellement la valeur de ces paramètres de différentes manières.

On commence par modifier la valeur de K , en conservant toutes les dépendances listées plus haut. Ainsi, plus K est grand, plus l'occultation est coûteuse, mais c'est également le cas des discontinuités de disparité. Au contraire, si K est plus faible, alors l'occultation tout comme les sauts de disparité sont moins coûteux. On choisit de modifier la valeur de K de deux façons différentes : soit en doublant (ou en divisant par 2) la valeur déterminée automatiquement par l'heuristique de KOLMOGOROV et ZABIH, soit en modifiant directement cette heuristique, en demandant à ce que K soit choisi de sorte qu'en moyenne un tiers (resp. un cinquième) des corrélations soient plus coûteuses que l'occultation (au lieu d'un quart dans l'algorithme original), ce qui a pour effet d'augmenter (resp. diminuer) la valeur de K . Les résultats obtenus par cette procédure sont présentés à la figure 4.6.

Il est ensuite possible de modifier le rapport entre la valeur de K et celle de λ (en conservant les définitions de λ_1 et de λ_2). Par défaut, ce rapport est de 5, on choisit donc de le doubler ou de le diminuer de moitié. Dans cette expérience, la valeur de K reste celle déterminée automatiquement par l'algorithme. Cette modification a pour effet de changer le poids relatif du terme d'attache aux données + occultation et du terme de régularité. La figure 4.7 montre les résultats obtenus suite à cette modification.

On peut également modifier le rapport entre λ_1 et λ_2 . Plus précisément, on conserve la valeur de K , celle de λ et de λ_1 , et on change le facteur dans la définition de λ_1 , en le diminuant ou en l'augmentant. Cela revient à pénaliser davantage les discontinuités de disparité qui ne coïncident pas avec une discontinuité de couleur. Les cartes obtenues sont présentées dans la figure 4.8.

On peut enfin modifier le seuil qui détermine si le coût de régularité est λ_1 ou λ_2 . Par défaut, il vaut 8. Ce seuil correspond à la sensibilité du détecteur de bords dans l'image : plus il est important, plus il y a de bords détectés. Les résultats obtenus à la suite de ces modifications sont présentés à la figure 4.9.

Pour terminer, on notera que le calcul du paramètre K dépend de l'estimation de l'intervalle de disparité. La figure 4.10 présente les cartes obtenues pour deux (sur-)estimations différentes de cet intervalle. Les paramètres sont alors calculés automatiquement (en suivant l'heuristique décrite plus haut).

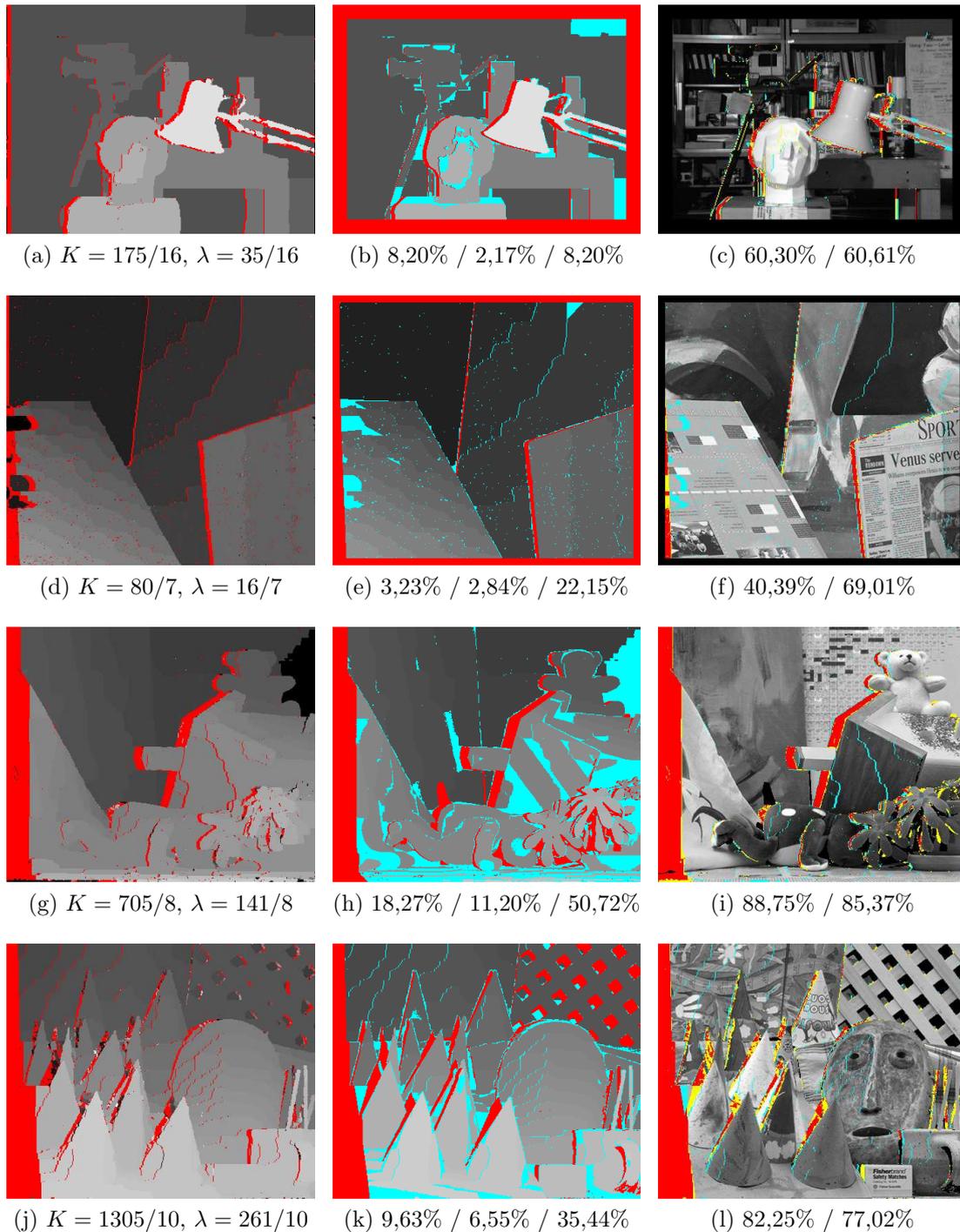


FIGURE 4.3 – Résultats obtenus à partir de l’algorithme original KZ2 (précision **pixellique**).
 Colonne de gauche : carte de disparité, avec en rouge les occultations détectées. En légende :
 les paramètres estimés automatiquement. Colonne du milieu : erreur d’estimation. En rouge,
 le masque des points dont la disparité n’est pas connue (d’après la vérité-terrain fournie
 par Middlebury). En cyan, les disparités mal estimées pour l’erreur pixellique (supérieure ou
 égale à 1). En légende : le pourcentage d’erreur pixellique / erreur Middlebury (strictement
 supérieure à 1) / erreur sous-pixellique (supérieure ou égale à 0,5) dans les zones occultées
 (d’après la vérité-terrain). Les pixels déclarés occultés par KZ2 alors qu’ils ne le sont pas sont
 comptabilisés comme des erreurs dans tous les cas. Colonne de droite : erreur dans la détection
 des occultations (en rouge les détections correctes, en jaune les faux négatifs et en cyan les
 faux positifs). En légende : le taux de précision et de rappel. De haut en bas : Tsukuba,
 Venus, Teddy et Cones.

Paire	Tsukuba	Venus	Teddy	Cones
Taille du volume de coût (en mégapixels)	1,8	3,3	10,1	10,1
Temps d'exécution (en secondes)	5	12	32	28
Nombre d'itérations	3,8	3,8	4,0	4,0

FIGURE 4.4 – Temps d'exécution et nombre d'itérations pour la méthode proposée. La taille du volume de coût est donnée par $N_x \times N_y \times N_t$, où $N_x \times N_y$ est la taille de l'image de référence et N_t la taille de l'intervalle de disparité.

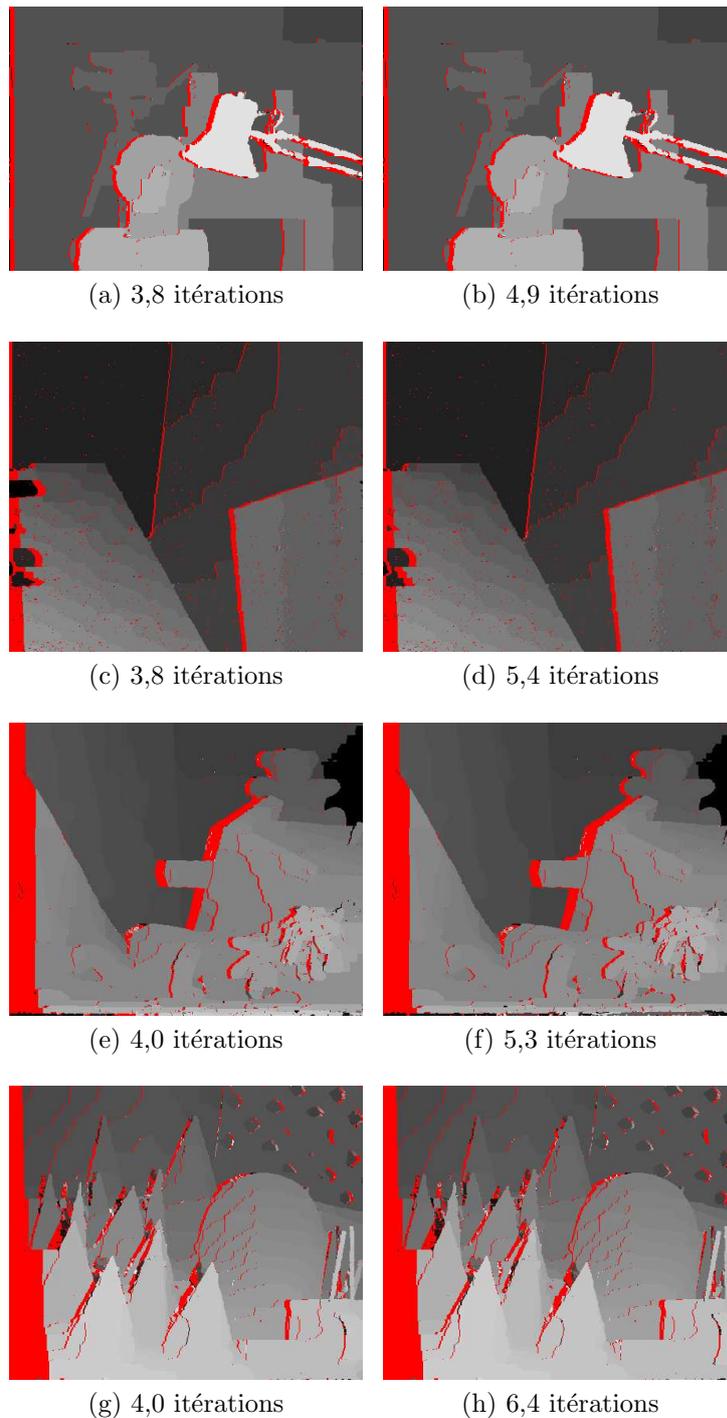


FIGURE 4.5 – Résultats obtenus à partir de l'algorithme KZ2, pour deux nombres d'itérations maximaux différents (4 pour la colonne de gauche, 8 pour celle de droite). En légende : le nombre d'itérations effectif. De haut en bas : Tsukuba, Venus, Teddy, Cones.

