

Planification de la trajectoire optimale du bras robotisé d'une machine de réparation de fissures

Steven A. VELINSKY

Université de Californie, Davis (États-Unis)

René MATHURIN*

Laboratoire Central des Ponts et Chaussées

Daniel-Marc DUCROS

Laboratoire Central des Ponts et Chaussées

RÉSUMÉ

Traitement d'images et optimisation combinatoire des calculs sont utilisés ici pour accroître l'ergonomie du module de commande d'un engin de travaux publics dédié à la réparation de fissures dans les chaussées. L'intervention de l'opérateur est ainsi limitée à la sélection des extrémités de segments de fissures à réparer. Le bras automatisé de cet engin supporte un dispositif qui injecte du bitume afin d'obstruer les fissures. Le pilotage précis de ce bras nécessite deux phases de traitement : l'extraction des coordonnées des points matérialisant la fissure et la détermination d'un ordre de réparation lorsque la zone considérée présente plusieurs fissures. La localisation des fissures est réalisée par un algorithme de vision, simple et robuste, fondé sur une technique de mise en correspondance de bloc d'images. La planification de la trajectoire est assimilée à la recherche d'un *plus court chemin* dans un graphe. S'agissant d'un problème algorithmique classique, une méthode d'optimisation combinatoire *ad hoc* est mise en œuvre pour estimer cette trajectoire. Quelques images, prises en condition normale de fonctionnement, montrent l'efficacité de l'approche algorithmique retenue.

DOMAINE : Sciences de l'ingénieur pour le génie civil.

ABSTRACT

Image processing and combinatorial optimization of computations are used herein to improve the ergonomics of the control unit on a pavement crack repair machine. Operator action is thereby limited to selecting the ends of the crack segments for repair. The machine's automatic arm carries a device which injects bitumen into cracks in order to fill them. The precise control of this arm requires two phases of data processing : extraction of the coordinates of the points marking the crack and determination of the order of repair when the zone in question contains several cracks. Cracks are located by means of a simple and robust vision algorithm based on an image block matching technique. The path is planned by searching for the shortest path in a graph. As this is a traditional algorithm problem a specific combinatorial method is employed to estimate the path. The effectiveness of the selected algorithm technique is shown by a number of images taken under normal operating conditions.

Introduction

L'Advanced Highway Maintenance and Construction Technology Center (AHMCT), centre de recherche commun à l'université de Californie à Davis et au California Department of Transportation (Caltrans), a pour vocation d'accroître la sécurité des chantiers de travaux publics. Les travaux de maintenance étant généralement réalisés sous trafic, ouvriers et automobilistes sont concernés. Un moyen de réduire le nombre d'accidents corporels sur les chantiers est de limiter l'intervention humaine. Aussi, ce centre a entrepris le développement de plusieurs machines pour automatiser les activités d'entretien, d'auscultation et de construction des infrastructures routières. Les travaux décrits dans cet article résultent d'une collaboration entre le AHMCT et le Laboratoire Central des Ponts et Chaussées (LCPC). Cette collaboration porte principalement sur le développement de nouvelles techniques d'instrumentation. Plus concrètement, il s'agit d'une contribution à la mise au point d'une machine automatique de réparation de fissures (*Crack Sealing Machine* ou CSM). Au volet sécurité évoqué précédemment, s'ajoute souvent l'aspect économique pour justifier l'automatisation d'une tâche de maintenance. Actuellement, la réparation des fissures est un travail fastidieux, qui demande une main d'œuvre importante. En Californie, Caltrans mobilise une équipe de huit ouvriers pour réaliser cette opération. Le rythme des réparations est jugé plutôt lent ; il se situe dans une fourchette comprise entre un et trois kilomètres de chaussées par jour. L'État de Californie entretient 24 000 km d'autoroute. De plus, une réparation manuelle entraîne une trop grande variabilité de la qualité d'exécution.

* Actuellement à la Délégation générale pour l'armement (Ministère de la Défense).

Après avoir réalisé plusieurs prototypes (Velinsky, 1993), le *AHMCT* termine actuellement le développement de l'ultime version avant le lancement d'une présérie. Ce prototype, nommé *OCCSM* pour *Operator Controlled Crack Sealing Machine*, est doté d'un bras automatisé de grande dimension (fig. 1a) capable de couvrir une surface de 4 × 4 m de chaussée. Les difficultés de conception sont concentrées dans le contrôle précis du déplacement de ce bras.

