

Optimisation à base de flot de graphe pour la mise en correspondance d'images stéréoscopiques

Sid Ahmed Fezza¹, Nacera Benamrane¹, Habib Zaidi²

¹Groupe Vision et Imagerie Médicale, Laboratoire SIMPA, Département d'Informatique, Faculté des Sciences, Université des Sciences et de la Technologie d'Oran "Mohamed Boudiaf", B.P 1505 El ' Mnaouer 31000, Oran, Algérie

² Division of Nuclear Medicine Geneva University Hospital CH-1211 Geneva 4, Switzerland
sidahmed.fezza@gmail.com, nabenamrane@yahoo.com, habib.zaidi@hcuge.ch

Résumé. Dans ce papier, on propose une approche pour résoudre le problème de la mise en correspondance des images stéréoscopiques. Parmi les algorithmes récents d'appariement proposés dans la littérature, on trouve ceux qui sont basés sur la coupure de graphe, ils transforment le problème de la mise en correspondance stéréoscopique en un problème de minimisation d'une fonction d'énergie globale. L'idée est de montrer comment on peut appliquer la technique de « flot de graphe » pour résoudre ce problème de minimisation. Plusieurs méthodes ont été proposées pour construire ce graphe, mais toutes considèrent en chaque pixel, toutes les hypothèses possibles pour la disparité. Notre contribution consiste à construire un graphe réduit en gardant seulement quelques valeurs potentielles de la disparité pour chaque pixel. Ces valeurs potentielles sont trouvées par une approche locale d'appariement. Ce graphe réduit permet d'accélérer considérablement le temps de calcul de cette méthode globale, en même temps d'améliorer sensiblement la qualité des images de disparité en préservant les discontinuités sur les contours présents dans les images.

Mots-clés: Stéréovision, mise en correspondance, coupure de graphe, minimisation d'énergie, flot de graphe.

1 Introduction

La vision stéréoscopique binoculaire est une méthode importante pour la reconstruction 3D d'une scène. Cette méthode nécessite trois étapes fondamentales pour trouver le relief de la scène: en premier *le calibrage* qui consiste à trouver les paramètres des capteurs, ensuite vient l'étape de *la mise en correspondance* ou *appariement* dont le but est de trouver les points homologues entre les deux images et produit comme résultat une carte de disparité qui représente l'ensemble des correspondances entre les deux images. Chaque pixel d'une carte de disparité représente l'amplitude de la disparité, c'est-à-dire, la distance entre la position du pixel de l'image gauche et celle de son correspondant dans l'image droite. La dernière étape

est la *reconstruction 3D* qui à partir des paramètres des capteurs et des correspondances de points reconstruit un modèle 3D de la scène.

De manière générale, la mise en correspondance stéréoscopique consiste à retrouver dans les images gauche et droite, les primitives homologues, c'est-à-dire, les primitives qui sont la projection de la même entité de la scène. Ces primitives peuvent être les *pixels* de l'image ou des points d'intérêt (comme les points de Harris), dans ce cas on parle de *stéréo dense (pixel-based stereo)*. Elles peuvent être aussi des primitives *structurées* tels que des segments, des contours..., dans ce cas on parle de *stéréo éparsé (feature-based matching)*. Dans le cadre de notre travail, les primitives que nous considérons sont les pixels des images.

La mise en correspondance s'avère être une tâche délicate dont la qualité du résultat détermine directement la qualité de la reconstruction 3D. Elle rencontre de nombreuses difficultés, notamment en présence de changements de luminosité entre les deux images, d'objets dont la surface est uniforme, d'occultations et de bruit dans les images. En raison de ces difficultés, les méthodes de mise en correspondance sont amenées à exploiter toutes les informations disponibles afin de faciliter la recherche et la détermination des correspondants. On a opté pour la méthode de coupure de graphe (*Graph Cuts*) qui a été introduite pour la première fois par S. Roy [2] pour résoudre le problème de la mise en correspondance en stéréovision.

Dans la section 2 nous allons observer les principaux travaux existants sur le problème de mise en correspondance à base de coupure de graphe. En section 3 nous décrivons comment est transformé le problème de la mise en correspondance stéréoscopique en un problème de minimisation d'une fonction d'énergie globale. Nous décrivons dans la section 4 notre approche et les deux méthodes que nous avons évaluées: coupure sur un graphe complet et coupure sur un graphe de taille réduite. Nous finirons par la section 5 où l'on présente notre implémentation et quelques résultats obtenus et par la section 6 qui conclue cet article.

