Comparaison de quelques algorithmes de chemin minimal pour établir la pseudo-vérité terrain des fissures sur des images 2D de chaussée

L. Yang[1] V. Baltazart[2]

[1]Xi'an Jiaotong University, Shaanxi, China
[2] IFSTTAR, LUNAM Université, CS4, 44340 Bouguenais, France

vincent.baltazart@ifsttar.fr

Résumé

L'évaluation des performances d'algorithmes automatique de détection de fissures nécessite au préalable d'établir une image de référence, ou encore pseudo-vérité terrain (PVT). Dans ce contexte, cette communication illustre le fonctionnement de 5 algorithmes de chemin minimal, pour établir la PVT de manière semi-automatique.

Mots Clef

chemins minimaux, fissures, pseudo-vérité terrain.

Abstract

The performance assessment of automatic crack detection algorithms requires a proper pseudo-ground truth data collection. The latter is usually performed by some semi-automatic process thanks to the use of single pair shortest path algorithms. This communication illustrates the performance of five different algorithms devoted to this aim in terms of running time and segmentation accuracy.

Keywords

Single-pair shortest path, cracking, pseudo-ground truth

1 Introduction

Des méthodes de segmentation automatique d'images de chaussée ont été développées pour les besoins de suivi de la qualité du réseau routier [1,2]. Parmi les défauts de surface, on s'intéresse en particulier à la détection de fissures.

L'évaluation des performances des algorithmes est basée sur le calcul d'un coefficient de similitude entre le résultat de la segmentation automatique de fissure et une image de référence, appelée pseudo-vérité terrain (PVT). Cette dernière est une image binaire représentant l'ensemble des fissures à détecter sur l'image, avec leur épaisseur associée.

Un opérateur établit la PVT en partie manuellement. Dans ce contexte, tout outil de traitement d'image est susceptible de faciliter le travail fastidieux de l'opérateur.

Parmi les algorithmes disponibles, cette communication se focalise sur les méthodes basées sur la recherche de plus courts chemins. Les méthodes de percolation [3] et LBP (Local Binary Pattern [4]) n'ont pas donné satisfaction sur les images fortement texturées de chaussées; en outre, elles sont plus lentes que les méthodes à base de chemins minimaux.

2 Pseudo-Vérité Terrain (PVT)

Les premiers travaux d'évaluation utilisaient en référence une segmentation manuelle des fissures, établies à partir du relevé de quelques opérateurs [2]. L'union des résultats formait la base de l'image de PVT. Cette méthode s'est révélée moins précise que la PVT semi-automatique, utilisée maintenant [1].

Dans la méthode semi-automatique, l'opérateur sélectionne à l'écran un point source et un point destination appartenant à la fissure. Un algorithme de plus court chemin détermine automatiquement le squelette de cette portion de fissure. La fissure est décrite ainsi par portions successives selon la complexité de sa géométrie. Une dernière étape consiste à estimer l'épaisseur le long du squelette de la fissure, en agrégeant itérativement au squelette les pixels d'un niveau de gris faible [1].

Différents algorithmes de chemins minimaux peuvent satisfaire le besoin de l'application .

3 Algorithmes testés

Deux familles d'algorithmes de plus courts chemins ont été testées. L'algorithme de Dijkstra [5] peut être utilisé indifféremment pour la problématique one-to-one (single pair shortest path) ou one-to-all (single source shortest path). Dijsktra est réputé être le plus rapide des algorithmes SSSP. Il minimise la distance de Manhattan (norme L1) entre deux pixels de l'image. En comparaison, les autres algorithmes testés sont dédiés à la problématique SPSP uniquement.

Dans la version bidirectionnelle de Dijkstra [5], la recherche du plus court chemin est initialisée simultanément aux deux point amorces, et la recherche progresse jusqu'à trouver un point de jonction. Le nombre de points visités se réduit de moitié environ, ainsi que le temps d'exécution.

Les algorithmes de la famille *(star) ont été proposés en alternative à Dijkstra. A* est le plus rapide et le plus précis des algorithmes de cette famille. A* utilise une fonction heuristique, qui oriente la recherche du chemin le plus court entre le pixel courant et le pixel destination. Il a fait l'objet d'une adaptation dans [6], pour les besoins de l'application.

En comparaison, la méthode de Fast Marching [7] est un algorithme, qui minimise la distance géodésique entre deux pixels (norme L2). Elle est très utilisée en traitement d'image pour la reconnaissance de formes (contours actifs).

