



HAL
open science

Production de carte dense de disparité dans un contexte industriel par stéréo-corrélation d'images sur deux noyaux mono-directionnels

Jimmy Pelcat, Sebastien Kramm, Abdelaziz Bensrhair

► To cite this version:

Jimmy Pelcat, Sebastien Kramm, Abdelaziz Bensrhair. Production de carte dense de disparité dans un contexte industriel par stéréo-corrélation d'images sur deux noyaux mono-directionnels. ORASIS - Congrès des jeunes chercheurs en vision par ordinateur, INRIA Grenoble Rhône-Alpes, Jun 2011, Praz-sur-Arly, France. inria-00610518

HAL Id: inria-00610518

<https://inria.hal.science/inria-00610518>

Submitted on 22 Jul 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Production de carte dense de disparité dans un contexte industriel par stéréo-corrélation d'images sur deux noyaux mono-directionnels

Jimmy Pelcat¹

Sébastien Kramm¹

Abdelaziz Bensrhair¹

¹ Laboratoire d'Informatique, de Traitement de l'Information et des Systèmes (LITIS)

INSA de Rouen
Avenue de l'université - BP8
76801 Saint-Etienne-du-Rouvray - France
jimmy.pelcat@insa-rouen.fr

Résumé

De nombreuses applications de localisation tri-dimensionnelle voient le jour dans le domaine industriel et la stéréovision est une solution technologique qui attire par sa simplicité mécanique. Deux caméras statiques placées à des endroits stratégiques peut s'avérer suffisantes pour répondre à cette problématique. La production de carte dense de disparité est une composante de la stéréovision qui permet d'obtenir une information de profondeur des objets de la scène. Dans cet article, nous présentons une méthode rapide de stéréo-corrélation. Nous montrons les limites de l'utilisation d'un noyau de corrélation carré et proposons une alternative par production de deux cartes denses de disparité à partir de deux noyaux mono-directionnels. Nous évaluons nos performances par une confrontation de nos résultats avec ceux obtenus à partir de la librairie publique OpenCV.

Mots Clef

Stéréovision, calibration, rectification, stéréo-corrélation, carte dense de disparité.

Abstract

Many applications of three-dimensional localization are emerging in the industrial field and the stereo vision is a technology solution attractive by its mechanical simplicity. Two static cameras placed at strategic locations may be sufficient to address this problem. Production of dense disparity map is a component of the stereo vision that provides depth information of objects in the scene. In this paper, we present a fast method of stereo correlation. We show the limits of the use of a squared correlation kernel and propose an alternative production of two dense disparity maps from two unidirectional kernels. We evaluate our performance with a comparison of our results with those obtained from the public library OpenCV.

Keywords

Stereo vision, calibration, rectification, stereo correlation, dense disparity map.

1 Introduction

La localisation 3D est une problématique récurrente dans les applications de guidage de robot. Les acteurs en vision industrielle font face à des demandes clientes pour améliorer et sécuriser leur processus de production. Les applications de "décaissage" ou de "dévraccage", qui consistent à extraire des éléments semblables, ordonnés ou non, présents dans un conteneur, sont un exemple de solutions recherchées chez les constructeurs automobiles notamment. Des solutions émergent dans ce sens. Notamment des solutions 2D associées à un système mécanique pour mettre les éléments sur un même plan connu ou des solutions par triangulation laser mettant en scène une caméra linéaire mobile associée à une mesure laser, ...

Cependant, la solution de stéréovision est une alternative intéressante par sa simplicité mécanique bien qu'elle mette en œuvre des traitements complexes. Nous proposons de répondre à cette problématique par la production de carte dense de disparité, représentative de la profondeur des objets de la scène, faisant appel à des méthodes utilisant des techniques corrélatives pour l'appariement de points.

Notre problématique est double : dans un temps raisonnable, comment obtenir une bonne précision.

Les techniques corrélatives se basent sur l'appariement de points image par minimisation d'un score de corrélation. Ce score est mesuré par la somme des écarts des niveaux de gris des pixels voisins des deux points à appairier. Les méthodes connues proposent un calcul intelligent des scores par glissement de la fenêtre de corrélation. En fait, pour deux calculs successifs du score de corrélation, des opérations ont déjà été réalisées et donc ils se basent sur la mémorisation du score précédent pour calculer le score suivant.

Dans notre contexte applicatif, notre système de stéréovi-

sion n'est pas aligné. Nous apporterons la rectification nécessaire à l'alignement des images afin d'accélérer le processus de stéréo-corrélation.

Dans cet article, nous mettons en évidence que le calcul de corrélation sur un noyau bi-dimensionnel peut être décomposé en deux calculs de corrélation sur des noyaux mono-dimensionnels tout en répondant à la problématique du temps de traitement.

Dans un premier temps, nous proposons une méthode qui a pour objectif d'améliorer la qualité des cartes denses de disparité pour les zones homogènes dans les images par projection d'une lumière structurée. Dans un second temps, nous mettons en évidence les erreurs d'appariement au niveau des zones proches des zones de contours et proposons de la résoudre par fusion de deux cartes denses de disparité construites à partir de deux noyaux différents.

Notre objectif est de garantir la conservation de la forme et de la taille des éléments de la scène.