2 Travaux existants

Si certaines méthodes utilisant la programmation dynamique tentent d'imposer une certaine régularité dans la fonction de disparité dans toutes les directions, la plupart n'exploitent pas totalement la cohérence bidimensionnelle. Ce défaut vient du fait que la majorité des méthodes utilisant la programmation dynamique appariées les pixels appartenant à la même ligne épipolaire sur les deux images sans prendre en compte une éventuelle continuité de l'image tridimensionnelle à reconstruire. La théorie des graphes permet de généraliser cette technique en deux dimensions. Roy et Cox [9] sont les premiers à appliqué une méthode globale fondée sur la coupure de graphe pour résoudre la mise en correspondance en stéréovision, ils ont introduit leur technique comme une généralisation des méthodes travaillant sur les lignes épipolaires pour construire des cartes de disparités. S.Roy a proposé une contrainte de cohérence locale qui suggère que la fonction de disparité est localement continue, ce qui veut dire que les pixels proches dans toutes les directions ont des disparités similaires. Roy intègre cette contrainte de cohérence locale avec une contrainte de ressemblance qui dépend de la variation des intensités des pixels appariés.

L'étape suivante de la méthode proposée par Roy et Cox, est de résoudre la fonction de disparité optimale sur toute l'image. Ceci peut être visualisé comme une maille 3D composée de plans, eux-mêmes constitués d'une image de nœuds. Il existe un plan pour chaque niveau de disparité, et chaque nœud représente un appariement entre deux pixels dans les images originales comme il est illustré à la figure 1(a).

Le maillage 3D est composé de points (i, j, d) , où (i, j) sont les coordonnées des pixels d'une des images et d les disparités possibles, d'une *source* s et d'un *puits* t . En interne, chaque point est connecté avec ses quatre voisins dans le plan par des arêtes dites de voisinage (appelé *n-link*), et avec les deux points dans les plans voisins avec des arêtes dites de disparité (appelé *t-link*). Roy et Cox cherchent dans ce graphe une coupe minimum séparant la source du puits. La carte de profondeur est construite à partir de la coupe minimum en associant pour chaque point (i, j) la plus grande disparité de l'arête associée le long de la coupe minimum choisie, comme il est illustré à la figure 1(a et b).

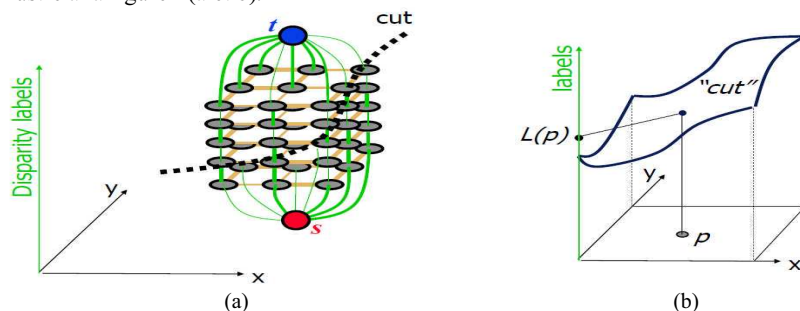


Fig. 1. (a) Le maillage 3D comme proposée par Roy et Cox, (b) la coupure minimale qui représente la surface de profondeur.

La technique de Roy et Cox a été par la suite formalisée par Veksler [8] puis Kolmogorov et Zabih [10, 11, 12] sous la forme d'un problème de labellisation visant à construire une carte de disparité. Chaque label est une valeur de disparité possible. Un label est associé à chaque pixel de manière à minimiser une énergie qui tient compte du voisinage du pixel pour avoir un résultat final régulier. Cette approche a un spectre d'application beaucoup plus large que celle proposée par S.Roy. En revanche, ces méthodes ont deux limitations, la première est qu'elles construisent des cartes de disparité qui ont toujours le défaut d'aplatir les objets ; la seconde est que, comme la fonction de pénalité entre deux voisins de labels différents n'est pas nécessairement convexe, la minimisation de l'énergie est un problème NP-complet dont on obtient finalement qu'une approximation. Par contre, Ishikawa [4] propose l'étude du cas où cette fonction est convexe et propose une méthode générale pour obtenir une fonction de pénalité convexe à une constante près sur la pénalité, ce qui peut être gênant dans le cas d'un échantillonnage irrégulier.