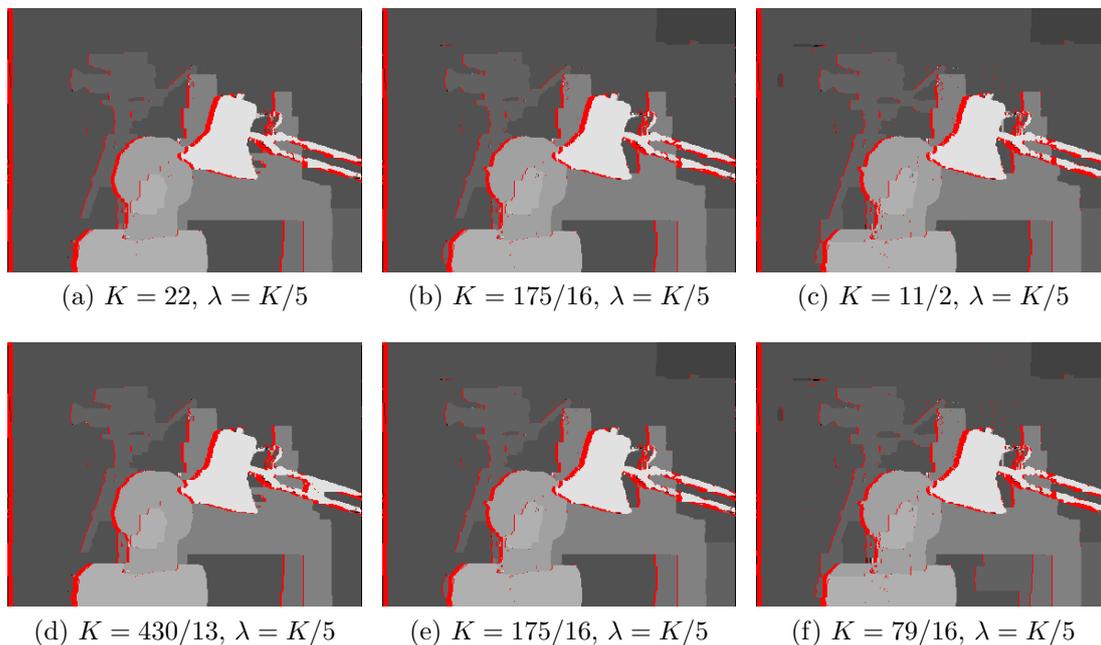


FIGURE 4.6 – Choix de K . On modifie la valeur de K de deux manières : en doublant/réduisant de moitié la valeur calculée automatiquement par l’algorithme (ligne du haut, colonne de gauche et colonne de droite respectivement) ; en choisissant K de sorte qu’en moyenne, un tiers (resp. un cinquième) des corrélations soit plus avantageux que l’occultation (colonne de gauche et de droite, respectivement). Au milieu : résultat obtenu avec K par défaut.

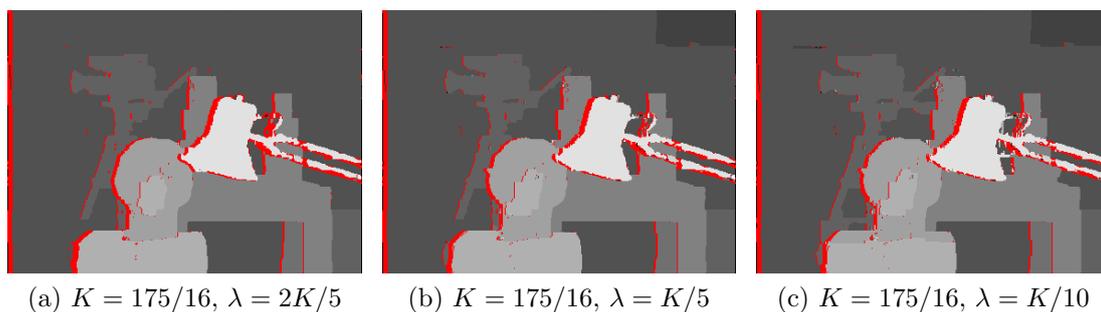


FIGURE 4.7 – Rapport entre le coût d’occultation K et le coût de régularité λ . On modifie le facteur de proportionnalité entre ces deux paramètres (en laissant les autres paramètres dépendant de la nouvelle valeur de λ) : en doublant la valeur de λ , ce qui revient à considérer $\lambda = 2K/5$ (colonne de gauche), soit en la réduisant de moitié, en choisissant $\lambda = K/10$ (colonne de droite). Le paramètre K reste inchangé pour les trois expériences. La colonne du milieu correspond au résultat obtenu avec les paramètres par défaut.

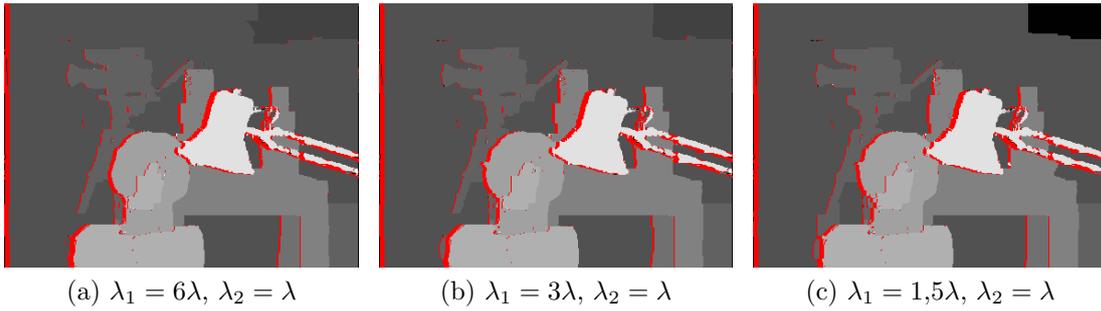


FIGURE 4.8 – Rapport entre le coût de régularité λ_1 (sans discontinuité de couleur) et le coût de régularité λ_2 (avec discontinuité de couleur). On modifie le facteur de proportionnalité entre ces deux paramètres : en doublant la valeur de λ_1 , ce qui revient à considérer $\lambda_1 = 6\lambda$ (colonne de gauche), soit en la réduisant de moitié, en choisissant $\lambda = 1,5\lambda$ (colonne de droite). Les paramètres K et λ_2 restent inchangés pour les trois expériences. La colonne du milieu correspond au résultat obtenu avec les paramètres par défaut.

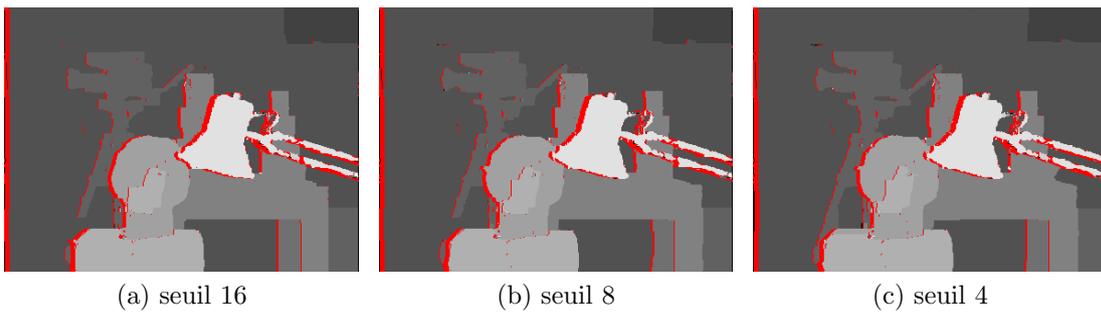


FIGURE 4.9 – Sensibilité du détecteur de bords. On modifie le seuil dans la définition du coût de régularité, qui décide si la pénalité est de λ_1 ou de λ_2 . Il vaut par défaut 8. Les paramètres K , λ_1 et λ_2 restent inchangés pour les trois expériences. La colonne du milieu correspond au résultat obtenu avec les paramètres par défaut.

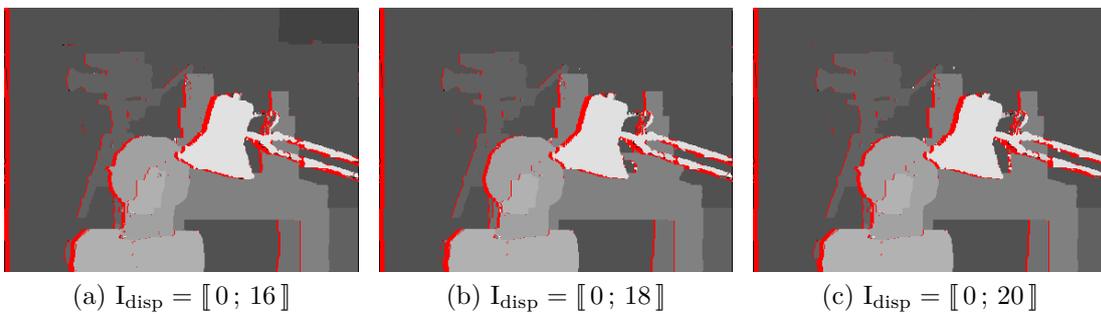


FIGURE 4.10 – Influence de l'estimation de l'intervalle de disparité. La colonne de gauche correspond au résultat obtenu avec les paramètres par défaut (estimation optimale de l'intervalle de disparité).