2 Calibration et Rectification

Dans l'optique de produire des cartes de disparité denses, l'objectif est de mettre en correspondance l'ensemble des points de la scène perceptibles dans les deux images. Néanmoins, la géométrie du système à deux caméras a certaines propriétés qui nous permettent de déterminer, pour chaque pixel de la première image, un ensemble de points candidats dans la seconde image. Ainsi, nous réalisons une calibration par estimation de la géométrie du système : la géométrie épipolaire. De nombreux travaux proposent des méthodes linéaires ou robustes pour l'estimation de la matrice fondamentale à partir d'un ensemble de correspondances répondant à la résolution du système :

$$m_2^T \cdot F \cdot m_1 = 0 \quad (1)$$

Dans notre contexte industriel, cette liste de paires de points est construite à partir de points de mires¹ répartis sur l'ensemble de la scène selon différentes orientations. L'équation est ainsi résolue à partir de méthodes linéaires car l'utilisation de mire garantit un appariement des points sans erreurs.

L'estimation de la géométrie épipolaire nous permet d'exprimer la relation entre chaque point de l'image de gauche et son correspondant dans l'image de droite. En effet, chaque point gauche a son correspondant à droite le long d'une ligne.

La problématique de mise en correspondance est réduite dans la mesure où on passe d'une résolution sur deux dimensions à une résolution sur une seule dimension. Néanmoins, des méthodes de rectification des images sont mises en œuvre afin de garantir l'alignement dans les images de tous les points similaires et, au final, la mise en correspondance est grandement simplifiée.

1. Les mires, sous forme d'un damier présentant une marque qui définit l'origine et les deux axes du repère, sont propriétés de la Société Cognex

Nous avons utilisé des méthodes de rectification récentes [8] car elles permettent de minimiser la déformation des images, nécessaire pour garantir la mise en correspondance par mesure de ressemblance sur des fenêtres englobantes.

3 Production de carte dense de disparité

Dans la littérature, on note deux approches pour construire des cartes denses de disparité : les méthodes par coupure de graphe dites "*graph cuts*" [2] [5] [11] et les méthodes corrélatives rapides [4] [9] [13].

Les méthodes par coupure de graphe consistent à construire un graphe où les pixels sont les nœuds et les arcs sont les mesures de similarité entre les pixels. L'objectif est de construire un ensemble de sous-graphes correspondant à des régions distinctes de la scène. Ce sont des méthodes itératives qui, dans notre contexte industriel, ne sont pas exploitables car les temps de traitement sont de plusieurs dizaines de secondes pour des images de faible résolution et sur un intervalle de disparité réduit [7].

C'est ainsi que nous avons privilégié les méthodes corrélatives qui consistent à minimiser un critère de corrélation entre un point gauche et un ensemble de points droits candidats à la mise en correspondance. Il existe différents indices de corrélation (SSD, CC, ZSSD, ZCC, ZNSSD, ZNCC) définis dans [3]. Leur complexité croissante améliore l'appariement mais impose des temps de traitement plus importants.

Dans notre contexte industriel, nous maîtrisons les différences de luminosité entre nos deux images ou plutôt nous faisons en sorte qu'il n'y ait pas d'écart de luminosité. C'est ainsi que notre choix se porte sur l'indice de corrélation SSD qui, par la même occasion, requiert des temps de traitement moindres.

Le SSD (*Sum of Squared Differences*) entre deux pixels est la somme des différences au carré de l'ensemble des pixels du noyau (fenêtre de corrélation de taille k avec $k = 2n + 1$, $n \in \mathbb{Z}$) respectivement dans l'image de gauche et dans l'image de droite. Par ailleurs, on notera SD_i comme la différence au carré entre deux pixels ayant une disparité d_i .

Le SDD s'exprime donc par la relation :

$$SSD_{x,y}(d) = \sum_{i=-\frac{k-1}{2}}^{\frac{k-1}{2}} \sum_{j=-\frac{k-1}{2}}^{\frac{k-1}{2}} (I_g(x, y) - I_d(x + d, y))^2 \quad (2)$$

Cependant, lors de la construction de cartes denses de disparité, des opérations identiques sont réalisées de nombreuses fois. On dénombre que pour deux points fixés gauche et droite, leur SD intervient dans k^3 calculs de SSD. Le principe de base de notre méthode est de construire une image SSD et une image de disparité qui sont mises à jour à chaque disparité d_i considérée. Ainsi, sur un intervalle de disparité $[d_{min} \dots d_{max}]$, nous construisons une image

SSD_i correspondant au résultat du calcul du SSD pour une disparité fixée d_i . (voir fig. 1)

Nous proposons une méthode qui se base sur deux principes : la séparabilité du calcul de SSD_i et la mémorisation du SD précédent pour calculer le SD suivant.