La figure 1b montre un prototype de conception plus ancienne. Pour celui-ci, les réparations des fissures longitudinales et transversales sont deux opérations disjointes : un bras est dédié à chaque typologie de fissure. Dans la nouvelle version, le bras télescopique, situé à l'arrière du camion, doit être en mesure de réparer une fissure, quelle que soit son orientation dans le plan. Les aspects mécaniques et les spécificités du contrôle/commande de ce bras sont exposés dans les articles de Feng et al. (1998a et 1998b).

Fig. 1 - Machine automatique de réparation de fissures.



a. Prototype de Crack Sealing Machine.
Le bras, composé de deux segments télescopiques, est monté sur un axe qui lui permet de pivoter sur près de 180 degrés.
L'extrémité du bras supporte une brosse métallique rotative pour décaper les abords de la fissure et une buse pour y injecter le bitume. Les images issues d'une caméra CCD, fixée sur un bras déployé manuellement au-dessus de la chaussée, sont utilisées pour guider le bras.



b. Prototype de conception ancienne.
Le bras situé sur le flanc droit du véhicule est dédié aux fissures longitudinales alors que le bras situé à l'arrière se charge des fissures transversales.

La première partie de cet article présente les algorithmes de vision permettant d'extraire les coordonnées des fissures. Si la mise en correspondance de blocs d'images constitue la base algorithmique, nous montrerons que, pour garantir la cohérence du suivi de la fissure, il est nécessaire d'y associer une « mémoire à court terme » des positions explorées. La deuxième partie est dédiée au problème de planification de la trajectoire du bras. Nous présenterons la méthode utilisée et nous justifierons notre choix par rapport aux contraintes d'utilisation de l'*OCCSM*. En dernier lieu, quelques images prises en situation défavorable montreront la robustesse de la méthode proposée.

Suivi de fissure par traitement d'images

Le traitement d'images appliqué à l'extraction de fissures a été largement étudié au sein du réseau des LPC, notamment par le service « Métrologie et Instrumentation » (SMI) du LCPC et par le Laboratoire régional des Ponts et Chaussées (LRPC) de Strasbourg. Il est important de noter que certains algorithmes, nécessaires à la mise en œuvre de ce type d'application, sont implantés dans

le logiciel PICTURE développé par le SMI. Plus généralement, la détection et la quantification, semi-automatique (Kim et *al.*, 1998) ou automatique (Kirschke et Velinsky, 1992 ; Mathurin et *al.*, 1994 ; Cheng et *al.*, 1999 ; Pynn et *al.*, 1999), de la fissuration de surface pour des matériaux comme le béton ou le bitume, ont largement été explorées par la communauté des « traiteurs d'images ». Cette problématique est par ailleurs partagée par d'autres applications, à des échelles différentes : l'extraction des réseaux routiers ou des réseaux hydrographiques dans des images aériennes et satellitaires, ou encore des vaisseaux sanguins en imagerie médicale (Rellier et *al.*, 2000 ; Radu Stefan, 2001). Néanmoins, aucune approche fédératrice ne semble émerger. Réaliser ces opérations demeure un problème ouvert dès lors que les conditions de prise de vue sont mal maîtrisées (par exemple, variation de l'intensité lumineuse, ombre portée, etc.) ou lorsque des objets parasites (tache d'huile, ligne blanche, etc.) *polluent* la scène. L'approche proposée, sans pour autant résoudre le problème de généralité, doit satisfaire à deux contraintes : **simplicité** et **robustesse**. **Simplicité**, car l'algorithme doit s'intégrer « *on line* » dans la boucle de contrôle/commande du bras et les temps de traitement doivent être acceptables, de l'ordre de la seconde, voire inférieurs. **Robustesse**, car il doit s'adapter aux variations de prise de vue imposées par une utilisation sur route.

L'application impose d'opérer un suivi semi-automatique de la fissuration. Toutes les fissures ne sont pas nécessairement à réparer. L'opérateur visualise et définit, sur son écran tactile, des points de « passages obligés » : généralement des extrémités de fissures ou des segments de fissure lorsque celles-ci se ramifient. Cette phase manuelle constitue un raccourci algorithmique appréciable et facilite le processus d'extraction des fissures.

L'algorithme de suivi proposé, nommé SACEa (*Semi Automatic Crack Extraction algorithm*) est fondé sur deux mécanismes complémentaires :

- i) le suivi de la fissure par une méthode classique de corrélation,
- ii) la mise en œuvre d'une mémoire dynamique pour éviter les « retours en arrière », garantissant ainsi la convergence vers la seconde extrémité de la fissure.