3 Minimisation d'énergie dans la vision

Plusieurs problèmes en vision peuvent être formulés comme un problème d'étiquetage (*Labeling Problem*). Dans un tel problème, on distingue un ensemble de sites et un ensemble d'étiquettes. Les sites représentent les primitives de l'image (*image features*), pour lesquels on veut estimer une quantité: des pixels, des segments...etc. Les étiquettes représentent les quantités associables à ces sites: l'intensité, la disparité, un numéro de région...etc.

Soit $P = 1, 2, \dots, n$ un ensemble de n sites, et soit $L = \{l_1, \dots, l_k\}$ un ensemble de k étiquettes. L'étiquetage est défini par une application de P dans L

$$f: P \rightarrow L \\ s_p \rightarrow f_p = f(s_p) = l_i$$

Donc $f(P) = \{f_1, \dots, f_n\}$

On affecte une fonction d'énergie à cette fonction d'étiquetage, une forme générale des fonctions d'énergies est donnée par:

$$E(f) = E_{data}(f) + \lambda E_{prior}(f) \quad (1)$$

Le premier terme $E_{data}(f)$ représente l'énergie intrinsèque aux données (*data energy*), qui traduit les contraintes de l'association des étiquettes aux données. Le deuxième terme $E_{prior}(f)$ regroupe les énergies extrinsèques (*prior energy*) qui traduisent les contraintes définies par des connaissances a priori. La constante λ peut contrôler l'importance relative des deux termes ; plus λ est grand plus on donne de l'importance aux informations a priori. La fonction d'énergie $E_{data}(f)$ doit être choisie pour affecter un coût important aux associations donnée/étiquette qui sont les moins pertinentes:

$$E_{data}(f) = \sum_{p \in P} D_p(f_p) \quad (2)$$

où $D_p(f_p) \geq 0$ mesure la qualité de l'associations de l'étiquette f_p avec le site p .

La fonction d'énergie a priori $E_{prior}(f)$ doit attribuer un coût important aux associations f_p non compatibles avec l'information a priori. Le choix de cette fonction dépend du type de problème, mais en général, une des fonctions d'énergie a priori exprime des contraintes de lissage (*smoothing*). Cette contrainte est très connue en vision par ordinateur, et elle est bien adaptée quand la qualité à estimer varie lentement partout où presque partout: en 3D, cela correspond à l'hypothèse que le monde est continu par morceaux. Une telle hypothèse est prise en compte en introduisant une énergie d'information a priori de type lissage E_{smooth}

Pour formuler l'énergie de lissage, on a besoin de modéliser comment les pixels interagissent entre eux: souvent il est suffisant d'exprimer comment un pixel interagit avec ses voisins. Notons N_p l'ensemble des pixels voisins du pixel p , et N l'ensemble de paires voisines $\{p, q\}$: N est dit un système de voisinage. L'énergie de lissage peut s'écrire:

$$E_{smooth}(f) = \sum_{\{p,q\} \in N} V_{\{p,q\}}(f_p, f_q) \quad (3)$$

où $V_{\{p,q\}}(f_p, f_q)$ est une fonction d'interaction de voisinage: cette fonction doit attribuer des pénalités aux paires $\{p, q\}$ si les pixels p et q ont des étiquettes différentes.

La forme de $V_{\{p,q\}}(f_p, f_q)$ détermine le type de lissage a priori. Avec ces notations, l'énergie de lissage globale est la somme des fonctions d'interaction de voisinage de tous les pixels voisins. Souvent on choisit d'écrire $V_{\{p,q\}}(f_p, f_q)$ sous la forme:

$$V_{\{p,q\}}(f_p, f_q) = u_{\{p,q\}} V(f_p, f_q) \quad , \quad u_{\{p,q\}} \in \mathbb{R}^+ \quad (4)$$