3.1 Séparabilité

Pour optimiser le traitement, notre méthode met en évidence la séparabilité du calcul d'une image de SSD_i . En effet, au lieu de calculer l'indice de corrélation sur toute la fenêtre, on décompose l'opération en une opération en ligne suivi d'une opération en colonne. La décomposition du calcul du SSD_i en deux opérations nous amène à construire une image temporaire $SSDL_i$, résultat de l'opération en ligne. Ainsi, nous exprimons $SSDL_i$ et SSD_i par :

$$SSDL_i(x, y) = \sum_{j=x-\frac{k-1}{2}}^{x+\frac{k-1}{2}} SD_i(j, y) \quad (3)$$

et

$$SSD_i(x, y) = \sum_{j=-\frac{k-1}{2}}^{\frac{k-1}{2}} SSDL_i(x, y) \quad (4)$$

3.2 Mémorisation

Les méthodes corrélatives [4] [9] [13] mettent en œuvre la réutilisation des scores préalablement calculés sur tout le noyau. La séparabilité permet également de conserver le calcul intelligent des $SSDL_i$ avec mémorisation du calcul précédent afin de réduire le nombre d'opérations. Il intervient à la fois sur le calcul en ligne et sur le calcul en colonne. Notre méthode se différencie par la propagation du calcul de $SSDL_i$ d'une ligne à l'autre car nous admettons que les bords de l'image, de la largeur du noyau, soient entachés d'erreurs. Les images étant enregistrées dans un buffer mono-dimensionnel (les lignes sont les unes derrière les autres), la propagation horizontale en découle logiquement.

Pour le calcul en ligne (voir fig. 2), on calcule le $SSDL_i$ du premier pixel et les pixels suivants sont calculés suivant :

$$SSDL_i(x+1, y) = SSDL_i(x, y) - SD_i(x - \frac{k-1}{2}, y) + SD_i(x+1 + \frac{k-1}{2}, y) \quad (5)$$

L'illustration de l'algorithme pour le calcul de $SSDL_i$ reflète le calcul pour une disparité $d_i = 0$. Néanmoins, pour les autres disparités, le principe est le même mais on ajoute un décalage de la valeur de la disparité sur l'une des deux images en fonction du signe de la disparité.

Pour le calcul en colonne (voir fig. 3), on calcule le SSD_i pour la première ligne de l'image puis on propage le calcul des pixels suivants tel que :

$$SSD_i(x, y+1) = SSD_i(x, y) - SSDL_i(x, y - \frac{k-1}{2}) + SSDL_i(x, y+1 + \frac{k-1}{2}) \quad (6)$$

3.3 Mise à jour

Le calcul de SSD_i est l'image de corrélation pour la disparité d_i dont les valeurs sont comparées à l'image SSD. Pour les points de SSD_i dont la valeur est inférieure à ceux de SSD , nous faisons une mise à jour de la carte de disparité avec la valeur de disparité courante d_i et une mise à jour de l'image SSD (voir fig. 4).

3.4 Conséquences

L'apport de notre méthode réside dans notre quasi indépendance à la taille du noyau. En effet, la taille du noyau n'intervient qu'à l'initialisation de la première valeur lors de l'opération en ligne et à l'initialisation de la première ligne lors de l'opération en colonne. Au final, les temps de traitement pour la production de notre carte dense de disparité sont uniquement proportionnels à la taille de nos images et à l'intervalle de disparité considéré correspondant au nombre de SSD_i construites et de mises à jour de SSD .

3.5 Validation croisée

Lors de la mise en correspondance, un point gauche est apparié avec le point droit à partir desquels le score de corrélation est minimal mais il faut s'assurer que le point droit minimise le score de corrélation avec ce même point gauche. Pour cela nous produisons deux cartes de disparité, gauche vers droite puis droite vers gauche. La carte dense de disparité finale est la fusion de ces deux images par conservation des disparités opposées dans l'intervalle des disparités.

Soit $[d_{min} \dots d_{max}]$ l'intervalle de disparité dans le sens gauche vers droite. Alors, l'intervalle de disparité dans le sens droite vers gauche sera $[-d_{max} \dots -d_{min}]$. Ainsi, la fusion doit vérifier, pour une disparité d_{gauche} dans le premier intervalle et une disparité d_{droite} dans le second, l'égalité $d_{gauche} = -d_{droite}$.

pour $y = 0$ à hauteur faire

pour $x = 0$ à largeur faire

si

$ImgDisp_{gauche}(x, y) = -ImgDisp_{droite}(x, y)$

alors

$ImgDisp(x, y) = ImgDisp_{gauche}(x, y)$

sinon

$ImgDisp(x, y) = 0$

4 Amélioration qualitative

Les cartes denses de disparité ainsi construites mettent en évidence plusieurs problématiques :

- nous avons des "sauts" de disparité sur la surface des objets
- nous avons de nombreuses erreurs d'appariement en zone homogène,

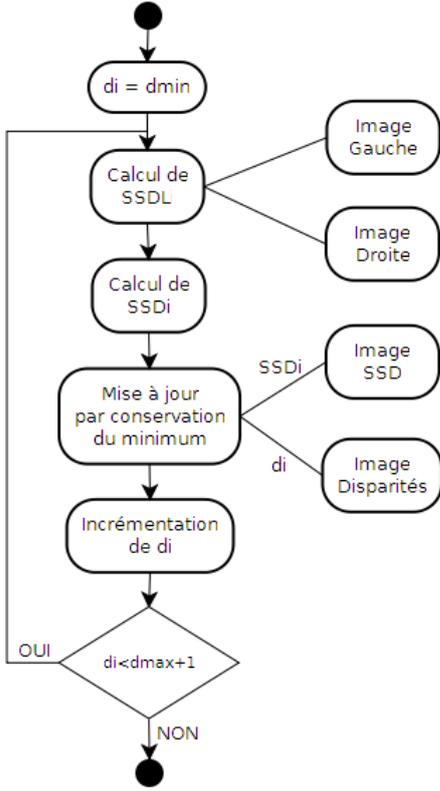


FIGURE 1 – Pseudo-algo du processus de production de carte dense de disparité

– nous avons de nombreuses erreurs d'appariement sur les zones proches des contours qui se traduit par un effet de dilatation (voir fig. 5(b) à fig. 5(d)).