Mise en correspondance de blocs

Le suivi de la fissure qui constitue la première étape de SACEa est fondé sur la technique bien connue de mise en correspondance de blocs d'images – ou fenêtres. Celle-ci revient à considérer une sous-image de taille $M \times N$ – appelée *fenêtre d'observation*, notée $w(x, y)$ – et à rechercher dans un voisinage proche – *l'espace de recherche* H –, une sous-image – *une fenêtre de recherche* notée $s(u, v)$ – ayant des caractéristiques statistiques proches. La position de la fenêtre d'observation est définie manuellement par l'utilisateur lorsqu'il désigne, au moyen d'une souris, le début de la fissure. La fenêtre de recherche, de dimension identique, sera déplacée, avec recouvrement, autour de la fenêtre d'observation. La corrélation 2D s'opère en recherchant dans H une position $s(u, v)$ qui *minimise* un critère de ressemblance entre ces deux fenêtres. Une fonction de coût $C(u, v)$ est calculée pour chaque position (u, v) de s . Une façon classique de comparer $w(x, y)$ et $s(u, v)$ est de calculer l'écart quadratique moyen entre ces deux fenêtres. Cet écart, noté rms , est défini par :

$$rms = \left[\frac{1}{M \times N} \sum_{i=0}^{i < M} \sum_{j=0}^{j < N} [w(x+i, y+j) - s(u+i, v+j)]^2 \right]^{1/2} \quad (1)$$

Pour s'adapter aux variations de niveaux de gris, de bout en bout de la fissure, d'autres caractéristiques statistiques sont calculées et intégrées à la fonction de coût. La position suivante de la fenêtre d'observation sera donnée par la valeur minimale de $C(u, v)$, définie par :

$$C(u, v) = rms + \left| \bar{w}(x, y) - \bar{s}(u, v) \right| + \left| \bar{w}(x, y) - \bar{t} \right| \rightarrow \text{Min} \quad (2)$$

où

- les variables $\bar{w}(x, y)$ et $\bar{s}(u, v)$ représentent respectivement les valeurs moyennes de l'intensité de $w(x, y)$ et de $s(u, v)$;
- la variable \bar{t} correspond à la valeur moyenne des valeurs de pixels déjà retenue comme faisant partie de la fissure.

L'emploi de blocs de taille 20×10 pour $w(x, y)$ et (u, v) donne de bons résultats pour l'étendue du champ de vision considéré. L'espace de recherche H a pour taille initiale $3M \times 3N$. Une rotation de 90 degrés de $w(x, y)$ permet de prendre en compte les changements de direction de la fissure. Utiliser un plus grand nombre d'angles de rotation aurait permis d'augmenter la précision sur la localisation mais aurait, dans le même temps, considérablement accru les temps de traitement.

Intuitivement, on conçoit aisément que, pris comme tel, cet algorithme a peu de chance de converger. Il existe une forte probabilité pour que celui-ci *boucle* sur des solutions déjà explorées.

Convergence du suivi

Deux mécanismes algorithmiques concurrents sont combinés pour garantir la convergence du processus de suivi. À la fonction de coût s'ajoute une minimisation de la distance entre la position candidate courante et l'extrémité opposée de la fissure. De plus, la gestion d'un historique des déplacements précédents limite l'espace de recherche à des zones non récemment explorées. Cette « *mémoire à court terme* » permet de ne pas revenir trop rapidement vers l'ensemble de solutions précédent. Ce mécanisme doit, tout en évitant un retour en arrière trop rapide, autoriser le suivi de fissures aux caractéristiques morphologiques complexes. La gestion dynamique de cette liste – communément appelée FIFO (*First In First Out*) –, et dont la taille spécifie la profondeur de la mémoire, permet de conserver un historique sous forme codée des derniers déplacements opérés.

Codage des directions

Le code de Freeman est une manière simple de représenter les déplacements dans l'espace de recherche. Habituellement dédié à la description du contour d'un objet, ce code permet ici de noter le déplacement entre deux points (contigus ou non). La direction du déplacement est une suite de nombres compris entre 0 et 7. La valeur 0 code un déplacement vers l'est, le 1 pour le nord-est, le 2 pour le nord, et ainsi de suite pour tous les huit déplacements réalisables en *8-connexité*. Pour une direction considérée m , le mouvement inverse n se calcule simplement en ajoutant 4 au code Freeman : $n = (m + 4) \text{ modulo } 8$ (l'emploi du modulo permet de rester dans l'étendue $[0-7]$).