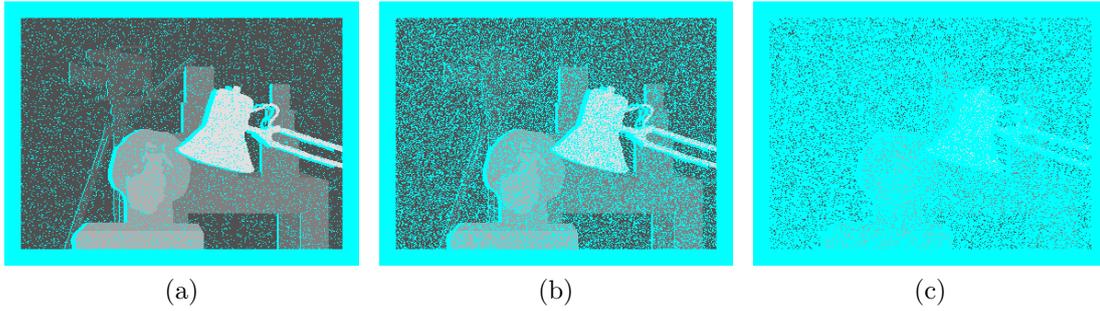


FIGURE 4.11 – Exemples de cartes éparses. À gauche, on a injecté 70% de la vérité-terrain comme information connue. Au milieu, ce taux est de 50%. À droite, il est de 10%.

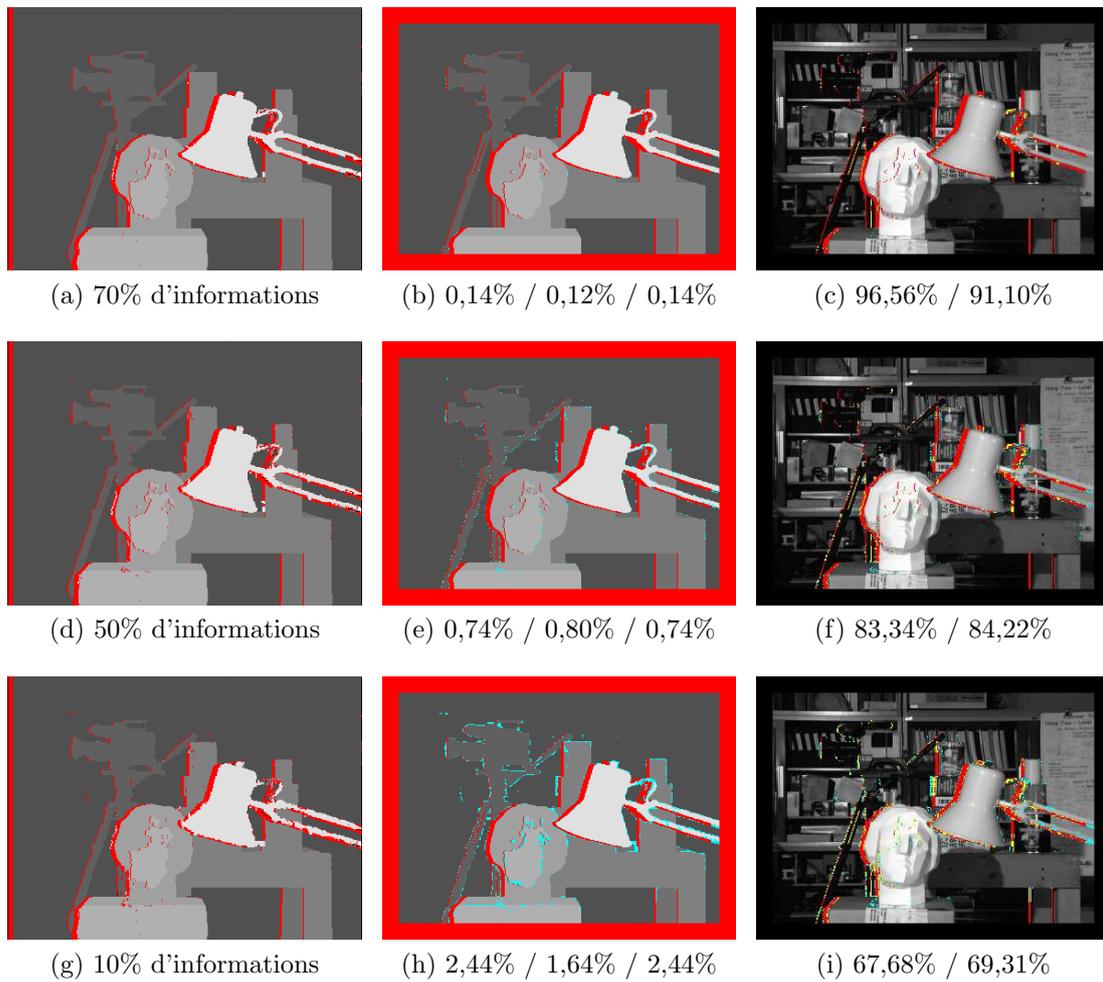


FIGURE 4.12 – **Densification** de la vérité-terrain. Colonne de gauche : carte de disparité, avec en rouge les occultations détectées. Colonne du milieu : erreur d'estimation. En rouge, le masque des points dont la disparité n'est pas connue (d'après la vérité-terrain fournie par Middlebury). En cyan, les disparités mal estimées pour l'erreur pixellique (supérieure ou égale à 1). En légende : le pourcentage d'erreur pixellique / erreur Middlebury (strictement supérieure à 1) / erreur sous-pixellique (supérieure ou égale à 0,5) dans les zones occultées (d'après la vérité-terrain). Les pixels déclarés occultés par KZ2 alors qu'ils ne le sont pas sont compabilisés comme des erreurs dans tous les cas. Colonne de droite : erreur dans la détection des occultations (en rouge les détections correctes, en jaune les faux négatifs et en cyan les faux positifs). En légende : le taux de précision et de rappel. De haut en bas : résultat obtenu à partir de 70%, 50% et 10% d'informations.

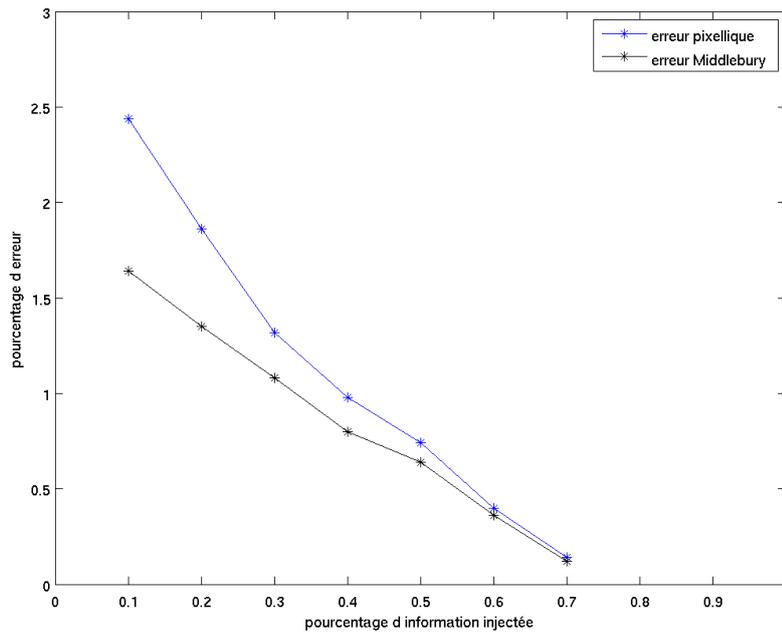


FIGURE 4.13 – Évolution des erreurs en fonction de la quantité d'information connue injectée.

4.6.2 Densification de cartes éparses

Densification sans occultation possible Pour tester cet algorithme, on utilise la vérité-terrain de la paire Tsukuba. On densifie alors les cartes éparses ainsi obtenues comme spécifié dans l'algorithme 3, en attribuant pour chaque pixel p dont la disparité $d(p)$ est connue le singleton $\{d(p)\}$ comme intervalle de disparité, et l'intervalle initial entier pour les autres pixels. On choisit ici de ne pas autoriser les pixels déjà estimés à devenir occultés.

On a choisi (figure 4.12) de laisser les paramètres estimés par défaut (qui correspondent à ceux de la paire Tsukuba). Les taux d'erreur pixellique et Middlebury en fonction de la quantité d'information conservée sont proposés dans la figure 4.13.

Estimation des occultations à partir d'une carte dense Dans la figure 4.14, on montre par ailleurs les résultats obtenus lorsque l'on utilise cet algorithme pour estimer les occultations à partir de la vérité-terrain dense. Pour cela, on affecte à chaque pixel p le singleton $\{d(p)\}$ comme intervalle de disparité et on les autorise à devenir occulté durant le processus. On propose les résultats obtenus avec les paramètres automatiques, puis obtenus en personnalisant les paramètres, où K est deux fois plus importants et λ deux fois plus petit.

4.6.3 Précision subpixellique

La méthode de KOLMOGOROV et ZABIH est théoriquement adaptable à la précision subpixellique, mais en pratique, il faut se ramener à un cadre pixellique. Ainsi, pour obtenir des cartes subpixelliques, on peut sur-échantillonner les images (horizontalement) d'un facteur $k = 2^n$, avec $n \in \mathbb{N}^*$, puis d'appliquer l'algorithme original avec un intervalle de disparité $I_{\text{disp}}^{h_k} = \llbracket k d_{\min} ; k d_{\max} \rrbracket$. Si $n = 1$, alors la précision atteinte est celle du demi-pixel. De manière générale, on obtient une précision de $1/k$ pixel. En procédant