4.1 Mesure subpixelique

Afin d'améliorer la résolution de nos images de disparité, nous avons mis en place la fonction d'interpolation parabolique [3] qui utilise les deux scores de SSD "voisins". En effet, si deux points (x,y) et $(x + d_e,y)$ minimise le SSD, on calcul d_f tel que :

$$d_f = d_e + 0.5 \frac{SSD(d_e + 1) - SSD(d_e - 1)}{2 \cdot SSD(d_e) - SSD(d_e + 1) - SSD(d_e - 1)} \quad (7)$$

Pour le réaliser, nous conservons l'ensemble des images SSD_i . Si deux points minimise le SSD pour $d = d_i$ alors nous utilisons les trois scores de SSD_{i-1} , SSD_i et SSD_{i+1} pour calculer la valeur subpixelique de la disparité.

4.2 Zones homogènes

Nous proposons d'utiliser la projection d'une image bruitée sur la scène afin de créer de la texture sur les éléments homogènes et cela de manière stéréoscopique. L'utilisation d'une lumière structurée est courante pour les applications où la scène ne présente pas suffisamment de texture

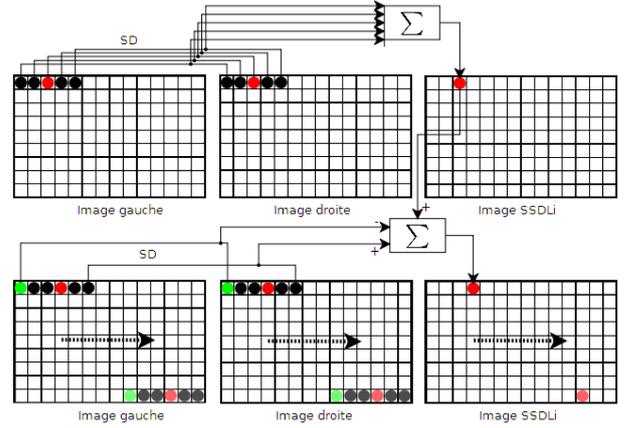


FIGURE 2 – Initialisation de la première valeur puis propagation horizontale pour le calcul en ligne (cas où $d_i = 0$)

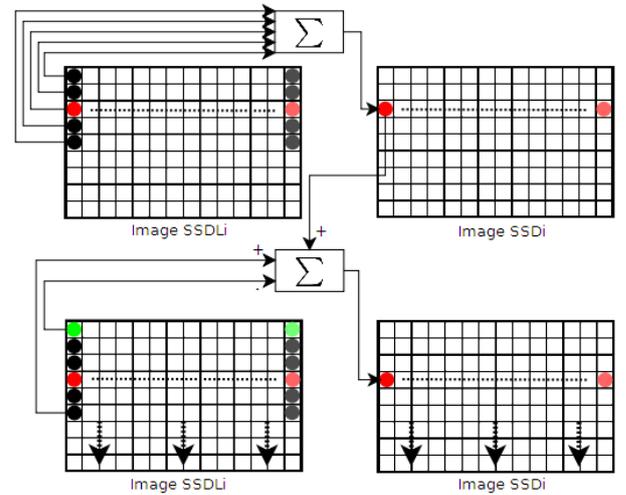


FIGURE 3 – Initialisation de la première ligne puis propagation verticale pour le calcul en colonne

pour satisfaire la stéréo-corrélation. Nous retrouvons notamment des travaux utilisant la projection de franges [12] ou de mouchetis [6] [10].

4.3 Effet de dilatation

L'effet de dilatation exprime l'erreur de calcul de la disparité près des zones de contours. En effet, plus le noyau est grand, plus l'effet de dilatation est important (voir fig. 5(b) à fig. 5(d)). Cet effet est particulièrement présent à proximité des contours avec une grande transition et présentant des zones homogènes. Les figures 6(a) et 6(b) montrent trois appariements de points pour trois noyaux différents. La problématique est que ces appariements engendreront la même valeur de disparité mais cette valeur sera affectée à un pixel différent, le point central du noyau considéré. Dans l'exemple, les points, plus ou moins éloignés de l'objet (rouleau) en fonction de la taille du noyau, auront une disparité proche de celle de l'objet lui-même alors qu'il devraient avoir la disparité du support.