En raison de l'épaisseur de la fissure – plusieurs pixels – un faisceau de directions non autorisées est calculé. Par exemple, si une direction précédente est codée 0 (*i.e.* d'ouest vers l'est), les directions 2, 3, 4, 5 et 6 ne seront pas explorées.

Gestion de l'historique

La longueur de la liste conditionne l'aptitude de l'algorithme à suivre des fissures aux morphologies complexes (par exemple, décrivant une boucle). Le choix de la taille de la liste doit être un compromis entre interdire des retours en arrière trop rapides et autoriser un retour vers des directions opposées aux déplacements localement estimés.

Par ailleurs, pour éviter les blocages – cas où toutes les directions sont interdites –, une politique de « *drainage anticipé* » des plus anciens éléments de la liste est mise en œuvre.

Une liste de douze éléments donne expérimentalement de bons résultats. La figure 2 montre un suivi pour une fissure longitudinale sur une chaussée en béton. On remarquera que le suivi n'a été perturbé ni par la médiocre qualité de l'image, ni par la présence de la bande de peinture blanche.

Fig. 2 - Suivi de fissure simple.



a. Image typique de fissure longitudinale sur une chaussée en béton.



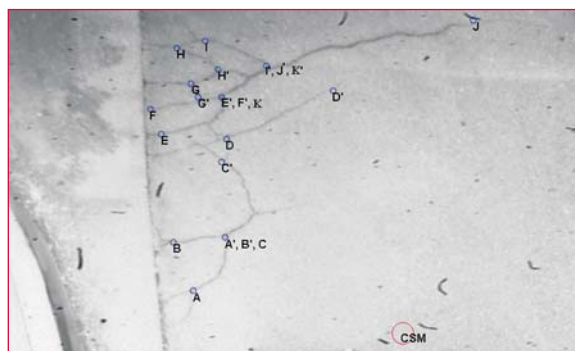
b. Extraction de la fissure ; le rectangle rouge à l'extrémité de la fissure représente la position initiale de la fenêtre d'observation.

Trajectoire optimale

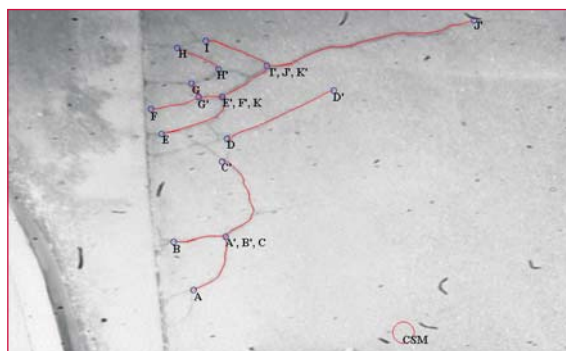
Une fois les fissures extraites de l'image, l'étape suivante consiste à calculer la trajectoire permettant au bras de boucher l'ensemble des fissures puis de revenir à une position de repli. Cette position est en général la position initiale du bras de robot. Cette opération revient à calculer le chemin reliant la position initiale du bras de robot à une extrémité de fissure (ou segment de fissure), puis de la seconde extrémité à une autre fissure, et ainsi de suite jusqu'à la dernière fissure puis enfin, de l'extrémité de cette dernière jusqu'à la position de repli du bras. La trajectoire optimale revient à minimiser la distance permettant de joindre chaque segment de fissure plus la distance à parcourir pour ramener le bras en position initiale.

En présence d'une scène avec une seule fissure, voire trois ou quatre, il est visuellement aisé pour l'opérateur de déterminer un plus court chemin. Pour des scènes plus complexes (fig. 3), typiquement quand la fissuration se traduit par du faïençage, une assistance logicielle est utile.

Fig. 3 - Suivi d'une fissure à morphologie complexe



a. Les dix segments de fissures à réparer ont été sélectionnés par l'opérateur. Ils sont notés AA', BB', ..., JJ'. Le rond rouge symbolise la position de repli du bras.



b. Extraction des segments de fissures sélectionnées.