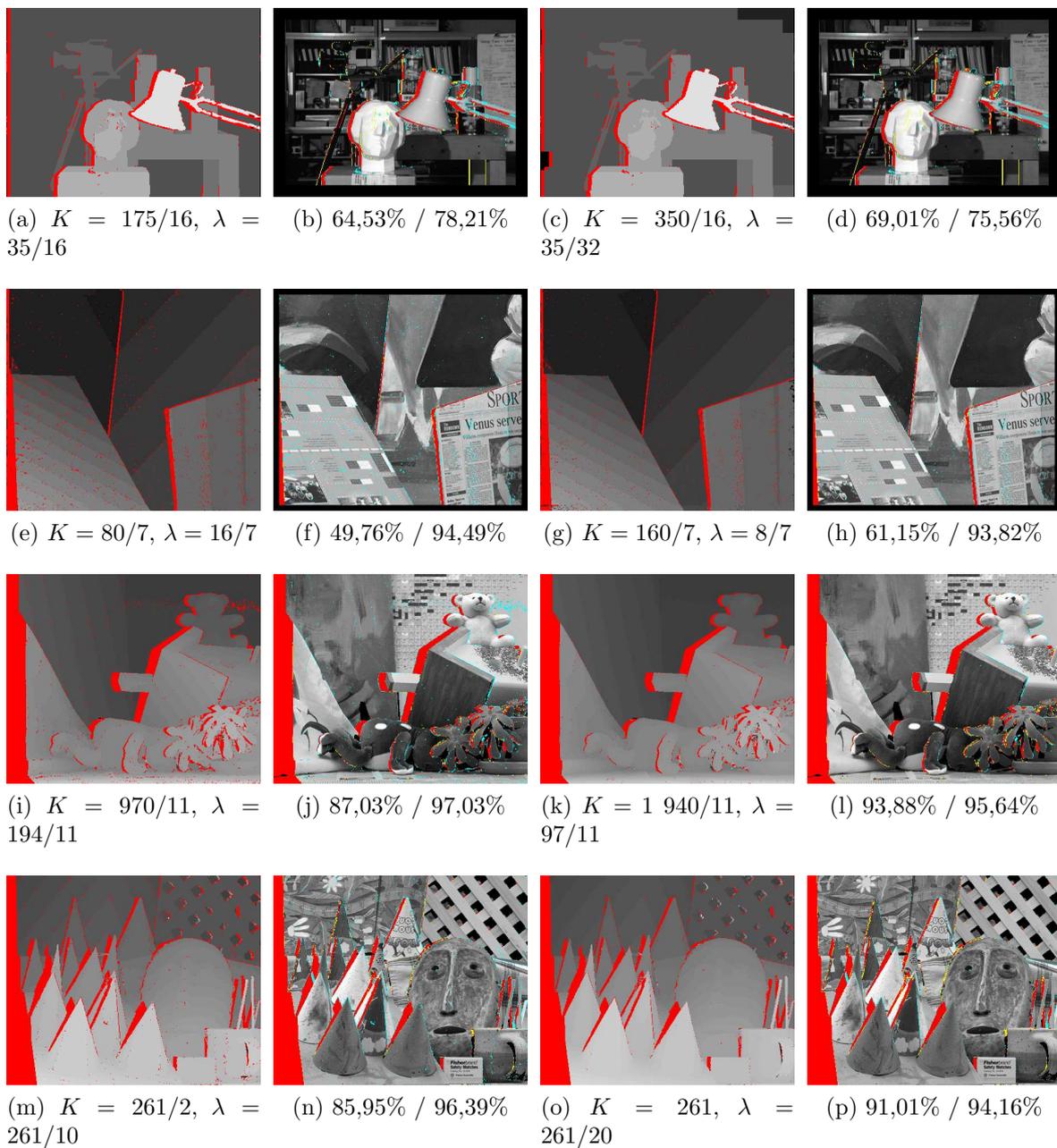


FIGURE 4.14 – Estimation des occultations à partir la vérité-terrain. Les deux premières colonnes : paramètres automatiques. Les deux dernières colonnes : paramètres personnalisés. En légende : colonnes 1 et 3 : paramètres de la méthode ; colonnes 2 et 4 : taux de précision et de rappel. De haut en bas : Tsukuba, Venus, Teddy et Cones.

de la sorte, on obtient une estimation de la disparité de la scène sur-échantillonnée. Pour se ramener à l'échelle initiale, il faut identifier pour chaque pixel de l'image de référence (à l'échelle pixellique) l'ensemble des disparités associées aux pixels issus de celui-ci. En d'autres termes, si $p = (x, y)$ est un pixel de l'image de référence à l'échelle initiale, alors un sur-échantillonnage d'un facteur de $k = 2^n$ le décompose en k pixels, de coordonnées (à l'échelle subpixellique) $(x, k y + \llbracket 0; k - 1 \rrbracket)$. L'ensemble des disparités d^n (à cette échelle) de ces points est alors donné par $\{d^{h_k}(p') \mid p' = (x, k y + \llbracket 0; k - 1 \rrbracket)\}$. Pour les ramener à la scène pixellique, il suffit de les diviser par k . On choisit alors d'attribuer au pixel p la valeur médiane des disparités lorsqu'au moins la moitié des pixels p' n'est pas occultée à l'échelle subpixellique, et de déclarer p occulté sinon.

On teste tout d'abord cet algorithme sur une rampe horizontale artificielle, où la paire est générée à partir d'une image quelconque et une disparité subpixellique affine $u(x, y) = 0.01x + 1.5$ (figure 4.15). On teste ensuite sur les quatre paires de Middlebury (figures 4.16 et 4.17). On notera que les paramètres, toujours calculés automatiquement, sont différents de ceux obtenus pour la précision pixellique, puisque la paire est sur-échantillonnée. Les temps de calculs sont présentés dans le tableau 4.22 pour différentes précisions. Un comparatif des scores est donné dans le tableau 4.21.

4.6.4 Raffinement subpixellique

Pour le raffinement subpixellique, on reprend l'expérience précédente, dans laquelle on spécifie cette fois l'intervalle de disparité en chaque pixel. Si le pixel p est occulté, alors on affecte aux pixels associés $\{p' = (x, k y + \llbracket 0; k - 1 \rrbracket)\}$ l'intervalle de disparité complet $I_{\text{disp}}^{h_k} = \llbracket k d_{\min}; k d_{\max} \rrbracket$. Si p est non occulté, de disparité pixellique $d(p)$, alors on attribue au pixel $p' = (x, k y)$ situé sur la grille pixellique l'intervalle de disparité $I_{\text{disp}}^{h_k}(p') = k d(p) + \llbracket -k + 1; k - 1 \rrbracket$, tandis que tout pixel interpolé $p' = (x, k y + i)$ (avec $i \in \llbracket 0; k - 1 \rrbracket$) se voit attribuer l'union des intervalles de disparité de ces deux plus proches voisins situés sur la grille pixellique. Pour repasser à l'échelle initiale après estimation de la disparité, on procède comme dans le cas précédent. On choisit ici de tester des raffinements successifs permettant de doubler la précision à chaque étape.

On commence par tester cette procédure sur la vérité-terrain des paires Middlebury : après une quantification de ces cartes théoriques (qui donnent des cartes à valeurs entières), on les raffine à l'aide de l'algorithme décrit au paragraphe précédent, en choisissant de doubler la précision. Les résultats obtenus sont visibles à la figure 4.18.

On présente ensuite les résultats obtenus en estimant dans un premier temps la carte pixellique à l'aide de l'algorithme original, puis en raffinant peu à peu, en partant à chaque nouvelle estimation de l'estimation précédente. Les résultats obtenus à la précision du demi-pixel et du quart de pixel sont respectivement présentés dans les figures 4.19 et 4.20, tandis que les temps de calculs cumulés pour différentes précisions peuvent être consultés dans le tableau 4.22. Les scores sont par ailleurs stockés dans le tableau comparatif 4.21.

4.6.5 Discussion

Algorithme original Les performances de l'algorithme original sont très satisfaisantes, tant en termes d'estimation de la carte de disparité qu'en termes de détection de l'occultation. L'erreur pixellique (la plus pertinente pour une méthode fournissant une carte de précision pixellique) varie entre 1,42% pour la paire Venus et 17,18% pour la paire Teddy. Cette dernière paire semble particulièrement souffrir de la quantifica-

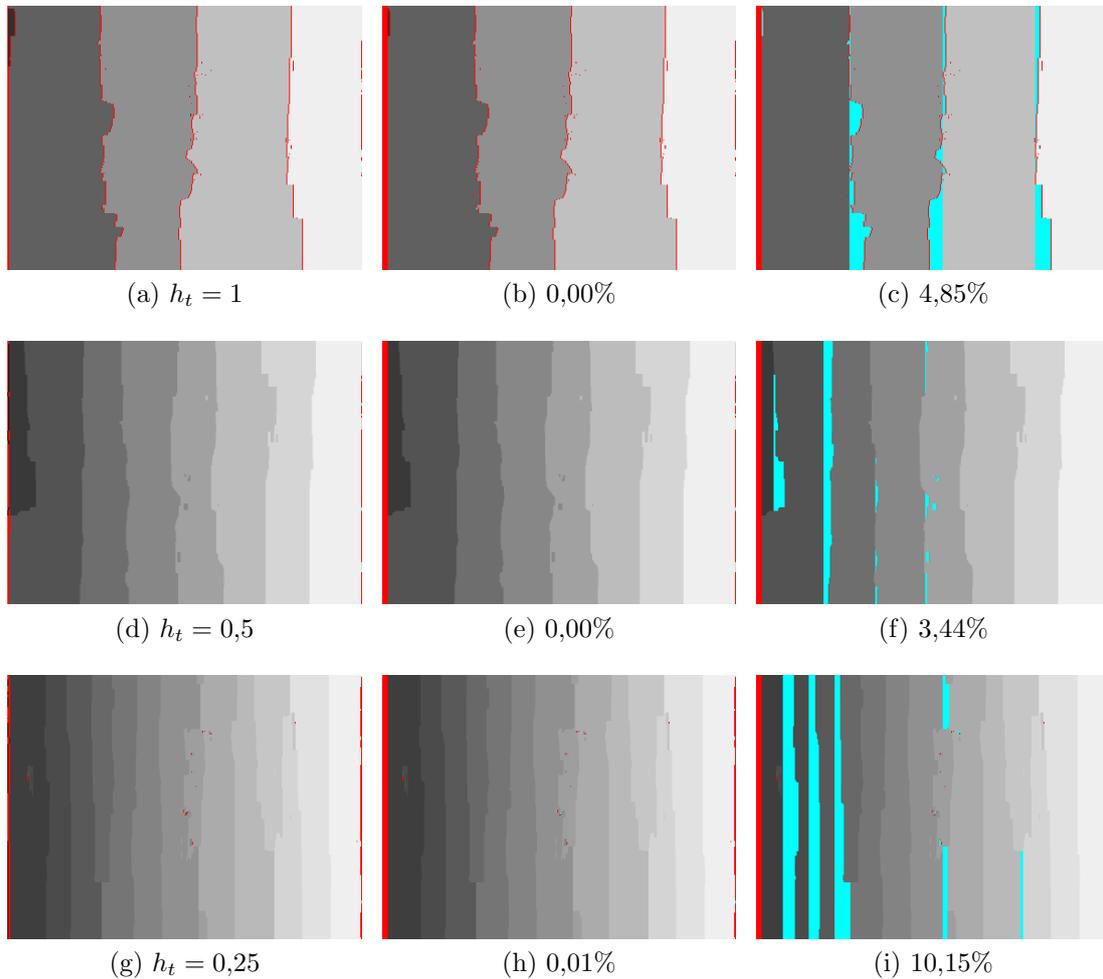


FIGURE 4.15 – Précision sous-pixellique. Exemple de la rampe artificielle. La paire synthétique est construite à partir de l'image de gauche dans la paire Tsukuba et une rampe de disparité u qui permet de générer la vue de droite. À gauche : carte de disparité, avec en rouge les occultations détectées. En légende : la précision de l'estimation h_t . Au milieu : erreur d'estimation pixellique. En rouge, le masque des points dont la disparité n'est pas connue (d'après la vérité-terrain fournie par Middlebury) ou détectés occultés. En cyan, les disparités mal estimées pour l'erreur pixellique (supérieure ou égale à 1). En légende : le pourcentage d'erreur pixellique sur l'ensemble des pixels non occultés (d'après la vérité-terrain ou l'estimation KZ2). À droite : erreur subpixellique (inférieure ou égale à 0.5) avec en légende le pourcentage correspondant.