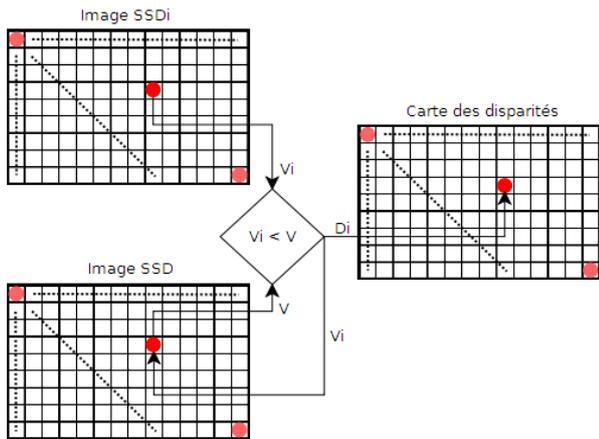


FIGURE 4 – Mise à jour de l’image SSD et de la carte dense de disparité

Pour réduire l’effet de dilatation qui est proportionnel à la taille du noyau, nous proposons de reconsidérer l’utilisation d’une fenêtre englobante carrée. En effet, elle engendre un effet de dilatation dans toutes les directions. C’est ainsi que notre approche se base sur la construction de deux cartes de disparité, l’une utilisant un noyau "ligne" et l’autre un noyau "colonne". Ces deux cartes sont produites de gauche vers la droite et nous ne mettons pas en œuvre le principe de validation croisée qui sera effectuée à l’étape suivante. L’objectif principal est que l’effet de dilatation sera uniquement horizontal pour le cas en ligne et vertical pour le cas en colonne.

La carte dense de disparité finale est une combinaison de ces deux images par comparaison des valeurs de disparité. Si la disparité dans les deux images est identique alors nous conservons cette disparité sinon on considère que nous n’avons pas de disparité. Au final, c’est la fusion des deux cartes qui réalise une validation croisée.

```

pour  $y = 0$  à hauteur faire
  pour  $x = 0$  à largeur faire
    si  $ImgDisp_{ligne}(x, y) = ImgDisp_{colonne}(x, y)$ 
      alors
        |  $ImgDisp(x, y) = ImgDisp_{ligne}(x, y)$ 
      sinon
        |  $ImgDisp(x, y) = 0$ 

```

Nous proposons également de conserver une certaine tolérance sur l’effet de dilatation. En effet, l’appariement, uniquement sur un noyau mono-dimensionnel, engendre davantage d’erreur que sur un noyau bi-dimensionnel dans la mesure où sa taille est k fois inférieure. La tolérance représente la hauteur du noyau "colonne" et également la largeur du noyau "ligne". Finalement, si on accepte un effet de dilatation de T pixels alors on utilisera un noyau "rectangulaire" de taille (T, k) et un autre noyau "rectangulaire" de taille (k, T) . (k étant la taille du noyau original). La to-

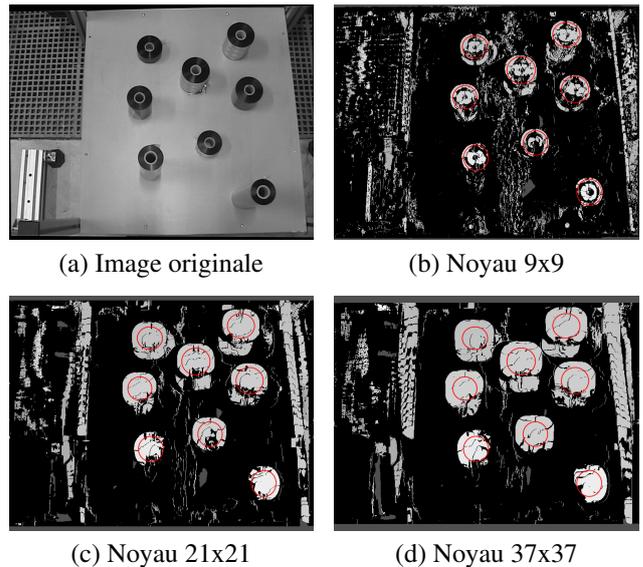


FIGURE 5 – Visualisation de l’effet de dilatation par rapport au contour supérieur des rouleaux en rouge

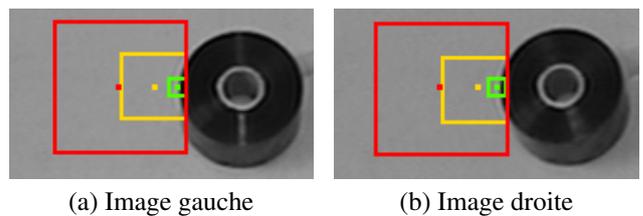


FIGURE 6 – Justification de l’effet de dilatation avec 3 noyaux de taille différente

lérance n’a que très peu d’influence sur le temps de traitement dû à notre quasi indépendance à la taille du noyau lors du calcul du SSD_i .

5 Expérimentation et résultats

5.1 Alignement des images

L’évaluation des performances de notre approche est basée sur l’utilisation d’images issues de caméras proches de l’alignement stéréoscopique. Nous avons donc calibré notre système afin d’apporter une rectification des images, nécessaire au processus de production de carte de disparité. A l’issue de la phase de rectification basée sur la méthode minimisant la déformation des images [8], nous présentons un exemple de résultats à la figure 7. Pour un ensemble de 3000 points de mesure répartis de manière homogène dans le volume étudié, près de 90% de ces points (vert) ont un écart inférieur au demi pixel et 99,3% (vert et jaune) ont un écart inférieur au pixel. Les erreurs les plus importantes (rouge) sont les points les plus éloignés du point central de la scène et donc leur influence est moindre par rapport au contexte applicatif.