Trouver la trajectoire optimale (au sens du plus court chemin à parcourir) peut être vu comme une déclinaison du « *Problème du voyageur de commerce* » (Dantzig et al., 1954 ; Glover et Punnam, 1997). Ce problème appartient à la large classe des problèmes **NP-Comple**t (*Nondeterministic Polynomial time complete problems*). En d'autres termes, le nombre de solutions réalisables fournies par l'analyse combinatoire croît selon l'expression $(N-1)!$, où N caractérise la taille du problème. Donner une approximation de la solution en utilisant des algorithmes tels que des *Algorithmes Gloutons* (AG) ou de *descente suivant la plus forte pente* peut conduire à des solutions dégradées (i.e. éloignées de l'optimum global). Rappelons qu'un AG est une méthode itérative pour laquelle, à chaque itération, on fixe la valeur d'une ou plusieurs variables sans jamais remettre en cause les choix antérieurs. En itérant ce procédé, on construit une suite de configurations de coûts décroissants. Cet algorithme converge rapidement vers une solution. Cette démarche naïve permet de faire converger la recherche vers le premier optimum local. Celui-ci a peu de chance d'être l'optimum global. Concrètement, pour la problématique qui nous intéresse, la solution calculée conduit souvent à une distance importante entre la fin de la dernière fissure et la position de repli. Pour cette raison, et aussi au vu des premiers résultats expérimentaux, nous nous sommes orientés vers l'utilisation d'une « méta-heuristique » de type algorithme de Recuit Simulé (RS). Fondée sur une analogie avec la thermodynamique, cette méthode, initiée par Metropolis et al. (1953), offre un cadre général pour résoudre des problèmes d'optimisation combinatoire. Soulignons que cette méthode d'optimisation ne garantit pas pour autant l'obtention de solutions exactes. Une démarche visant à introduire des techniques plus sophistiquées, telles que la relaxation Lagrangienne (Hiriart-Urruty et al., 1993 ; El Ghaoui et al., 1997) qui renseigne sur l'écart à l'optimum, ou plus rapides comme la méthode Tabou (Glover, 1989), ne se justifie pas du point de vue des contraintes de l'application.

Codage de la trajectoire et mouvements

Afin de pouvoir permuter rapidement l'ordre de réparation des segments de fissures, la trajectoire est codée par une simple séquence de nombres avec deux nombres consécutifs par segment. Chaque nombre se réfère aux coordonnées absolues d'une extrémité de fissure. Par exemple, les dix segments isolés dans la figure 2 peuvent être codés par la séquence [0 ; 1, 2 ; 3, 4 ; 5, 6, 7 ; 8, 9 ; ... ; 19, 20 ; 0]. La valeur 0 correspond à la position de repli. À cette configuration particulière correspond une longueur de trajectoire (*i.e.* la somme des distances entre chaque segment, plus les distances à parcourir pour rejoindre les positions de repli). Trouver la trajectoire optimale revient à calculer une configuration de la séquence qui minimise cette distance.

Le changement de configuration s'effectue par des permutations (ou *mouvements*) de l'ordre des nombres de cette séquence. Deux types de mouvements, effectués aléatoirement, sont considérés :

- i) les permutations du sens de réparation des fissures : dans notre exemple, changer l'ordre du premier segment donne la nouvelle séquence [0 ; 2, 1 ; 3, 4 ; 5, 6, 7 ; 8, 9 ; ... ; 19, 20 ; 0] ;
- ii) les permutations dans l'ordre de réparation des fissures : permuter la deuxième et la dernière fissure conduit à la configuration [0 ; 2, 1 ; 19, 20 ; 5, 6, 7 ; 8, 9 ; ... ; 3, 4 ; 0].

Recuit Simulé

L'algorithme de Recuit Simulé est une méthode de résolution de problèmes d'optimisation (Kirkpatrick et *al.*, 1983 ; Aart and Korst, 1989). Il se résume en un parcours pseudo-aléatoire de l'espace des variables (vecteur d'états) d'une fonction f à minimiser. Cette méthode est dérivée de la physique statistique. Elle est fondée sur une analogie thermodynamique, qui consiste à chauffer un matériau à haute température et à le laisser refroidir lentement, par paliers, afin de permettre au système d'atteindre son énergie libre minimale. L'idée est de minimiser une « énergie » liée à une « température ». Contrairement aux méthodes classiques d'optimisation qui n'acceptent que des diminutions de l'énergie, et n'autorisent ainsi une convergence que vers un minimum local, le Recuit Simulé accepte d'autres transformations, mais avec une probabilité qui décroît avec la température. À chaque itération de l'algorithme, le nouveau vecteur d'état x_{n+1} est calculé à partir de l'ancien x_n par ajout d'un bruit d'écart type T . Le vecteur x_{n+1} est retenu si $f(x_{n+1}) < f(x_n)$; dans le cas contraire, il est retenu avec une probabilité $P_a = \exp((f(x_{n+1}) - f(x_n))/T)$. Autrement dit, à température élevée, beaucoup de transformations sont acceptées, mais plus la température décroît, et moins les transformations qui augmentent l'énergie sont retenues. Les transformations qui accroissent l'énergie sont acceptées, même si elles dégradent momentanément la solution, pour permettre de sortir d'un minimum local. Cette méthode de résolution, sans garantir de solution exacte, fournit une solution proche de l'optimum sans avoir été piégé dans un minimum local.