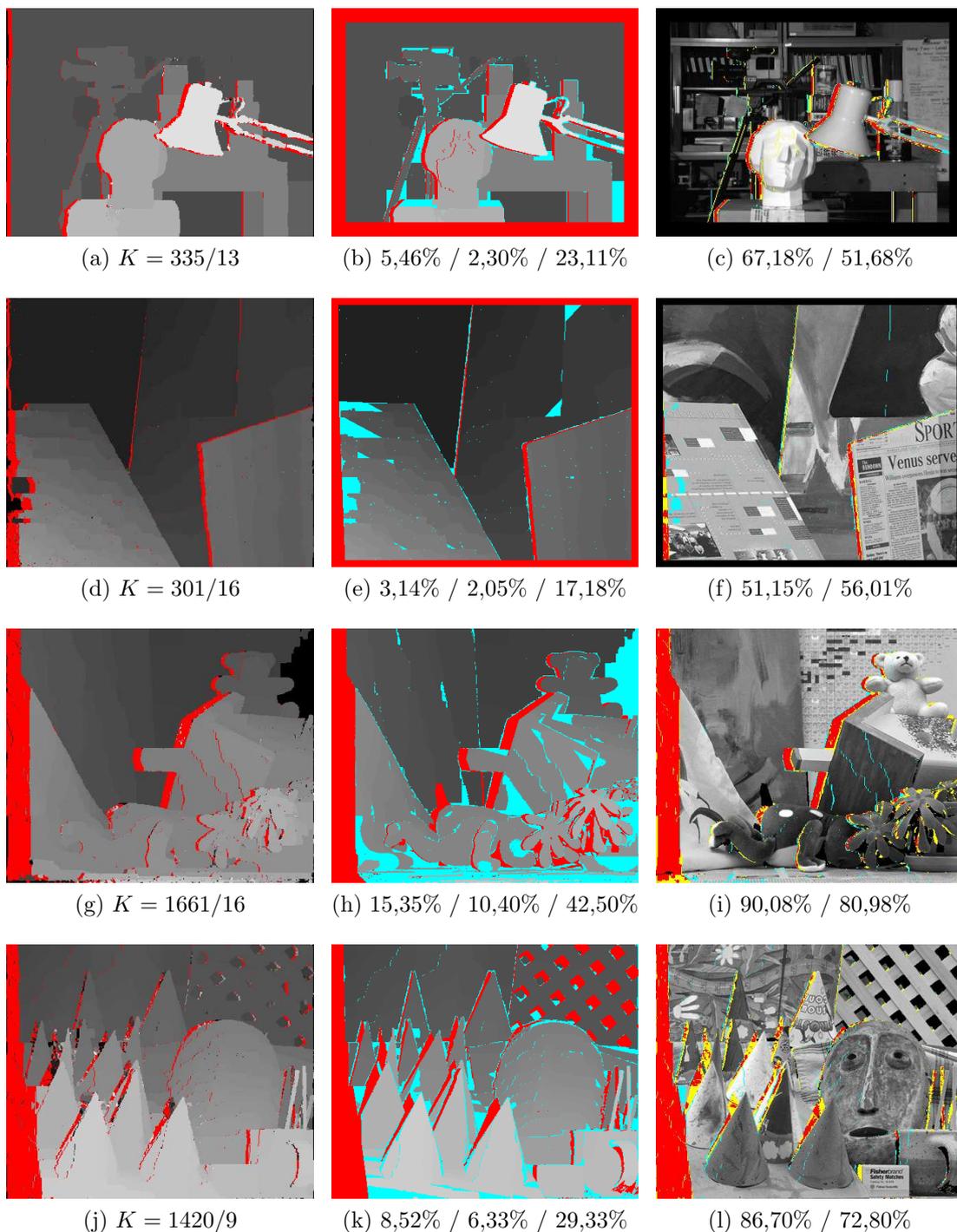


FIGURE 4.16 – **Précision sous-pixellique** ($h_t = 0,5$). Colonne de gauche : carte de disparité, avec en rouge les occultations détectées. En légende : les paramètres estimés automatiquement. Colonne du milieu : erreur d'estimation. En rouge, le masque des points dont la disparité n'est pas connue (d'après la vérité-terrain fournie par Middlebury). En cyan, les disparités mal estimées pour l'erreur pixellique (supérieure ou égale à 1). En légende : le pourcentage d'erreur pixellique / erreur Middlebury (strictement supérieure à 1) / erreur sous-pixellique (supérieure ou égale à 0,5) dans les zones occultées (d'après la vérité-terrain). Les pixels déclarés occultés par KZ2 alors qu'ils ne le sont pas sont compabilisés comme des erreurs dans tous les cas. Colonne de droite : erreur dans la détection des occultations (en rouge les détections correctes, en jaune les faux négatifs et en cyan les faux positifs). En légende : le taux de précision et de rappel. De haut en bas : Tsukuba, Venus, Teddy et Cones.

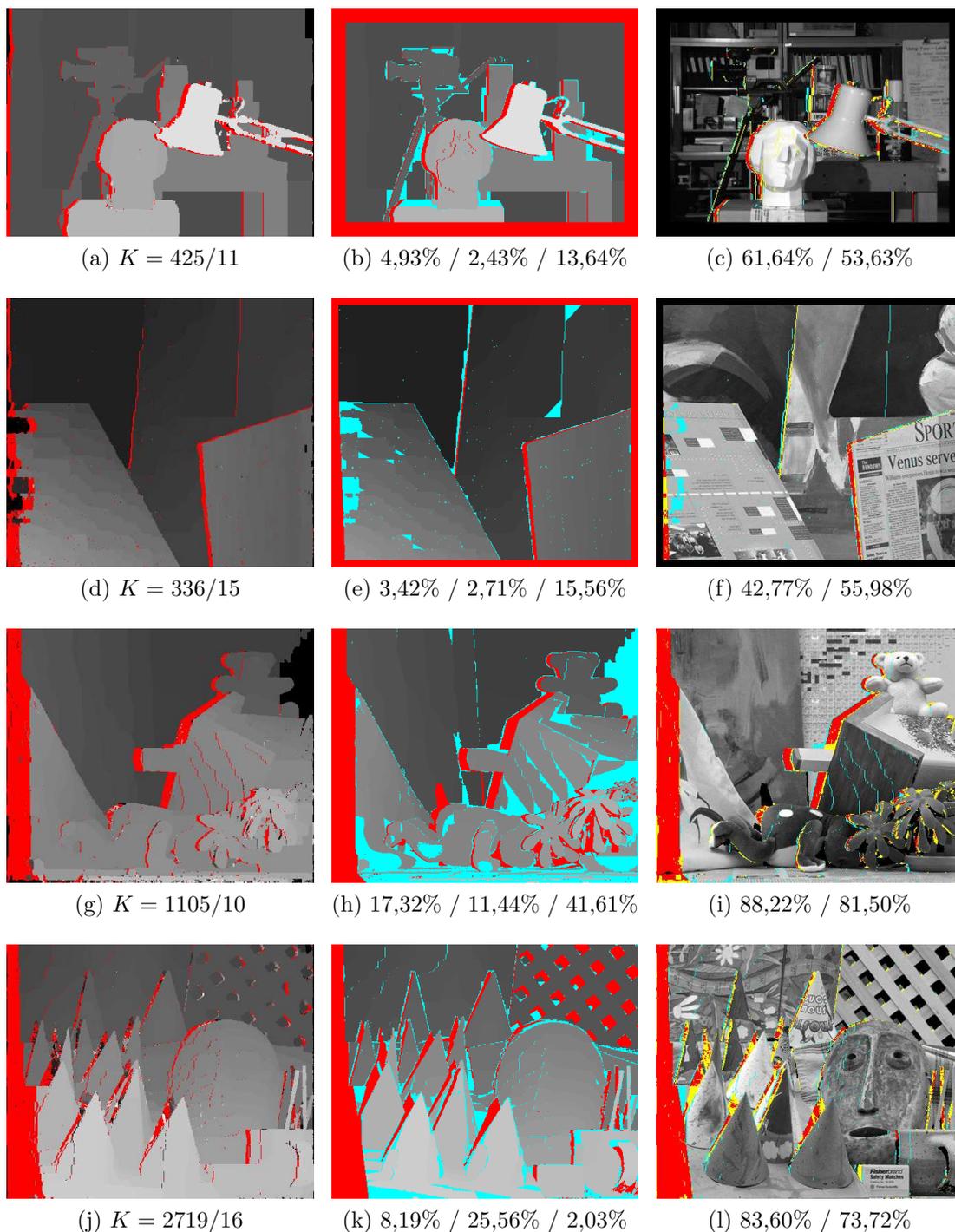


FIGURE 4.17 – **Précision sous-pixellique** ($h_t = 0,25$). Colonne de gauche : carte de disparité, avec en rouge les occultations détectées. En légende : les paramètres estimés automatiquement. Colonne du milieu : erreur d'estimation. En rouge, le masque des points dont la disparité n'est pas connue (d'après la vérité-terrain fournie par Middlebury). En cyan, les disparités mal estimées pour l'erreur pixellique (supérieure ou égale à 1). En légende : le pourcentage d'erreur pixellique / erreur Middlebury (strictement supérieure à 1) / erreur sous-pixellique (supérieure ou égale à 0,5) dans les zones occultées (d'après la vérité-terrain). Les pixels déclarés occultés par KZ2 alors qu'ils ne le sont pas sont compabilisés comme des erreurs dans tous les cas. Colonne de droite : erreur dans la détection des occultations (en rouge les détections correctes, en jaune les faux négatifs et en cyan les faux positifs). En légende : le taux de précision et de rappel. De haut en bas : Tsukuba, Venus, Teddy et Cones.