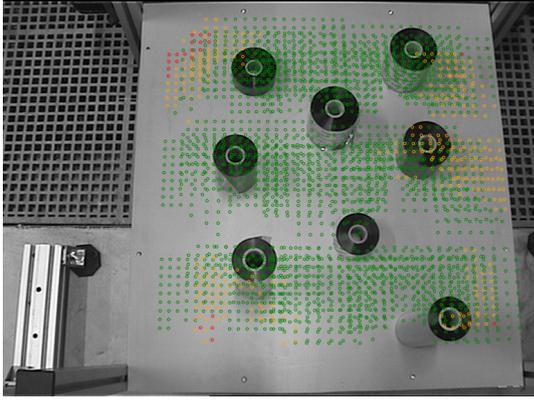


FIGURE 7 – Répartition de l'erreur de rectification

5.2 Implémentation

Pour l'implémentation de notre algorithme de production de carte dense de disparité, nous construisons nos deux cartes de disparité, respectivement avec un noyau en ligne et un noyau en colonne, sur deux cœurs de processeur différents. Une série de mesure nous permet d'estimer un gain de 1,8 en temps de calcul sur l'ensemble du traitement.

Aussi, nous utilisons le jeu d'instructions SSE4 disponible sur les processeurs Intel P4 et AMD, afin d'utiliser des registres XMMX 128 bits. Cela nous permet d'accélérer le traitement en parallélisant certaines parties de l'algorithme. Lors de la phase d'initialisation des trois images de disparité (à des valeurs nulles), le XMMX permet de traiter 16 octets de données en une seule instruction. Le pixel étant représenté sur un octet, on peut traiter les images de disparité par bloc de 16 pixels, soit un gain de 16 par rapport à une implémentation classique. Lors de la phase d'initialisation de l'image SSD (à des valeurs maximales) et lors de propagation verticale dans SSD_i , la taille des données à traiter dépend de la taille du noyau. Dans notre cas, nous utilisons l'indice de corrélation SSD qui est compris entre 0 et $k^2 255^2$ et donc nous montrons que, pour un noyau carré compris entre 3 et 257, les données à traiter sont sur quatre octets (long), soit un gain potentiel de quatre ($16/4 = 4$). L'apport en temps de calcul pour notre traitement est évalué à un gain de l'ordre de 1,3.

5.3 Projection de mouchetis

Les prochains résultats montrent l'évolution entre la production de carte dense de disparité à partir de l'image originale (voir fig. 8(a)) et celle à partir de l'image issue de la scène sur laquelle nous avons projeté une image de bruit (voir fig. 8(b)). Ensuite, nous montrons les résultats issus de la correction de l'effet de dilatation.

Les images rectifiées utilisées sont de taille 488 par 489 (238632 pixels) et l'intervalle de disparité est $[-40..30]$, soit un total de 70 valeurs de disparité à considérer.

Nous pouvons voir sur la figure 8(c) et 8(d), l'apport du mouchetis sur notre scène par rapport aux zones homogènes. On dénombre pour l'image originale 140863 points

non appariés soit 59%, contre 105471 pour l'image mouchetée, soit 44,2%.

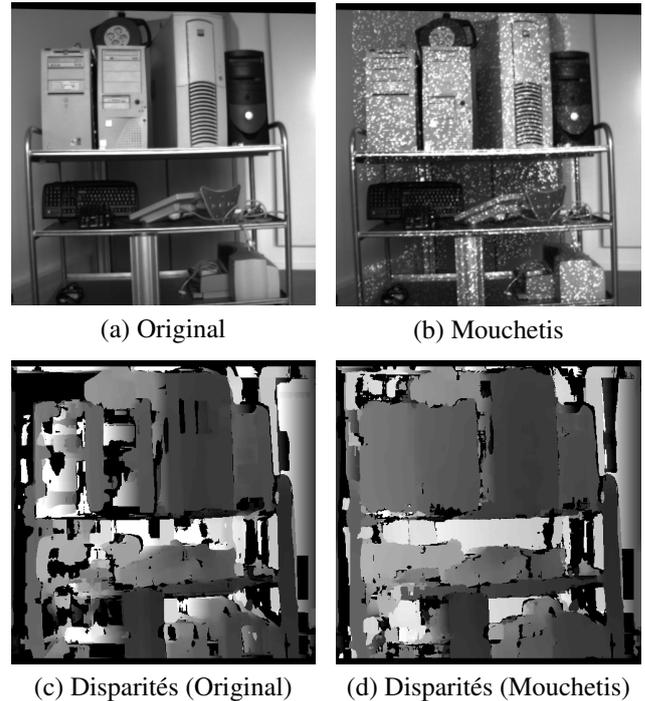


FIGURE 8 – Influence du mouchetis pour un noyau 21x21 (utilisation du SSD avec validation croisée)

Les images sont le résultat de la production de carte dense de disparité à partir de notre méthode sans prise en compte de l'effet de dilatation. Afin de supprimer les erreurs d'appariement, nous avons utilisé le procédé de validation croisée.