Éléments génériques du Recuit Simulé

Si la solution initiale est générée aléatoirement, quatre paramètres sont nécessaires au Recuit Simulé :

- la température initiale,
- le mode de décroissance,
- le nombre d'itérations par palier,
- le test de convergence.

■ **Température initiale.** Le choix de la température initiale conditionne en partie le déroulement optimal de la recherche. Une température de départ trop faible interdit d'entrée certaines remontées énergétiques $\exp(-\Delta f/T)$ qui permettraient de sortir d'un minimum local. *A contrario*, une température initiale trop élevée pénalise le temps de convergence en acceptant trop de solutions sans garantir la décroissance énergétique de la fonction de coût. Dans notre implantation, nous avons utilisé une méthode classique couramment adoptée lors du choix de la température initiale. En pratique, cela revient à calibrer la température pour qu'après une centaine d'itérations le taux de transformation T soit défini par $T = \text{transformations acceptées} / \text{transformations proposées} \approx 50\%$.

■ **Palier et décroissance de la température.** L'algorithme de Recuit Simulé prévoit d'effectuer un certain nombre d'itérations à température constante, puis d'abaisser cette température par paliers successifs. Le protocole de refroidissement est une suite finie de températures, décroissantes par palier. Au palier m ; $0 < m < M$, on associe une température T_m et une longueur de palier L_m . On suppose que $T_0 > T_1 > \dots > T_M$ et que $L_0 > 0$. L'abaissement de la température est effectué soit par

une décroissance logarithmique, soit par une décroissance géométrique. Nous avons adopté un processus de décroissance géométrique de la température de la forme : $T_{(k+1)} = \lambda \times T_k$; $0,8 < \lambda < 0,9$. La taille des paliers isothermes peut être constante ou variable. L'augmentation du nombre d'itérations par palier conduit à tester plus de transformations à basse température pour éventuellement sortir d'un minimum local. Afin de privilégier le temps d'exécution de l'algorithme par rapport à la qualité de la solution, nous avons implanté un algorithme de recuit fondé sur une progression à taille fixe des paliers. Cette taille se détermine expérimentalement.

■ **Convergence.** On décide que la convergence est atteinte lorsque la solution reste identique pendant un nombre d'itérations suffisamment élevé. Le Recuit Simulé possède la propriété de convergence stochastique vers l'optimum global sous condition de décroissance infiniment longue, avec des pas suffisamment fins de la température. En pratique, on cherche un compromis entre ces deux paramètres pour se rapprocher des conditions de convergence théorique, en choisissant des pas de décroissance faibles et des paliers isothermes suffisamment longs. Dans le cas qui nous intéresse, le nombre maximal d'itérations est proportionnel au nombre de segments de fissures à boucher.

Résultats expérimentaux

Les résultats obtenus en condition opérationnelle de fonctionnement de l'engin sont encourageants. Les figures 4, 5 et 6 soulignent deux aspects importants des approches retenues, à savoir une robustesse dans le suivi et l'intérêt d'utiliser un algorithme d'optimisation.

La figure 4 souligne la capacité de l'algorithme à s'adapter aux conditions de prise de vue. Malgré la sur ou sous-exposition des différentes images, le processus de suivi de la fissure a pu être réalisé.

La figure 5 montre la détection d'une fissure complexe (en rouge) et la trajectoire (en bleu) que doit suivre le bras de robot pour effectuer la réparation. La position de repli est représentée par un rond rouge. On remarquera qu'en raison de la mauvaise qualité de l'image une portion de fissure (entre B' et C) n'a pas pu être extraite. Cette portion devra être suivie manuellement par l'opérateur.

La figure 6 met en évidence l'intérêt d'utiliser un algorithme d'optimisation stochastique pour trouver la trajectoire minimale. La position de repli du bras de robot de la CSM est symbolisée par un rond rouge. La première solution (fig. 6a) est générée par un algorithme *Glouton*. La figure 6b est obtenue après convergence d'un RS. Dans ce cas précis, le gain, en terme de réduction de la longueur de trajectoire, procuré par l'algorithme de RS est supérieur à 15 %. Pour des configurations différentes, cet écart peut être supérieur.

Fig. 4 - Robustesse aux conditions de prise de vue.