où V est un potentiel homogène, et $u_{\{p,q\}}$ est un terme multiplicateur dépendant de la clique considérée. Comme potentiel linéaire on a choisi ce potentiel, $V(f_p, f_q) = |f_p - f_q|$. En stéréovision le terme $u_{\{p,q\}}$ est souvent une fonction décroissante de la norme du gradient entre les sites p et q , ce qui permet de favoriser la coïncidence des discontinuités de la disparité avec les contours de l'image de référence. Ce choix est traduit par la fonction $u_{\{p,q\}}$: $u_{\{p,q\}} = U(|I_p - I_q|)$. Le terme $u_{\{p,q\}}$ représente la pénalité d'affecter des disparités différentes pour les pixels voisins p et q ; la valeur de cette pénalité doit être petite pour une paire $\{p, q\}$ qui est sur un contour, donc avec une large différence d'intensité $|I_p - I_q|$. En pratique, on utilise une fonction empirique décroissante.

4 Fonction d'énergie et coupure de graphe

4.1 Coupure sur un graphe complet

En reprenant la formulation générale des problèmes d'étiquetage (eq:1) et en considérant le potentiel linéaire décrit précédemment, on obtient l'énergie globale suivante:

$$E(f) = \sum_{p \in P} D_p(f_p) + \lambda \sum_{\{p,q\} \in N} u_{\{p,q\}} |f_p - f_q| \quad (5)$$

Nous montrons ci-dessous que l'on peut déterminer le minimum global d'une telle fonction en résolvant le problème de la détermination de la coupure minimale dans un graphe. Cette approche est due à S.Roy [2], mais elle a profité de la reformulation proposée par Olga Veksler [8].

Rappelons qu'un graphe est une paire (V, E) , où V est l'ensemble de sites et E est l'ensemble des arêtes entre les sites $E = \{(u, v) | u, v \in V\}$.

On considère le graphe pondéré $G = (V, E)$ où V contient deux nœuds particuliers: une source s et un puits t . Soit k le nombre d'appariements possibles (en stéréovision, k est donné par la plage admissible de disparités, liée à la profondeur minimale et maximale à laquelle se trouvent des objets dans la scène perçue). À chaque pixel p on associe une chaîne de nœuds p_1, p_2, \dots, p_{k-1} . Ces nœuds sont connectés par des arêtes

appelées *t-link* et notées $t_1^p, t_2^p, \dots, t_k^p$ avec $t_1^p = [s, p_l]$, $t_j^p = [p_{f-1}, p_j]$ et $t_k^p = [p_{k-1}, t]$. À chaque t_j^p on affecte une capacité $K_p + D_p(l_j)$ où D_p est le coût d'appariement du pixel considéré pour la valeur de disparité correspondante, généralement ce coût est égal à $(I_p - I_{(p+j)})^2$ et K_p est une constante qui satisfait la contrainte 6. À chaque paire de pixels voisins p et q les chaînes correspondantes sont reliées par des arêtes appelées *n-link*, au niveau $j \in \{1, 2, \dots, k-1\}$: le *n-link* $\{p, q\}$ est de capacité $u_{\{p,q\}}$.

$$K_p > (K-1) \sum_{q \in N_p} u_{\{p,q\}} \quad (6)$$

La capacité d'une coupure *s-t* de ce graphe est la somme des capacités de toutes les arêtes coupées. Vue la méthode de construction du graphe, la capacité de la coupure est constituée de deux parties : la première est la somme des capacités des arêtes *t-link*, et la deuxième est la somme des capacités des arêtes *n-link*. En effet l'addition de la constante K_p sert à assurer l'unicité de la coupure de chaque *t-link*, voir [2] pour une preuve de cette unicité.

Une autre méthode pour assurer l'unicité de la coupure de chaque *t-link* est proposée par Ishikawa et Geiger [4]. Leur graphe est très proche mais il est orienté et ils ajoutent à la chaîne des sites associés au pixel p , une chaîne inverse mais à capacité infinie.

Une coupure de graphe consiste à diviser le graphe en deux parties. Les *t-link* coupés forment la surface de profondeur recherchée (figure 1(b)), ce qui permet d'associer une disparité à chaque pixel. Le problème de coupure de graphe peut se résoudre par le flot maximum. Ford et Fulkerson [6] ont montré que le flot de la source s vers le puits t fait saturer un ensemble d'arêtes divisant les sites en deux parties S et T . Le problème majeur de cette méthode est son temps de calcul énorme, c'est pour cela qu'on a opté pour une nouvelle méthode basée sur un graphe réduit qu'on construit, comme nous le montrons dans la section suivante.