5.4 Correction de l'effet de dilatation

Pour évaluer notre algorithme visant à limiter l'effet de dilatation, nous le confrontons aux résultats obtenus avec la librairie OpenCV [1]. La fonction de production de carte de disparité intégrée dans OpenCV est basée sur l'indice de corrélation SAD (*Sum of Absolute Differences*). Cet indice est différent du nôtre mais, dans une problématique de minimisation, le SAD et le SSD ont des résultats équivalents. La différence entre les 2 méthodes réside dans le fait que nous n'avons pas l'effet de dilatation de la surface. Nous conservons donc la forme rectangulaire et nous n'avons pas d'estimation de la disparité sur les zones d'occultation. (voir fig. 9)

Nous avons quantifié le nombre de points qui représentent la forme rectangulaire afin d'évaluer le pourcentage de conservation des formes (voir fig. 10). Nous avons évalué nos performances selon le critère de tolérance et nous obtenons des écarts, avec le nombre de points issus de l'image de vérité terrain, inférieurs au demi pourcent (soit une cinquantaine de points) alors que la méthode de OpenCV dépasse les 2% (autour de 400 points d'écart).

Aussi, l'effet de dilatation augmente proportionnellement pour les cartes denses de disparité issues de OpenCV alors que notre algorithme est insensible à cette problématique. (voir fig. 11)

On peut noter l'influence de la tolérance (voir fig. 12). Plus la tolérance augmente, moins la carte de disparité est bruitée mais moins l'estimation des contours est précise.

6 Conclusion

Différentes approches pour produire des cartes denses de disparité existent mais les méthodes par stéréo-corrélation permettent d'avoir un bon compromis entre la qualité des images et le temps de traitement. Notre méthode pour produire des cartes denses de disparité est rapide en étant indépendante de la taille du noyau utilisé et permet de conserver les formes en occultant les zones de dilatation. Notre méthode garantit un bon appariement des points image proches des zones de contours, supprimant totalement les effets de dilatation engendrés par l'utilisation même de la stéréo-corrélation classique. L'approche par décomposition du noyau de corrélation en deux noyaux plus simples montre de meilleures performances que la librairie OpenCV, une des références dans le domaine.

Références

- [1] <http://opencv.willowgarage.com>.
- [2] Y. Boykov, O. Veksler, and R. Zabih. Markov random fields with efficient approximations. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 648–655, 1998.
- [3] F. Devernay. *Vision stéréoscopique et propriétés différentielles des surfaces*. PhD thesis, Ecole Polytechnique, France, février 1997.
- [4] L. Di Stefano, M. Marchionni, and S. Mattoccia. A fast area-based stereo matching algorithm. *Image and Vision Computing*, 22(12):983–1005, 2004.
- [5] H. Ishikawa and D. Geiger. Segmentation by grouping junctions. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 125–131, 1998.
- [6] S.B. Kang, J. Webb, L. Zitnick, and T. Kanade. A multibaseline stereo system with active illumination and real-time image acquisition. *ICCV*, pages 88–93, 1995.
- [7] V. Kolmogorov and R. Zabih. Visual correspondence with occlusions using graph cuts. *International Conference on Computer Vision*, pages 508–515, 2001.
- [8] J. Mallon and P.F. Whelan. Projective rectification from the fundamental matrix. *Image and Vision Computing*, 23(7):643–650, 2005.
- [9] K. Mühlmann, D. Maier, J. Hesser, and R. Männer. Calculating dense disparity maps from color stereo images, an efficient implementation. *IJCV*, 47(1-3):79–88, 2002.
- [10] V. Rodehorst, A. Koschan, and K. Spiller. Color stereo vision using hierarchical block matching and active color illumination. *International Conference on Pattern Recognition*, pages 835–839, 1996.
- [11] S. Roy. Stereo without epipolar lines : A maximum flow formulation. *International Journal of Computer Vision*, 1(2):1–15, 1999.
- [12] D. Scharstei and R. Szelisk. High-accuracy stereo depth maps using structured light. *CVPR*, 1:195–202, 2003.
- [13] O. Veksler. Fast variable window for stereo correspondence using integral images. *CVPR*, pages 556–561, 2003.