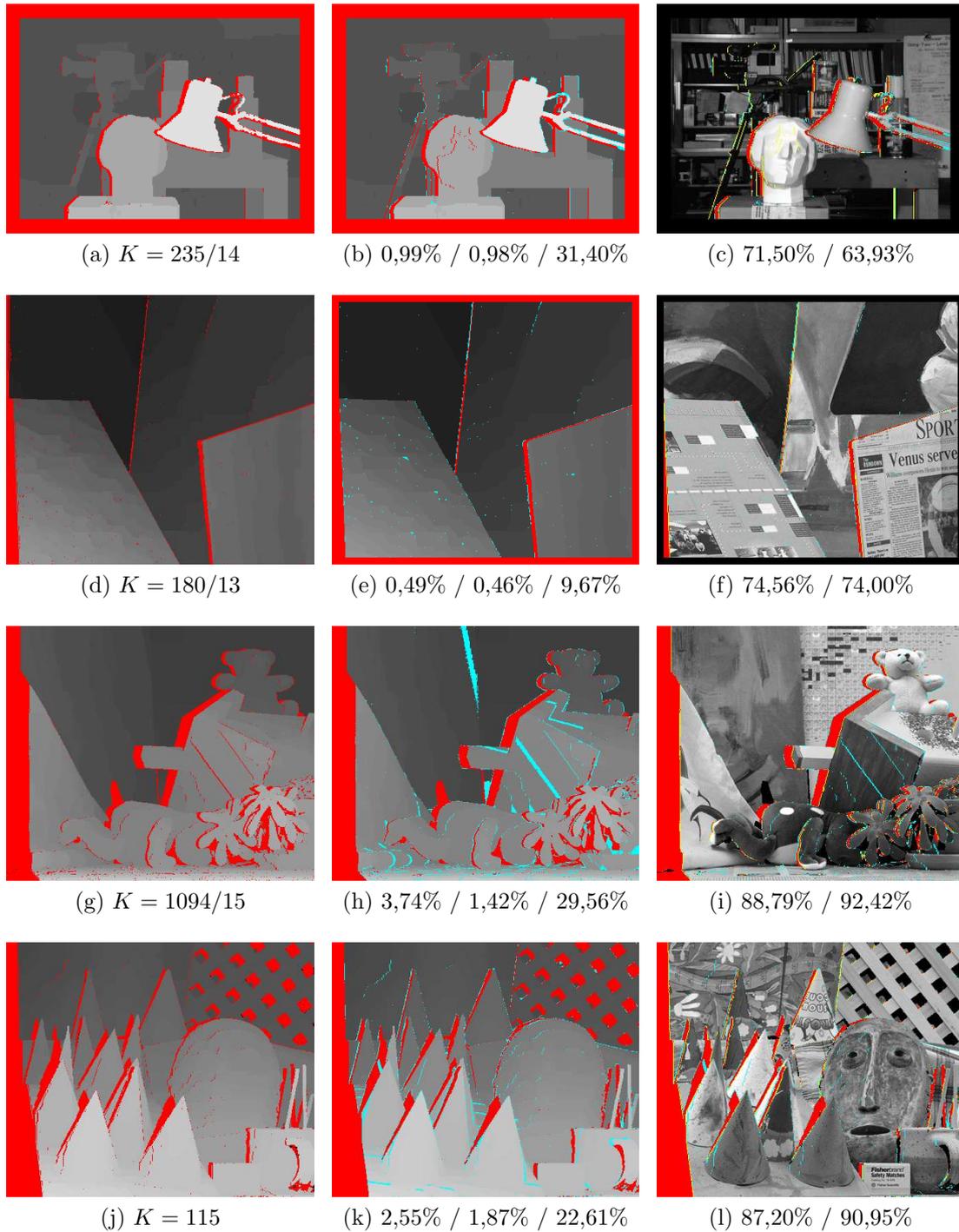


FIGURE 4.18 – Raffinement sous-pixellique ($h_t = 0,5$) à partir de la vérité-terrain quantifiée. Colonne de gauche : carte de disparité, avec en rouge les occultations détectées. Colonne de gauche : carte de disparité, avec en rouge les occultations détectées. En légende : les paramètres estimés automatiquement. Colonne du milieu : erreur d'estimation. En rouge, le masque des points dont la disparité n'est pas connue (d'après la vérité-terrain fournie par Middlebury). En cyan, les disparités mal estimées pour l'erreur pixellique (supérieure ou égale à 1). En légende : le pourcentage d'erreur pixellique / erreur Middlebury (strictement supérieure à 1) / erreur sous-pixellique (supérieure ou égale à 0,5) dans les zones occultées (d'après la vérité-terrain). Les pixels déclarés occultés par KZ2 alors qu'ils ne le sont pas sont compabilisés comme des erreurs dans tous les cas. Colonne de droite : erreur dans la détection des occultations (en rouge les détections correctes, en jaune les faux négatifs et en cyan les faux positifs). En légende : le taux de précision et de rappel. De haut en bas : Tsukuba, Venus, Teddy et Cones.

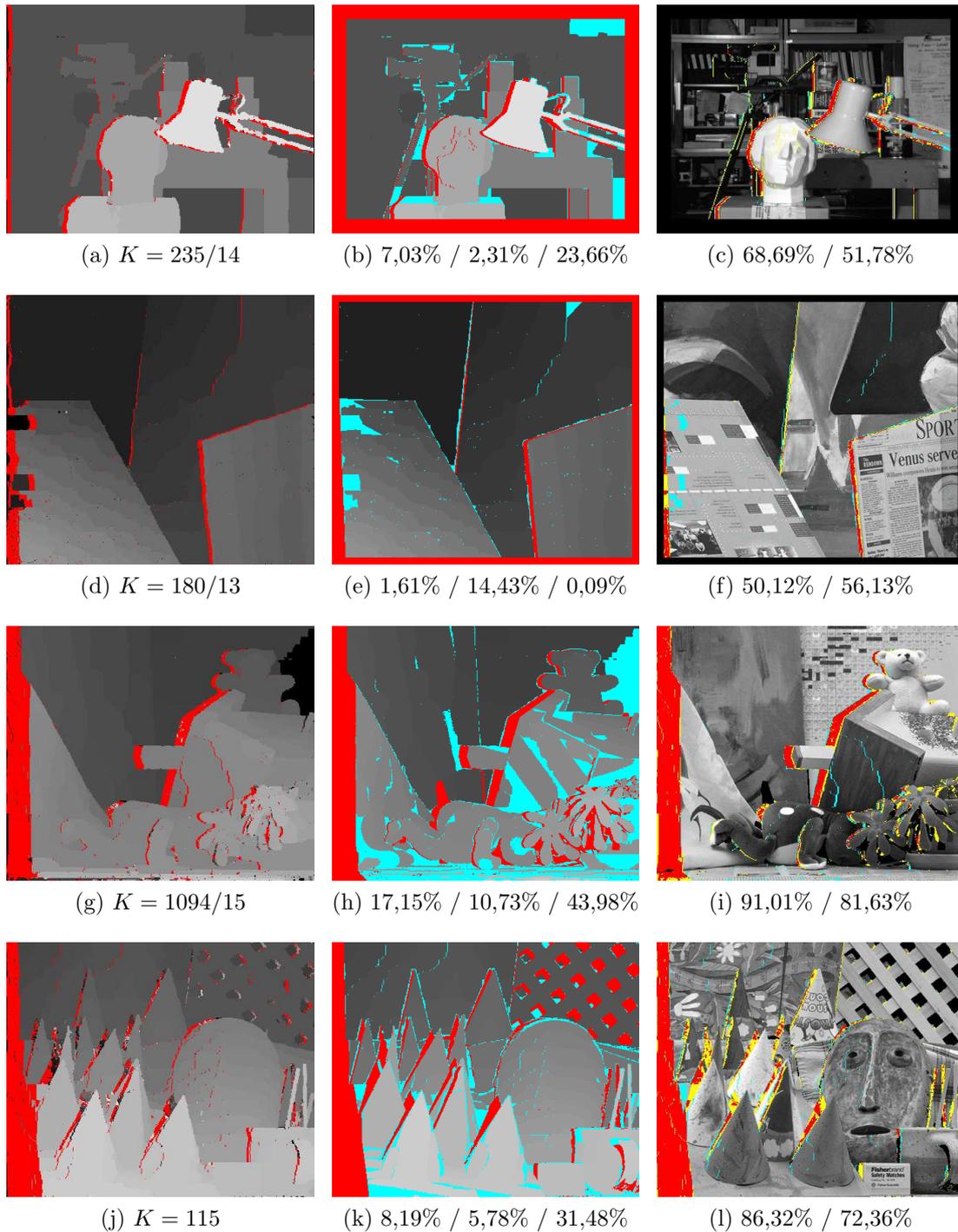


FIGURE 4.19 – **Raffinement sous-pixellique** ($h_t = 0,5$). Colonne de gauche : carte de disparité, avec en rouge les occultations détectées. En légende : les paramètres estimés automatiquement. Colonne du milieu : erreur d'estimation. En rouge, le masque des points dont la disparité n'est pas connue (d'après la vérité-terrain fournie par Middlebury). En cyan, les disparités mal estimées pour l'erreur pixellique (supérieure ou égale à 1). En légende : le pourcentage d'erreur pixellique / erreur Middlebury (strictement supérieure à 1) / erreur sous-pixellique (supérieure ou égale à 0,5) dans les zones occultées (d'après la vérité-terrain). Les pixels déclarés occultés par KZ2 alors qu'ils ne le sont pas sont compabilisés comme des erreurs dans tous les cas. Colonne de droite : erreur dans la détection des occultations (en rouge les détections correctes, en jaune les faux négatifs et en cyan les faux positifs). En légende : le taux de précision et de rappel. De haut en bas : Tsukuba, Venus, Teddy et Cones.