4.2 Coupure sur un graphe réduit

Pour résoudre le problème du temps de calcul qui est énorme, Zureiki et al. [3] propose de construire un graphe réduit: pour chaque pixel on garde dans le graphe seulement quelques disparités potentielles, issues d'une méthode locale de mise en correspondance, alors que la méthode générale proposée notamment par S.Roy, considère toutes les valeurs possibles de disparité. Partant du graphe complet et pour chaque *t-link*, on supprime tous les nœuds sauf les N choisis. La figure 2 détaille la construction du graphe réduit. Les arêtes pointillées sont les arêtes du graphe complet. Les nœuds non supprimés sont en rouge, les nouveaux *t-links* sont en bleu et les nouveaux *n-links* sont en vert. La figure 3(a) illustre une projection frontale du graphe réduit. Par une méthode d'appariement local (stéréo corrélation), on calcule pour chaque pixel, les coûts de mise en correspondance du pixel p avec toutes les valeurs possibles de disparité dans la plage de disparité $[d_{min}, d_{max}]$. Puis on choisit les N meilleures valeurs (pour notre exemple, on va considérer $N = 4$ sans manquer de généralité). Ce choix peut être fait selon différents critères, par exemple, avec un score

ZNCC très classique en stéréo, on peut garder les valeurs de disparité autour du sommet, ou les N (s'ils existent) maximaux locaux. On va noter ces quatre disparités choisies pour le pixel p , comme $d_{1,p}$, $d_{2,p}$, $d_{3,p}$, $d_{4,p}$, et les coûts correspondants par $D_{\{p,d_{1,p}\}}$, $D_{\{p,d_{2,p}\}}$, $D_{\{p,d_{3,p}\}}$, $D_{\{p,d_{4,p}\}}$.

Pour réduire la taille du graphe, nous faisons la simplification suivante: au lieu de garder pour chaque pixel une chaîne ($t-link$) à $k + 1$ arêtes, on supprime toutes les arêtes sauf (à titre illustratif) $N = 4$ arêtes. Ainsi, à chaque pixel p on associe $N - 1$ niveaux (pour l'exemple c'est 3). À chaque $t-link$ on affecte une capacité $C + D_{\{p,d_{i,p}\}}$, où C est une constante qui satisfait la contrainte suivante:

$$C > N * \max_{\{p,q\} \in N} (u_{\{p,q\}}) * |d_{max} - d_{min}| \quad (7)$$

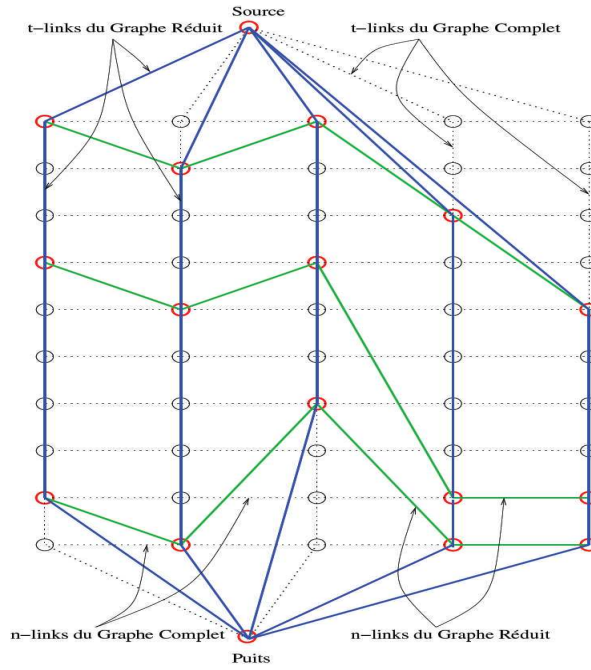


Fig. 2. Construction du graphe réduit [3].

À chaque paire de pixels voisins p et q , les chaînes correspondantes sont reliées par des arêtes ($n-link$) aux $(N - 1)$ niveaux, et de capacité selon l'équation (8).