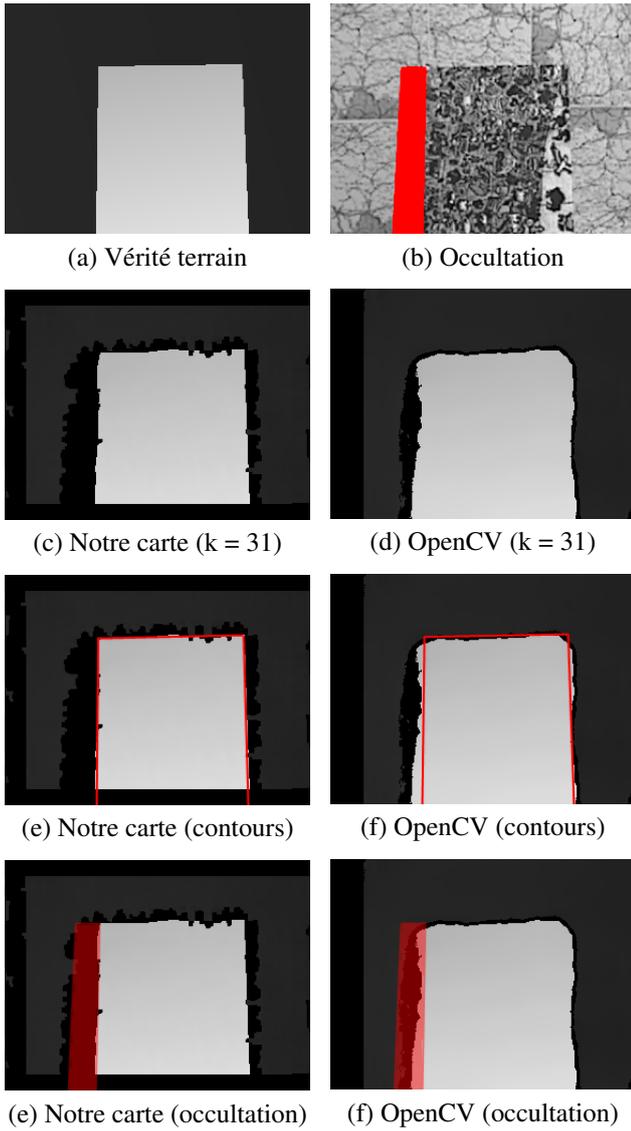


FIGURE 9 – Correction de l'effet dilatation pour différents noyaux et comparaison avec les résultats issus de OpenCV (<http://vision.middlebury.edu/stereo/data/>)

	Tolérance	Noyau k = 21		Noyau k = 31		Noyau k = 41	
Vérité terrain		18832 pts		17422 pts		16014 pts	
Fusion	3	18633 pts	1,06%	17248 pts	1,00%	15947 pts	0,42%
	5	18864 pts	0,17%	17494 pts	0,41%	16167 pts	0,96%
	7	19151 pts	1,69%	17725 pts	1,74%	16316 pts	1,89%
OpenCV		19370 pts	2,86%	17786 pts	2,09%	16337 pts	2,02%

FIGURE 10 – L'écart en pourcentage du nombre de points appartenant à la forme rectangulaire et comparaison avec les performances de OpenCV

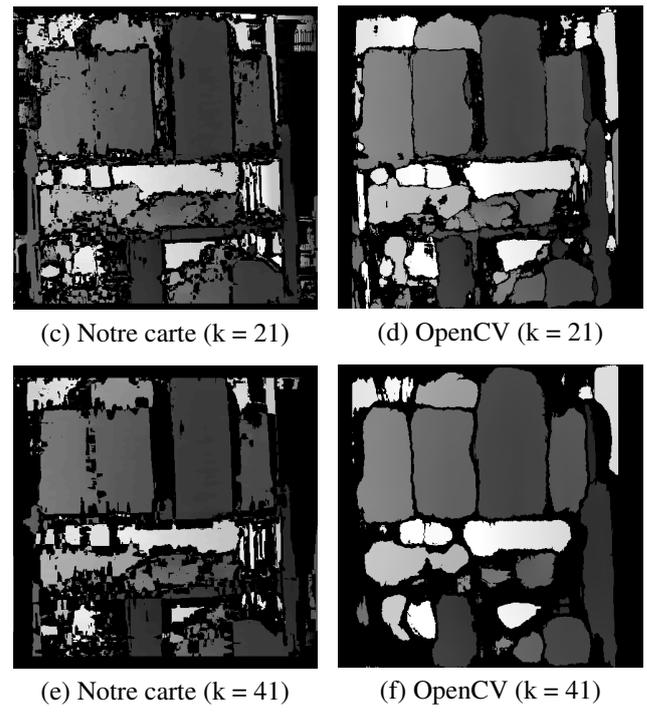


FIGURE 11 – Évolution de l'effet de dilatation en fonction du noyau et comparaison avec OpenCV

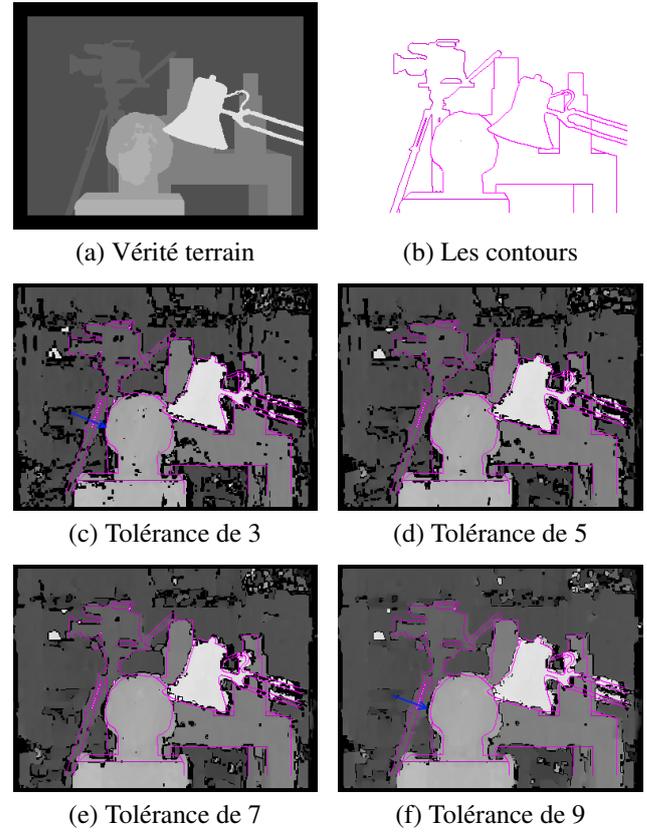


FIGURE 12 – Influence de la tolérance (<http://vision.middlebury.edu/stereo/data/>)