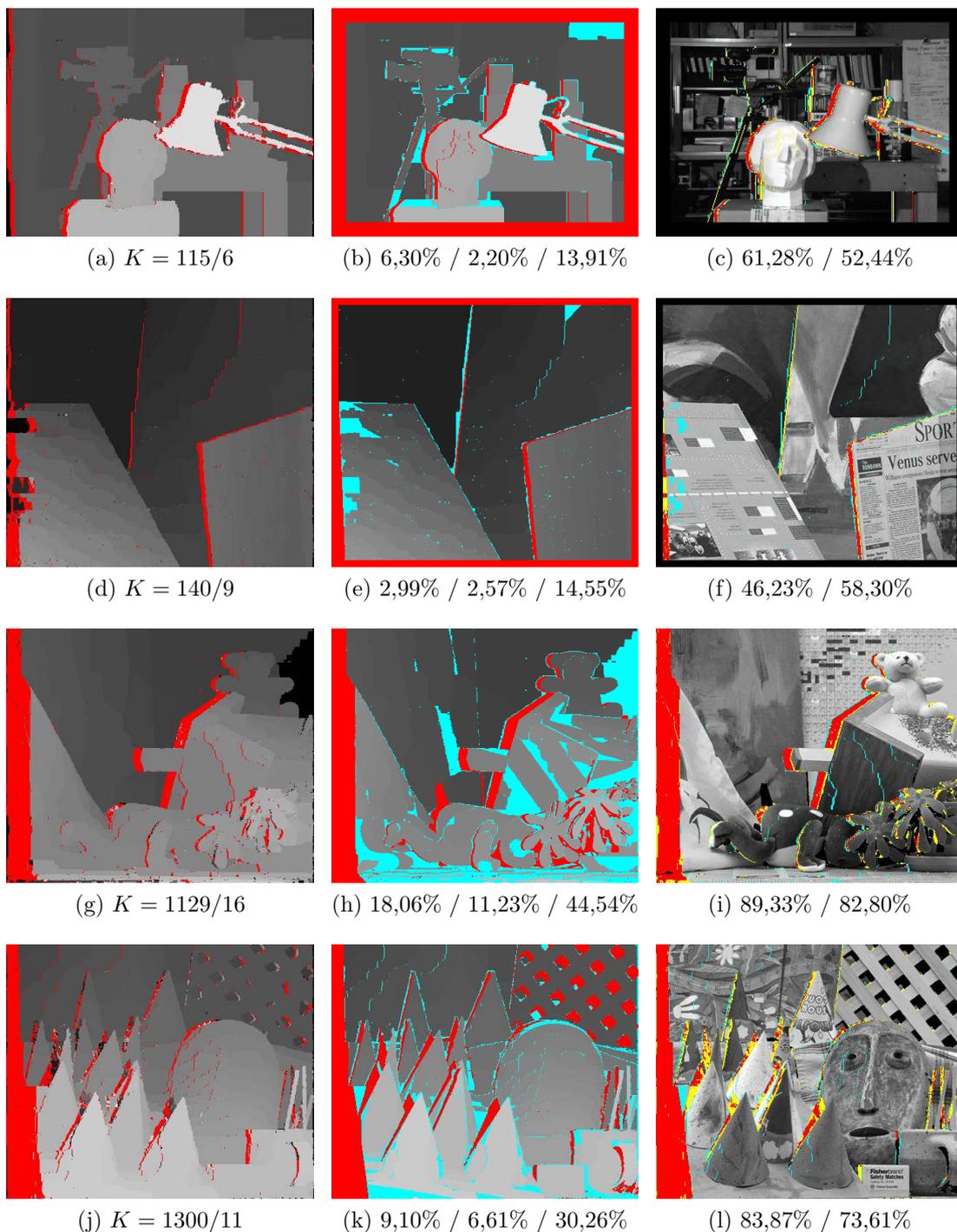


FIGURE 4.20 – **Raffinement sous-pixellique** ($h_t = 0,25$). Colonne de gauche : carte de disparité, avec en rouge les occultations détectées. En légende : les paramètres estimés automatiquement. Colonne du milieu : erreur d'estimation. En rouge, le masque des points dont la disparité n'est pas connue (d'après la vérité-terrain fournie par Middlebury). En cyan, les disparités mal estimées pour l'erreur pixellique (supérieure ou égale à 1). En légende : le pourcentage d'erreur pixellique / erreur Middlebury (strictement supérieure à 1) / erreur sous-pixellique (supérieure ou égale à 0,5) dans les zones occultées (d'après la vérité-terrain). Les pixels déclarés occultés par KZ2 alors qu'ils ne le sont pas sont compabilisés comme des erreurs dans tous les cas. Colonne de droite : erreur dans la détection des occultations (en rouge les détections correctes, en jaune les faux négatifs et en cyan les faux positifs). En légende : le taux de précision et de rappel. De haut en bas : Tsukuba, Venus, Teddy et Cones.

Paire	Tsukuba	Venus	Teddy	Cones
Estimation pixellique ($h_t = 1$)	8,20%	3,23%	18,27%	9,63%
Estimation sous-pixellique ($h_t = 0,5$)	5,46%	3,14%	15,35%	8,52%
Raffinement sous-pixellique ($h_t = 0,5$)	7,03%	2,63%	17,15%	8,19%
Estimation sous-pixellique ($h_t = 0,25$)	4,93%	3,42%	17,32%	8,74%
Raffinement sous-pixellique ($h_t = 0,25$)	6,30%	2,99%	18,06%	7,30%

(a) Erreur pixellique

Paire	Tsukuba	Venus	Teddy	Cones
Estimation pixellique ($h_t = 1$)	2,71%	2,84%	11,20%	6,55%
Estimation sous-pixellique ($h_t = 0,5$)	2,30%	2,05%	10,40%	6,33%
Raffinement sous-pixellique ($h_t = 0,5$)	2,31%	2,23%	10,73%	5,78%
Estimation sous-pixellique ($h_t = 0,25$)	2,43%	2,71%	11,44%	6,36%
Raffinement sous-pixellique ($h_t = 0,25$)	2,20%	2,57%	11,23%	6,61%

(b) Erreur Middlebury

Paire	Tsukuba	Venus	Teddy	Cones
Estimation pixellique ($h_t = 1$)	8,20%	22,15%	50,72%	35,44%
Estimation sous-pixellique ($h_t = 0,5$)	23,11%	17,18%	42,50%	29,33%
Raffinement sous-pixellique ($h_t = 0,5$)	23,66%	15,34%	43,98%	31,48%
Estimation sous-pixellique ($h_t = 0,25$)	13,64%	15,56%	41,61%	27,05%
Raffinement sous-pixellique ($h_t = 0,25$)	13,91%	14,55%	44,54%	31,48%

(c) Erreur sous-pixellique

FIGURE 4.21 – Erreurs d'estimation pour les trois méthodes testées (estimation pixellique, estimations sous-pixelliques et raffinement sous-pixellique). Les raffinements sous-pixelliques s'entendent respectivement comme le raffinement au demi-pixel de la carte pixellique et le raffinement au quart de pixel de ce premier raffinement. Les scores en rouge correspondent pour chaque erreur et chaque paire aux scores les plus faibles.

Paire	Tsukuba	Venus	Teddy	Cones
Taille du volume de coût (en mégapixels)	3,6	6,6	20,2	20,2
Temps d'exécution (en secondes) pour la méthode sous-pixellique	20	59	139	151
Temps d'exécution (en secondes) pour le raffinement sous-pixellique	8	23	74	75

(a) $h_t = 0,5$

Paire	Tsukuba	Venus	Teddy	Cones
Taille du volume de coût (en mégapixels)	7,2	13,2	40,4	40,4
Temps d'exécution (en secondes) pour la méthode sous-pixellique	104	271	711	633
Temps d'exécution (en secondes) pour le raffinement sous-pixellique	42	111	285	292

(b) $h_t = 0,25$

FIGURE 4.22 – Temps d'exécution pour chaque niveau de précision : (a) demi-pixel, (b) quart de pixel. La taille du volume de coût est donnée par $N_x \times N_y \times N_t/h_t$, où $N_x \times N_y$ est la taille de l'image de référence et N_t la taille de l'intervalle de disparité. Pour le raffinement, il s'agit du temps cumulé (on comptabilise le temps de calculs de chaque échelle).

tion de la disparité, en particulier au niveau de la maison. Il s'agit en effet de surfaces dont la disparité varie de manière subpixellique, et la valeur pixellique retenue par l'algorithme ne coïncide pas toujours avec la disparité pixellique la plus proche de la disparité réelle. Notons que la paire Tsukuba est la seule qui possède une vérité-terrain pixellique, ce qui explique que, pour une estimation pixellique de la disparité, l'erreur pixellique et l'erreur subpixellique soient égales.

La détection des occultations est également très bonne, puisqu'on atteint un taux de 82,25% pour la précision et de 85,37% pour le rappel avec Teddy (ces scores seront améliorés par la suite). Il faut toutefois souligner que les occultations peuvent très isolées (un pixel), comme dans la paire Venus, ce qui ne correspond pas à une réalité pertinente (voir l'analyse sur l'occultation réalisée dans le chapitre 2). Enfin, on notera la grande variabilité des valeurs du paramètre K , qui est compris entre 10 et 130,5. Vu la sensibilité de l'algorithme à la valeur de ses paramètres, il semble illusoire d'espérer obtenir des résultats globalement corrects en fixant une valeur unique pour K .

L'algorithme est très rapide, puisqu'il fournit des résultats en quelques secondes. Cela est principalement dû à l'efficacité de l'algorithme de FORD-FULKERSON utilisé pour calculer la coupure minimale. Pour un nombre d'itérations donné, la complexité est de l'ordre du nombre de sommets et d'arcs construits dans le graphe. Or, on construit au plus 5 arcs par sommets, et le nombre de sommets est majoré par le double du nombre de pixels (on ne construit au plus que deux sommets par pixel). Le nombre d' α -*expansion moves* par itération est donné par le nombre de disparité dans l'intervalle de disparité. on en déduit que la complexité est proportionnelle à la taille du volume de coût, ce que l'on vérifie dans le tableau 4.4. On voit par ailleurs dans la figure 4.5 qu'en augmentant le nombre d'itérations maximal autorisé, les résultats varient peu qualitativement. Seule la paire Venus semble en bénéficier (un des trous sur la gauche est comblé). Le choix de ce nombre semble donc justifié.

Lorsqu'on modifie la valeur de K , en conservant les dépendances des autres paramètres (figure 4.6), on constate qu'effectivement, plus K est grand, moins la carte

présente d’occultation et plus elle est régulière. Cela est visible en particulier dans le fond de la scène. En revanche, cette sur-régularisation a pour effet d’introduire des artéfacts sur le bord droit de la lampe, en introduisant une mauvaise estimation des contours de boîtes sur la table. Dans l’exemple (d), pour la valeur de K vaut approximativement 33 (et est donc environ trois fois plus grand que par défaut), tout le fond de la scène est bien estimé (sur le bord droit des pieds de la table en particulier), mais les deux barres dans le bras de la lampe fusionnent. Au contraire, si K prend une valeur plus faible, alors l’occultation tout comme les sauts de disparité sont plus présentes. On remarque en particulier l’apparition d’occultations isolées au dessus de la lampe et à gauche de la caméra. Si on modifie la valeur de λ par rapport à celle de K (figure 4.7), on change le poids relatif du terme d’attache aux données + occultation et du terme de régularité. Si λ est choisi plus faible (égal à $K/10$ par exemple), alors il devient moins coûteux d’introduire des discontinuités ; au contraire, s’il est plus grand ($\lambda = K/2.5$ par exemple), alors la régularité est privilégiée, quitte à introduire davantage d’occultation. Une fois encore, l’effet le plus visible de ce changement de régularité s’observe sur le fond de la scène (coin en haut à droite). L’effet du rapport entre λ_1 et λ_2 est plus subtil (figure 4.8). Lorsque λ_1 est choisi grand ($\lambda_1 = 6\lambda$), alors il devient encore plus coûteux d’introduire une discontinuité de disparité qui ne coïncide pas avec une discontinuité de couleur. L’algorithme a donc tendance à aligner les discontinuités de couleur et de disparité, ce que l’on voit de manière très nette dans le coin en haut à gauche de la scène (à comparer avec l’image de référence). La discontinuité que l’on observe suit exactement un contour présent dans l’image. Enfin, la modification du seuil qui détermine si le coût de régularité est λ_1 ou λ_2 (figure 4.9) ne semble pas avoir d’effet très visible sur les résultats.