Capacité de $n-link$ au niveau $i =$

$$u_{\{p,q\}} * (|d_{i,p} - d_{i,q}| + 1) \quad (8)$$

4.3 L'approche proposée

L'approche proposée par Zureiki et al. [3] ne permet pas de traduire la coupure du graphe réduit par une fonction d'énergie analytique et elle ne respecte pas la contrainte de lissage exprimée par le deuxième terme de l'énergie globale (eq: 5), car la capacité des *n-links* n'est pas cohérente avec la formulation de cette énergie globale.

La structure du graphe utilisée dans [3] est à l'origine de ces deux faiblesses. Pour illustrer cela observons l'exemple de la figure 3(a) où la coupure (représentée par un trait pointillé de couleur violette) coupe trois *t-links* (les trois arêtes verticales). Maintenant concentrons nous sur les deux premiers *t-links* (celui qui est le plus à gauche et celui du milieu), ces deux *t-links* représentent des niveaux de disparité différents. Contrairement au graphe complet, dans le graphe réduit les arêtes verticales de même abscisse ne réfèrent pas forcément la même valeur de disparité, car la disparité n'est plus calculée à travers l'indice (*zero-based index*) de l'arête coupée mais plus tôt il y a une correspondance directe entre le coût et la valeur de disparité, comme il a été expliqué dans la section précédente.

Malgré cette différence de disparité il n'y a aucune pénalité qui a été affectée à la coupure, c'est-à-dire que cette structure de graphe ne pénalise pas correctement les variations de disparité entre des pixels voisins. Nous proposons par conséquent une nouvelle structure de graphe introduisant une pénalité qui prend en considération ce cas.

La nouvelle structure du graphe est construite comme ceci: on construit un graphe tel qu'indiqué sur la figure 3(b). La base du graphe est une grille dont les axes verticaux représentent chaque pixel p et les axes horizontaux chaque niveau de disparité possible d_i , par exemple pour $N=4$ on a quatre niveaux. A chaque position (p, d_i) on associe une arête verticale avec la capacité $C + D_{\{p,d_i,p\}}$, où C est une constante qui satisfait la contrainte (7). Les arêtes horizontales correspondent à la fonction de pénalité (notons qu'elle diffère de l'équation (8) dans les indices):

$$u_{\{p,q\}} * (|d_{i,p} - d_{i+1,q}| + 1) \quad (9)$$

ces arêtes horizontales sont appelées *n-link*.

On ajoute à cela des nœuds auxiliaires (couleur orange) entre les nœuds principaux (couleur noire), et des arêtes auxiliaires (couleur rouge) appelées *a-link* qui relient certains nœuds principaux aux nœuds auxiliaires. Ces arêtes auxiliaires sont ajoutées afin de répondre aux problèmes survenus par l'utilisation du graphe dans [3]. On affecte à ces arêtes auxiliaires une capacité selon l'équation (8). Les arêtes qui relient la source à la grille et la grille au puits ont une capacité infinie.

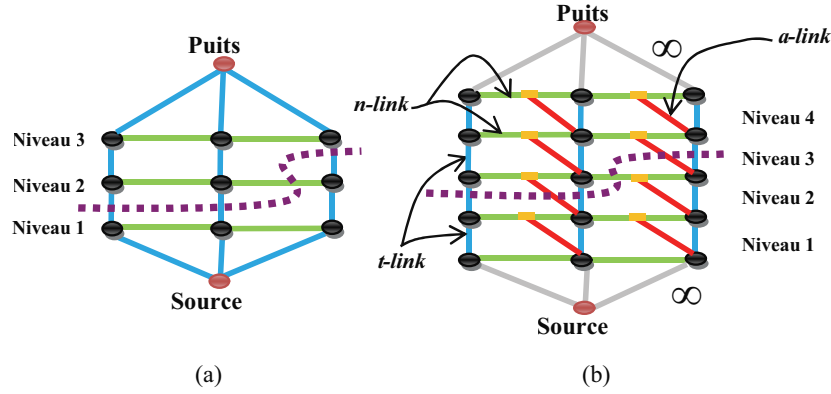


Fig. 3. (a) Graphe réduit selon [3] avec un exemple de coupure, (b) notre graphe réduit avec un exemple de coupure.

Donc si vous remarquez bien, dans la figure 3(b) la coupure est obligée de couper les arêtes de pénalité (*n-links* et *a-links*) dont le coût est proportionnel à la différence des niveaux de disparité entre les pixels voisins.

