HAL
open science

RANSAC algorithm and elements of graph theory for automatic plane detection in 3D point clouds

Martyna Poreba, François Goulette

To cite this version:

Martyna Poreba, François Goulette. RANSAC algorithm and elements of graph theory for automatic plane detection in 3D point clouds. Symposium de PTFiT (Polish Society for Photogrammetry and Remote Sensing), Sep 2012, Poland. pp.301-310. hal-00846335

HAL Id: hal-00846335

https://minesparis-psl.hal.science/hal-00846335v1

Submitted on 19 Jul 2013

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.

Archiwum Fotogrametrii, Kartografii i Teledetekcji Vol. 24, 2012, s. 301 - 310
ISBN 978-83-61576-22-8

AUTOMATYCZNA DETEKCJA PŁASZCZYZN W CHMURZE PUNKTÓW W OPARCIU O ALGORYTM RANSAC I ELEMENTY TEORII GRAFÓW

RANSAC ALGORITHM AND ELEMENTS OF GRAPH THORY FOR AUTOMATIC PLANE DETECTION IN 3D POINT CLOUD

Martyna Poręba [1,2], François Goulette [2]

[1] Akademia Górniczo – Hutnicza w Krakowie, Katedra Geoinformacji, Fotogrametrii i Teledetekcji Środowiska
[2] MINES ParisTech, CAOR Centre de Robotique, Mathématiques et Systèmes

SŁOWA KLUCZOWE: chmura punktów, segmentacja, RANSAC, graf, algorytm najbliższego sąsiada, etykietowanie, spójny komponent

STRESZCZENIE: Artykuł przedstawia metodę automatycznego wyodrębniania punktów modelujących płaszczyzny w chmurach punktów pochodzących z mobilnego bądź statycznego skaningu laserowego. Zaproponowany algorytm bazuje na odpornym estymatorze RANSAC umożliwiającym iteracyjną detekcję płaszczyzn w zbiorze cechującym się znacznym poziomem szumu pomiarowego i ilością punktów odstających. Aby zoptymalizować jego działanie, dla każdej wykrytej płaszczyzny uwzględniono relacje sąsiedztwa pomiędzy punktami przynależnymi. W tym celu zastosowano podejście oparte na teorii grafów, gdzie chmura punktów traktowana jest jako graf nieskierowany, dla którego poszukiwane są spójne składowe. Wprowadzona modyfikacja obejmuje dwa dodatkowe etapy: ustalenie najbliższych sąsiadów dla każdego punktu wykrytej płaszczyzny wraz z konstrukcją listy sąsiedztwa oraz etykietowanie spójnych komponentów. Rezultaty uzyskane pokazują iż algorytm poprawnie wykrywa płaszczyzny modelujące, przy czym niezbędny jest odpowiedni dobór parametrów początkowych. Czas przetwarzania uzależniony jest przede wszystkim od liczby punktów w chmurze. Nadal jednak aktualny pozostaje problem wrażliwości algorytmu RANSAC na niską gęstość chmury oraz nierównomierne rozmieszczenie punktów.

1. WPROWADZENIE

Rekonstrukcja trójwymiarowych modeli budynków odgrywa w ostatnich latach znaczącą rolę w dziedzinie fotogrametrii i widzenia komputerowego. Rozwój technologii skaningu laserowego w aspekcie szybkości pomiaru, precyzji punktów oraz rozdzielczości otworzył nowe horyzonty badań, zawłaszcza w temacie modelowania zabudowy na podstawie nieuporządkowanej chmury punktów. Istotnym etapem przetwarzania i analizy danych jest segmentacja mająca na celu wyodrębnienie fragmentów o tym samym znaczeniu semantycznym. Jako kryterium jednorodności przyjmuje się na ogół położenie punktów, krzywiznę czy kierunek wektora normalnego. W świecie realnym budynki w znacznej mierze formowane są jako kombinacja płaszczyzn. Segmentacja automatyczna

ma na celu rozkład chmury punktów na podzbiory punktów reprezentujące poszczególne płaszczyzny. Istniejące metody segmentacji chmur punktów stanowią rozwinięcie dobrze znanych algorytmów, dedykowanych pierwotnie cyfrowemu przetwarzaniu obrazów. Zapewniają one minimalizację strat informacji przestrzennej występujących w przypadku konieczności konwersji danych 3D do postaci 2D. Krótki przegląd metod i algorytmów możliwych do zastosowania przedstawiony przez (Vosselman *et al.,* 2004) i (Boulaassal, 2010) pozwala wyróżnić dwie kategorie metod segmentacji. Pierwsza rodzina obejmuje algorytmy działające na zasadzie łączenia, takie jak metoda rosnących płaszczyzn (and. *surface growing*) czy algorytm dziel-łącz (ang. *split - merge*). Drugą grupę stanowią techniki umożliwiające automatyczną rekonstrukcję form zdefiniowanych poprzez prymitywy geometryczne w postaci sfery, płaszczyzny, cylindra lub stożka. Zalicza się do nich transformatę Hougha (ang. *Hough Transformation*) i algorytm RANSAC (ang. *RANdom SAmple Consensus*).