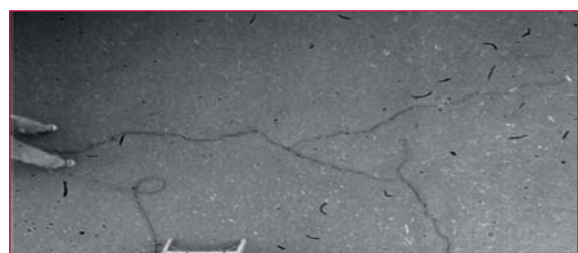
a. Image initiale sous-exposée.



b. Extraction de la fissure.



c. Image initiale sur-exposée.

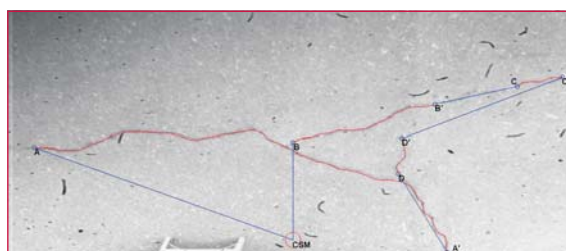


d. Extraction du segment de fissure sélectionné.

Fig. 5 - Planification de trajectoire du robot.

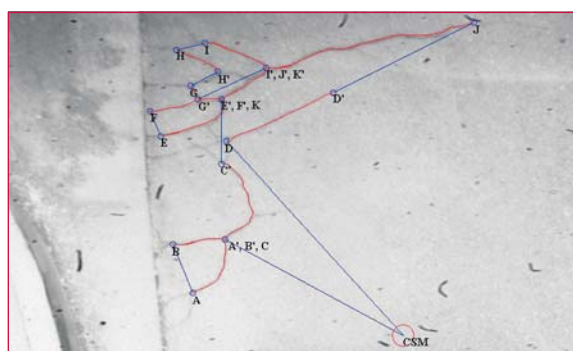


a. Image initiale, quatre segments sont sélectionnés par l'opérateur : AA', BB', CC' et DD'.

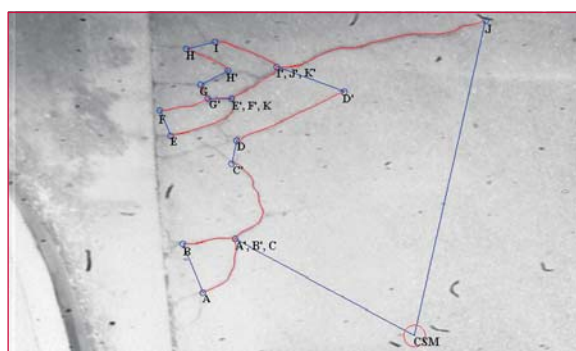


b. La ligne bleue détermine la trajectoire que doit décrire le bras pour boucher ces segments à partir de la position de repli (CSM). La trajectoire optimale est CSM, AA', DD', C'C, B'B, CSM.

Fig. 6 - Comparaison des performances respectives des algorithmes de Recuit Simulé et Glouton.



a. Solution sous-optimale générée par un algorithme de recherche Glouton.



b. Trajectoire optimale utilisant un algorithme de Recuit Simulé.

Conclusion

Le principal objectif de ce travail était d'accroître l'ergonomie du module de commande de l'engin de travaux publics OCCSM. Associée à une phase manuelle, la planification de la trajectoire du bras a nécessité de combiner des algorithmes de vision et une méthode classique d'optimisation combinatoire. L'approche proposée procure un confort d'emploi apprécié des utilisateurs *in fine*.

Du point de vue traitement d'images, la démarche algorithmique, malgré sa simplicité, s'est révélée très robuste. Cette robustesse a pu être expérimentée pour des conditions extrêmes d'utilisation. Les évolutions futures porteront essentiellement sur le processus d'acquisition de l'image.

La morphologie complexe des fissures impose d'avoir recours à des outils d'optimisation pour estimer une trajectoire minimale. Même s'il s'agit d'un problème difficile, au sens algorithmique du terme, sa taille permet de le traiter avec un temps de calcul raisonnable. Une fois les extrémités de fissures pointées, moins d'une seconde de calcul est nécessaire pour proposer à l'opérateur un parcours de réparation. La taille du problème est conditionnée par le nombre de segments à traiter : tout au plus quelques dizaines. Au-delà, le gestionnaire de l'ouvrage devra probablement envisager un type de réparation plus adéquat.