On dit qu'une coupure est *potentiellement minimale*, lorsque la coupure satisfait la remarque suivante : elle ne coupe aucune arête infinie et ne coupe qu'une et une seule arête d'abscisse p_i .

L'énergie totale minimisée par la coupure de notre nouveau graphe réduit peut être écrite:

$$E(f) = \sum_{\substack{p \in P \\ f_p \in \{d_{p,1}, \dots, d_{p,N}\}}} D_p(f_p) + \lambda \sum_{\substack{\{p,q\} \in N \\ f_p \in \{d_{p,1}, \dots, d_{p,N}\} \\ f_q \in \{d_{q,1}, \dots, d_{q,N}\}}} u_{\{p,q\}} |f_p - f_q| \quad (10)$$

5 Implémentation et résultats expérimentaux

Dans la littérature on trouve deux approches pour résoudre le problème de flot maximal [5]. La première est l'algorithme du *chemin augmentant* dû à Ford et Fulkerson [6], et la deuxième est de type *Push-Relabel*. On a choisi une implémentation du type *chemin augmentant* et plus précisément un nouvel algorithme proposé par Boykov et Kolmogorov [7], ils ont montré que sur des graphes typiques pour la vision, leur algorithme est 2 à 5 fois plus rapide que les autres algorithmes tel que *Push-Relabel*.

Pour comparer l'importance de notre méthode exploitant un graphe réduit, soit W et H la largeur et la hauteur de l'image, $[0, d_{max}]$ la plage de disparités, N le nombre des meilleurs candidats donnés par une méthode locale de mise en correspondance. Soit v le nombre des nœuds et e le nombre des arêtes. Dans le graphe original, on a:

$v = W * H * (d_{max} - 1) + 2$, $e \approx 6 * v = 6 * W * H * (d_{max} - 1)$. La complexité théorique de l'algorithme du *chemin augmentant* est de $O(e * |MaxFlot|)$ [5], où *MaxFlot* est la valeur du flot maximal. Dans le cas du graphe complet, la complexité est de l'ordre $O(6v * |MaxFlot|)$, et puisque dans le graphe réduit on a supprimé un nombre important d'arêtes et de nœuds ceci aura comme conséquence une valeur de complexité beaucoup plus inférieure à celle obtenue avec le graphe complet. Rappelons nous une chose est que les algorithmes de coupure de graphe sont proportionnels (d'ordre 2 ou 3) en nombre de nœuds et d'arêtes, le temps de calcul pour trouver une coupure minimale dans le graphe réduit est très inférieur au temps pour trouver cette coupure dans le graphe complet. Le tableau 1 donne des résultats expérimentaux en temps de calcul. Malgré que notre structure contienne plus de nombre de nœuds et d'arêtes par rapport à [3], nos temps de calcul sont très proches. C'est principalement dû à l'algorithme [7] qu'on a utilisé. Contrairement à [3] où ils ont utilisé l'algorithme de *Push-Relabel*. Mais on a obtenu une nette amélioration en terme de qualité d'image de disparité, ceci est illustré par la figure 4.

Tableau 1. Temps de calcul en secondes pour la mise en correspondance par coupure de graphe complet et réduit.

Taille de l'image	Graphe Complet	Notre Graphe Réduit		Graphe Réduit de [3]	
		N= 4	N= 5	N= 4	N= 5
<i>Tsukuba</i> 384*288	75 s [10]	15 s	30 s	/	/
<i>Sawtooth</i> 434*380	>130 s	16 s	48 s	15 s	50 s

La figure 4 illustre des résultats obtenus sur deux images *Tsukuba* et *Sawtooth* issues de la base d'images de [1] mise à disposition sur leur site <http://cat.middlebury.edu/stereo/>. Les cartes de disparité (vérité terrain) sont aussi tirées de [1]. Notons que la qualité des images de disparité obtenue avec notre approche est supérieure à celle obtenue dans [3] (on n'a pas pu comparer avec l'image *Tsukuba* car les auteurs n'ont présenté leurs résultats qu'avec l'image *Sawtooth*).

Pour *Tsukuba* la taille est de 384*288 et 16 niveaux de disparité, et pour *Sawtooth*, la taille est de 434*380 et 20 niveaux de disparité. Le test a été effectué sur un P4 à 3.2 GHz avec 512 Mo de RAM.