On voit ainsi que le paramètre le plus critique est le coût d’occultation K . KOLMOGOROV et ZABIH ont proposé une heuristique qui semble fournir une valeur raisonnable pour ce paramètre, mais il est utile de signaler que ce calcul, qui dépend d’un pourcentage sur les assignements, est en réalité très sensible au choix de l’intervalle de disparité. Si celui-ci est mal estimé, par exemple s’il est beaucoup plus grand que sa véritable valeur, alors la valeur de K augmente (puisqu’on ajoute des corrélations possibles).

Densification Dans la figure 4.13, on observe que, de manière attendue, les pourcentages d’erreurs décroissent à mesure que la quantité d’information (c’est-à-dire de disparité connue) augmente dans les cas considérés. En particulier, introduire 10% d’informations permet de passer d’une erreur pixellique de 8,20% à 2,44%, soit une erreur divisée par trois.

La détection des occultations dans les vérités-terrains donne dans l’ensemble de bons taux de rappel, allant jusqu’à 97,03%. Le taux de précision est moins bon lorsque les paramètres sont choisis automatiquement, car on observe dans ce cas de nombreuses détections isolées (1 pixel). En réglant manuellement ces paramètres, on parvient à les éliminer en partie, mais en perdant quelques détections initialement correctes (comme le montre la baisse du taux de rappel dans la seconde expérience). Ainsi, si la disparité est correctement estimée, la méthode KZ2 fournit un bon détecteur d’occultation, à condition d’en modifier les paramètres.

Précision subpixellique L’estimation pixellique de la disparité réalisée à partir des paires sur-échantillonnées montre des résultats variables. Pour le cas de la rampe (figure 4.15), on observe ainsi que le résultat est visuellement plus satisfaisant à mesure que l’on augmente la précision, car la scène présente une disparité subpixellique. La

quantification introduite par la méthode KZ2 introduit dans ce cas des artéfacts gênants pour l'interprétation visuelle. Cependant, on observe dès cette expérience que l'erreur d'estimation calculée à partir de la vérité-terrain ne reflète pas toujours le gain apporté par l'augmentation de la précision. Ainsi, la carte précise au quart de pixel est visuellement plus proche de la rampe affine, mais présente des erreurs plus importantes (+6,71% pour l'erreur subpixellique dans le pire des cas). Notons que l'erreur pixellique est la même pour chaque précision.

Sur des scènes plus complexes, le gain apporté par la précision subpixellique dépend de la scène et de l'erreur mesurée. À part pour Tsukuba, on observe une amélioration des résultats avec la précision subpixellique (qu'elle soit directe ou obtenue par raffinement). Ce comportement peut être expliqué par le fait qu'il s'agit de la seule carte pixellique des quatre paires considérées, alors que, visiblement, la scène n'est pas constante par morceaux (on pourra considérer le cas de la lampe par exemple). Aussi, il est raisonnable de penser que, dans ce cas précis, la fiabilité de la précision subpixellique de la vérité-terrain est à mettre en cause. L'erreur subpixellique la plus faible est par ailleurs obtenue – pour les trois paires à vérité-terrain subpixellique – est obtenue avec la précision du quart de pixel, avec un gain maximal de 9,11 atteint pour la paire Teddy.

Considérons ensuite le raffinement pixellique. On commence par analyser le raffinement réalisé à partir d'une version quantifiée de la vérité-terrain des paires de Middlebury. Commençons par noter que l'erreur pixellique n'est pas systématiquement nulle, et ce, bien que la carte raffinée ne diffère au plus que d'un demi-pixel de la carte initiale quantifiée. Cela est dû au fait que l'écart de la disparité raffinée avec la vérité-terrain subpixellique peut être égale à 1 tout en conduisant à un écart à la version quantifiée égal à 0,5. En effet, si la disparité correcte (subpixellique) vaut 1,5 par exemple, sa quantification conduit à la valeur 1 (suivant la convention choisie) et une erreur d'un demi-pixel dans l'étape de raffinement peut aboutir à la valeur 0,5. D'autres erreurs pixelliques, observées au niveau des discontinuités de la scène, s'expliquent quant à elle par la convention choisie pour passer d'une carte sur-échantillonnée à une carte à l'échelle initiale et par le choix des intervalles de disparité adaptatifs. Les pixels situés dans de telles régions sont associés dans la scène sur-échantillonnée à des pixels interpolés dont l'intervalle de disparité est plus grand que celui du pixel considéré (car son voisin possédant une disparité pixellique différente, l'union de leurs deux intervalles est nécessairement plus large). Ainsi, l'erreur commise peut excéder le pixel. Néanmoins, on observe que l'erreur Middlebury est nulle partout sauf pour la paire Cones, ce qui s'explique par le fait que, pour cette erreur, seules discontinuités de la scène peuvent être impliquées (l'erreur de quantification étant inférieure ou égale à 1). Le raffinement des vérité-terrains montrent, dans trois cas sur quatre, un gain pour l'erreur subpixellique. À nouveau, la seule paire qui échoue à ce test est la paire Tsukuba. L'erreur double entre l'estimation pixellique (par l'algorithme original) et le raffinement, pourtant réalisé à partir d'une information fiable.

Intéressons-nous enfin à l'expérience de raffinement, réalisée à partir de l'estimation pixellique. Les résultats sont naturellement moins bons que ceux obtenus à partir de la vérité-terrain quantifiée, car le raffinement dépend fortement de l'initialisation. En effet, si une erreur est commise à une échelle, alors elle ne peut être corrigée aux échelles suivantes, à moins que le pixel ne soit déclaré occulté. Néanmoins, pour certaines paires, l'erreur pixellique reste proche (voire plus fiable) que celles obtenues en estimant directement la disparité subpixellique. Pour terminer, comparons en termes d'efficacité l'estimation subpixellique et le raffinement. Le gain apporté par la stratégie

de raffinement est nettement visible dans les tableaux de la figure 4.22. Le temps de calcul est ainsi divisé (en moyenne) par 2,3.

Conclusion

La méthode originale de KOLMOGOROV et ZABIH reste l'une des rares méthodes à produire des détections satisfaisantes de cartes d'occultations, tout en étant d'une complexité algorithmique raisonnable. Cette performance repose sur deux particularités de la méthode. D'une part, la fonctionnelle d'énergie n'est pas minimisée exactement, mais décroît selon une règle de descente (les *expansion moves*), dont chaque pas est choisi de manière optimale. D'autre part, l'optimisation du pas de descente repose entièrement sur la représentabilité par un graphe de l'énergie sous-jacente, qui découle du choix particulier de la fonctionnelle initiale. La résolution se fait donc par coupure de graphes, dont les algorithmes sont connus pour être particulièrement efficaces grâce à la dualité avec le problème de recherche de flot maximal. Malgré les contraintes liées à la représentabilité des énergies considérées, l'algorithme reste suffisamment souple pour permettre des modifications importantes. Nous avons ainsi pu spécifier l'intervalle de disparité de chaque pixel. Cette possibilité permet d'exploiter l'efficacité des coupures de graphes au profit d'autres problèmes, tels que la densification de cartes éparées ou le raffinement subpixelique de cartes pixeliques. Une telle approche peut en particulier améliorer les résultats obtenus avec des méthodes plus fiables (pour produire des cartes éparées ou pixeliques) mais incapables de densifier leurs résultats ou trop coûteux en termes de calcul pour envisager une précision subpixelique.

Néanmoins, la fonctionnelle d'énergie considérée dans cette méthode ne repose sur aucune modélisation explicite du phénomène d'occultation. Celle-ci n'est considérée que comme une alternative moins coûteuse que la mise en correspondance. Or, le chapitre précédent a démontré que l'occultation obéissait à des règles précises. Celles-ci ne sont partiellement que prises en compte (et de manière implicite) dans le terme d'injectivité. Ainsi, bien qu'expérimentalement, la détection d'occultation reste satisfaisante, elle introduit parfois des occultations isolées (de la taille d'un pixel) qui sont tout à fait artificielles et ne correspondent pas à de véritables occultations. Par ailleurs, ce terme repose sur le choix d'un paramètre qui n'est pour l'instant pas rigoureusement justifié et ni compris. Lorsque l'on utilise la méthode comme simple détecteur d'occultations, on observe en particulier que la valeur proposée par défaut de ce paramètre n'est visiblement plus adaptée au problème.

Références

- [1] Stan BIRCHFIELD and Carlo TOMASI. A pixel dissimilarity measure that is insensitive to image sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(4) :401–406, 1998.
- [2] Adrian BONDY and Uppaluri SR MURTY. *Graph Theory (Graduate Texts in Mathematics 244)*. Springer, 2008.
- [3] Yuri BOYKOV and Vladimir KOLMOGOROV. An experimental comparison of Min-Cut/Max-Flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9) :1124–1137, 2004.

-
- [4] Lester R. FORD and Delbert R. FULKERSON. Maximal flow through a network. *Canadian Journal of Mathematics*, 8(3) :399–404, 1956.
- [5] Lester R. FORD and Delbert R. FULKERSON. *Flows in networks*. Princeton University Press, 2015.
- [6] Vladimir KOLMOGOROV. *Graph based algorithms for scene reconstruction from two or more views*. PhD thesis, Cornell University, 2004.
- [7] Vladimir KOLMOGOROV, Pascal MONASSE, and Pauline TAN. KOLMOGOROV and ZABIH’s graph cuts stereo matching algorithm. *Image Processing On Line*, 4 :220–251, 2014.
- [8] 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.
- [9] Vladimir KOLMOGOROV and Ramin ZABIH. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2) :147–159, 2004.
- [10] 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.
- [11] Neus SABATER. *Fiabilité et précision en stéréoscopie : application à l’imagerie aérienne et satellitaire à haute résolution*. PhD thesis, École normale supérieure de Cachan, 2009.