Remerciements. Les auteurs remercient :

- la Délégation générale pour l'armement (DGA), Direction des systèmes de forces et de la prospective. En effet, ces travaux, entrepris au cours d'un séjour d'étude à l'université de Davis en Californie, ont été menés en marge du contrat de recherche ERE N°9960054 financé par cet organisme ;
- l'Office of New Technology and Research of the California Department of Transportation, pour son soutien au travers du Advanced Highway Maintenance and Construction Technology Center.

- AART E., KORST J., *Simulated Annealing and Boltzmann Machine*, John Wiley and Sons, **1989**.
- CHENG H.D., CHEN J.R., GLAZIER C., HU Y.G., Novel Approach to Pavement Cracking Detection Based on Fuzzy Set Theory, *Journal of Computing in Civil Engineering, Transactions ASCE*, Vol. 13, **4**, **1999**, pp. 270-280.
- EI GHAOUI L., LEBRET H., ROTTEMBOURG B., Optimisation convexe pour les sciences de l'ingénieur, *Revue scientifique et technique de la Défense*, Vol. 35, septembre **1997**.
- DANTZIG G., FULKERSON D., JOHNSON S., *Solution of a large-scale traveling-salesman*, Operational Research Society, Vol. 2, **1954**, pp. 393-410.
- FENG X., BAKER T.L., VELINSKY S.A., Long Reach Robotic Manipulator for Advanced Highway Maintenance and Construction Equipment, ROMANSY 12 – Theory and Practice of Robots and Manipulators, *Proc. of the Twelfth CISM-IFTOMM Symposium on Theory and Practice of Robots and Manipulators*, **1998**, pp. 309-316.
- FENG X., BENNETT D.A., VELINSKY S.A., Embedded PC Control System for a Highway Crack Sealing Machine, *Proc. of the ASME Design Technical Conference*, **1998**, Paper #DETC98/CIE-5531.
- GLOVER F., PUNNEN A.P., The Traveling Salesman Problem : New Solvable Cases and Linkages with the Development of Approximation Algorithms, *Journal of the Operational Research Society*, Vol. 48, **1997**, pp. 502-510.
- GLOVER F., Tabu Search, Part 1, *ORSA Journal of Computing*, Vol. 1, **3**, **1989**.
- HIRIART-URRUTY J.-B., LEMARÉCHAL C., *Convex Analysis and Minimization Algorithms*, Springer-Verlag, **1993**.
- KIM Y., HAAS C., GREER R., Path Planning for a Machine Vision Assisted, Tele-operated Pavement Crack Sealer, *Journal of Transportation Engineering, Transactions ASCE*, Vol. 124, **2**, **1998**, pp. 137-143.
- KIRSCHKE K.R., VELINSKY S.A., Histogram-Based Approach For Automated Pavement Crack – Sensing, *Journal of Transportation Engineering, Transactions ASCE*, Vol. 118, **5**, **1992**, pp. 700-710.
- KIRKPATRICK S., GELATT Jr C.D., VECCHI M.P., Optimization by Simulated Annealing, *Science*, Vol. 220, **4598**, **1983**, pp. 671-680.
- METROPOLIS N., ROSENBLUTH A.W., ROSENBLUTH M.N., TELLER A.H., TELLER E., Equation of State Calculations by Fast Computing Machines, *J. Chem. Phys.*, Vol. 21, **6**, **1953**, pp. 1087-1092.
- MATHURIN R., GUÉDON-DUBIED J.-S., MARTINEAU F., Quantification des gels d'alcali-réaction par traitement numérique d'images, *Journées des sciences de l'ingénieur du réseau des laboratoires des Ponts et Chaussées*, Presqu'île de Giens, 4-7 octobre **1994**.
- PYNN J., WRIGHT A., LODGE R., Automatic identification of cracks in road surfaces, *Seventh International Conference on Image Processing and Its Applications*, Vol. 2, p. 671, Manchester, UK, 13-15 July **1999**.
- RADU STEFAN Stoica, *Processus ponctuels pour l'extraction de réseaux linéiques dans les images satellitaires et aériennes*, Thèse de l'université de Nice-Sophia Antipolis, 9 février **2001**, 105 pages.
- RELLIER G., DESCOMBES X., ZERUBIA J., *Deformation of a Cartographic Road Network on a SPOT Sattelite Image*, ICIP, Vancouver, Canada, septembre **2000**.
- VELINSKY S.A., Heavy Vehicle System for Automated Pavement Crack Sealing, *Heavy Vehicle Systems*, Vol. 1, **1**, **1993**, pp. 114-128.