Le tableau 2 présente des résultats quantitatifs (comme cela est fait dans [1]). On note l'absence des résultats de [3] qui n'ont pas présenté des résultats quantitatifs dans leur article.

Tableau 2. Erreurs statistiques.

Méthode	% d'erreurs totales	% d'erreurs >±1
Notre méthode	6.9	1.4
Graphe Complet [8]	10.1	5.9
Graphe Complet [10]	6.7	1.9
Corrélation	28.5	12.8

6 Conclusion

Nous venons de décrire dans cet article, une amélioration d'une méthode fondée sur la coupure de graphe. La combinaison d'une méthode locale, capable de sélectionner pour chaque pixel gauche, un ensemble réduit de correspondants possibles dans l'image droite a permis d'éviter l'explosion combinatoire des méthodes de *graph cuts* exécutées sans élagage préalable du graphe. Notre contribution a consisté à développer un graphe original pour mieux respecter la contrainte de lissage, et permettre de traduire la coupure du graphe réduit par une fonction d'énergie analytique.

Nous pensons que la méthode peut encore être améliorée pour prendre en compte des disparités *sous-pixelliques*, et par la suite construire des cartes de disparité réelles. Nous souhaitons aussi adapter cette méthode globale pour traiter les occultations.

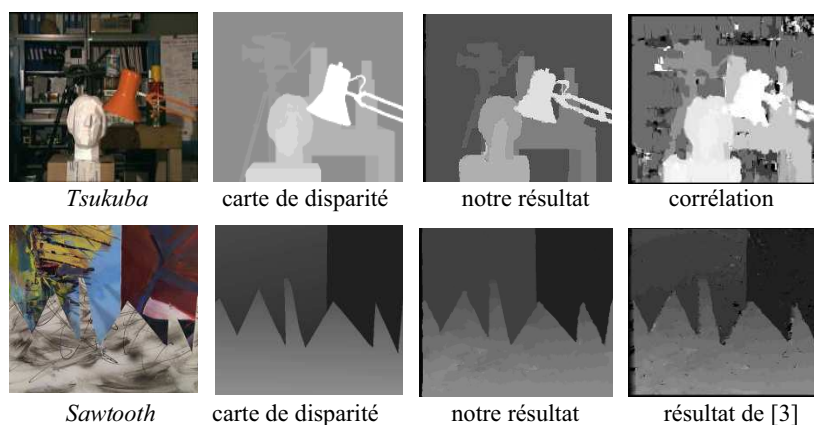


Fig. 4. Les résultats de notre méthode.

Références

1. Scharstein, D., and Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, vol. 47, no. 1-3, pp. 7–42, (2002)
2. Roy, S.: Stereo without epipolar lines: A maximum flow formulation, *IJCV*, vol. 34, no. 2-3, pp. 147–161, (1999)
3. Zureiki, A., Devy, M., and Chatila, R.: Stereo matching using reduced-graph cuts. In *IEEE International Conference on Image Processing (ICIP)*, San Antonio, Texas (USA), September (2007)
4. Ishikawa, H.: *Global Optimization Using Embedded Graphs*, PhD thesis, New York University, (2000)
5. Cormen, T., and Rivest, R., and al.: *Introduction to Algorithms*, Second Edition, the MIT Press, (2001)
6. Ford, L. and Fulkerson, D.: *Flows in Networks*, Princeton University Press, (1962)

7. Boykov, Y., and Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. In IEEE Trans. on PAMI , vol. 26, pp. 1124–1137, (2004)
8. Veksler, O.: Efficient graph-based energy minimization methods in computer vision, Ph.D. thesis, Cornell University, (1999)
9. Roy, S., Cox, I.: A Maximum-Flow Formulation of the N-camera Stereo Correspondence Problem, International Conference on Computer Vision, (1998)
10. Kolmogorov, V., Zabih, R.: Computing Visual Correspondence with Occlusions using Graph Cuts, International Conference on Computer Vision, (2001)
11. Kolmogorov, V., Zabih, R.: Multi-camera Scene Reconstruction via Graph Cuts, European Conference on Computer Vision, (2002)
12. Kolmogorov, V., Zabih, R.: What Energy Functions can be Minimized via Graph Cuts? European Conference on Computer Vision, (2002)