Une variante récente [8], que nous avons appelé Self Terminating Fast Marching (STFM) nécessite d'initialiser la recherche en un seul point, l'autre extrémité de la fissure étant automatiquement

déterminée. Dans une certaine mesure, la recherche du chemin peut s'étendre aux différentes ramifications d'une fissure. Ceci facilite le travail de l'opérateur pour déterminer la PVT. La contrepartie est un temps d'exécution plus important.

4. Illustration

Les algorithmes ont été programmés sous Matlab et testés sur des images simulés (dont on connait la vérité terrain), ainsi que sur des images de chaussée, dont on a déterminé la PVT au préalable. Les méthodes se classent de manière identique sur les deux types d'images. On a choisi d'illustrer le fonctionnement des 5 algorithmes testés sur une image réelle de 462×690 pixels.

Les résultats sont classés de l'algorithme le plus rapide ou plus lent. Pour donner un ordre de grandeur, la méthode A* prend environ 4 sec. sur l'image de test (HP Zbook14 Intel Core i7 @ 2.7 GHz). La méthode STFM ausculte le plus faible nombre de pixels mais prend environ 60 sec. du fait d'une complexité calculatoire plus importante.

Une comparaison qualitative a montré que les algorithmes de norme L1 (Dijkstra, A*) permettaient de mieux suivre la trajectoire chaotique des fissures, alors que les techniques L2 (FM, STFM) avaient tendance à lisser les variations de direction de la fissure, réduisant ainsi la valeur du coefficient de similarité (DICE) de quelques pourcent, comme indiqué en dernière colonne de la Fig. 1.

Bibliographie

[1] R. Amhaz, S. Chambon, J. Idier, V. Baltazart, Automatic crack detection on 2D pavement images : An algorithm based on minimal path selection, accepté à IEEE TITS, août 2015

[2] Chambon S. and J.-M. Moliard, Automatic road pavement assessment with image processing: Review and comparison, Int. J. Geophysics, 2011.

[3] Yamaguchi T., S. Nakamura and S. Hashimoto, An Efficient Crack Detection Method Using Percolation-Based Image Processing, ICIEA 2008.

[4] Hu Y. and Zhao C., "A Novel LBP Based Methods for Pavement Crack Detection," Journal of Pattern Recognition Research, Vo. 5, no. 1, 2010

[5] Wagner D., T. Willhalm, Speed-Up Techniques for Shortest-Path Computations, , STACS 2007, LNCS 4393, pp. 23–36, Springer-Verlag, 2007

[6] L. Yang, V. Baltazart, R. Amhaz, P. Jiang, A new A-star algorithm adapted to the semi-automatic detection of cracks within grey level pavement images, 8th ICDIP, Chengdu, China, May 2016

[7] L.D. Cohen and Ron Kimmel, "Global minimum for active contour models: A minimal path approach," International Journal of Computer Vision, vol. 24, no. 1, pp. 57–78, (1996)

[8] Kaul V., A. Yezzi, and Y. Tsai, "Detecting curves with unknown endpoints and arbitrary topology using minimal paths," in IEEE Trans. PAMI, 2012, vol. 34, pp. 1952–1965.

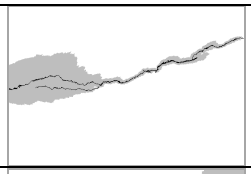
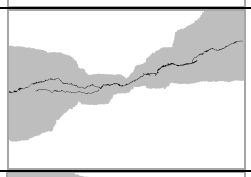
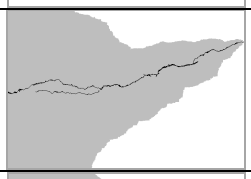
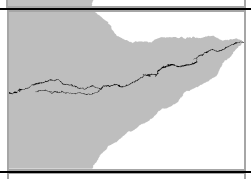
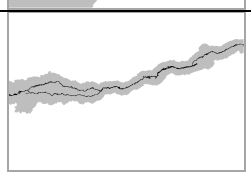
Algorithme	Zone visitée	Nombre de pixels visités	Temps d'éxecution relatif	Critère de similitude DICE
A* [6]		38361 (12.0 %)	1	0.92
Dijkstra bidirectionnel [5]		109609 (34.4 %)	× 1.6	0.94
Dijkstra [5]		195783 (61.4 %)	× 4	0.94
Fast Marching [7]		203108 (63.7 %)	× 6.6	0.86
Self-Terminating Fast-Marching [8]		39773 (12.5 %)	× 14.8	0.87

Figure 1. Illustration du fonctionnement des 5 algorithmes SPSP pour établir la PVT de manière semi-automatique