Przeprowadzone badania literaturowe pokazują iż na etapie segmentacji danych dominują metody rosnących płaszczyzn czy też te oparte o algorytm RANSAC. W swoich pracach (Bauer *et al*., 2005) i (Boulaassal, 2010) wykorzystują algorytm RANSAC dla celów detekcji fasad budynków. Aby zoptymalizować działanie tego estymatora w procesie wyodrębniania płaszczyzn, został on poddany licznym modyfikacjom. (Awwad *et al.,* 2010) oraz (Delmerico *et al*., 2011) proponują uwzględnienie kierunku wektora normalnego obliczonego w każdym punkcie chmury. Algorytm (Yang *et al.,* 2010) polega na podziale chmury punktów na małe bloki, zawierające nie więcej niż trzy płaszczyzny. Następnie w każdym wokselu aproksymowane są płaszczyzny, których liczba weryfikowana jest poprzez zasadę MDL (ang. *Minimum Description Length*). Wykryte płaszczyzny łączone są ostatecznie ze sobą w oparciu o pewne parametry lokalne. Metody (Deschaud, 2010), (Tuttas *et al.,* 2010) bazują natomiast na algorytmie rosnących płaszczyzn polegającym kolejno na identyfikacji płaszczyzny pierwotnej, a następnie jej ekspansji, czyli grupowaniu punktów według pewnych ustalonych parametrów określających ich wzajemne podobieństwo. Istnieje ponadto wiele wariantów metody rosnących płaszczyzn, różniących się między sobą sposobem wyboru płaszczyzn początkowych jak również kryterium rozrostu. Liczne są również prace porównujące rezultaty segmentacji chmur punktów przeprowadzonych z użyciem wymienionych algorytmów. (Jarząbek-Rychard, *et al.,* 2010) konfrontuje wydajność estymatora RANSAC z metodą rosnących płaszczyzn. Przeprowadzone testy polegające na detekcji połaci dachowych w danych z lotniczego skaningu laserowego pozwalają stwierdzić, iż RANSAC jest rozwiązaniem lepszym w przypadku segmentacji modeli nieskomplikowanych geometrycznie. Z drugiej jednak strony jest on bardziej podatny na łączenie w jedną płaszczyznę matematyczną punktów przynależących do różnych, w rzeczywistości odrębnych obiektów. W podobnym kontekście realizuje swoje studium porównawcze (Tarsha-Kurdi, 2008) testując algorytm RANSAC i transformatę Hougha. Stwierdza on, że RANSAC jest bardziej przystosowany do przetwarzania danych przestrzennych.

W niniejszej pracy poruszono problematykę segmentacji chmur punktów – w pełni automatycznej detekcji płaszczyzn. Zaproponowana metoda bazuje na algorytmie RANSAC, którego działanie zostało zoptymalizowane poprzez uwzględnienie relacji sąsiedztwa pomiędzy punktami tworzącymi daną powierzchnię matematyczną tak, aby współpłaszczyznowym obiektom przypisać odrębne, ograniczone płaszczyzny modelujące.

2. ALGORYTM RANSAC

Algorytm RANSAC jest iteracyjną metodą stosowaną w celu estymacji parametrów poszukiwanego modelu matematycznego obiektu w zbiorze danych zawierającym znaczną ilość punktów nienależących do modelowanej powierzchni (Fischler *et al.,* 1981), (Zuliani, 2012). Fakt ten, czyni estymator RANSAC szczególnie interesującym dla celów przetwarzania chmur punktów obarczonych szumem i błędnymi pomiarami. Działanie algorytmu obejmuje zasadniczo dwie fazy: inicjalizację i test, obydwie powtarzane iteracyjnie.

Etap inicjalizacji polega na wyborze w sposób losowy minimalnego zbioru punktów (kworum), niezbędnego do jednoznacznego wyznaczenia parametrów estymowanego modelu geometrycznego. W przestrzeni euklidesowej płaszczyznę definiują cztery parametry $\Theta_{1\div4}$, z czego trzy są niezależne. Parametry hipotetycznego modelu M(Θ) wyznaczane są zatem w oparciu o trzy elementy losowej próbki. Następnie pozostałe punkty ze zbioru porównywane są z tak utworzonym modelem. W tej fazie niezbędne jest przyjęcie progu (δ) określającego zakres, w którym punkt (p) pasuje do modelu. Wszystkie punkty spełniające zadane kryterium (ang. *inlieres*) dodawane są do zbioru CS (ang. *Consensus Set*) (1), a cały etap określany jest mianem testu.

$$CS = \{p \in N : e_M(p, M(\theta)) \leq \delta\} \tag{1}$$

Różnica między punktem a modelem obliczana jest na podstawie równania (2):

$$e_M(p, M(\theta)) \overset{def}{=} \min dist(p, p^*) = \sqrt{\sum_{i=1}^{n} (p_i - p_i^*)^2} \tag{2}$$

gdzie p_i^* oznacza rzut ortogonalny punktu p na model M(Θ).

Dobór odpowiedniej wartości tolerancji (δ) - szerokości bufora wokół płaszczyzny średniej - odgrywa ważną rolę na poziomie stabilności algorytmu RANSAC i wpływa na jakość wyodrębnionych płaszczyzn. Jako że skanowane obiekty nie są utworzone z idealnych powierzchni, a chmura punktów posiada pewną miąższość spowodowaną szumem pomiarowym, wpasowanie modelu odbywa się z zachowaniem pewnej elastyczności. Najczęściej tolerancja (δ) definiowana jest jako maksymalna odległość testowanego punktu od teoretycznego modelu. Według (Hartley et al., 2003) wartość δ może zostać wyznaczona przy założeniu, że wszystkie dane obarczone są szumem gaussowskim, czyli błędem o rozkładzie normalnym $\eta \sim N(0, \sigma_\eta I)$ dla $\eta = p\text{-}p^*$. Zatem tolerancja (δ) określa z prawdopodobieństwem P czy dany punkt należy do strefy buforowej.

$$P(e_M(p, M(\theta) \leq \delta) = P[\sum_{i=1}^{n} \eta_i^2 \leq \delta^2] = P[\sum_{i=1}^{n} (\frac{\eta_i}{\sigma_\eta})^2 \leq \frac{\delta^2}{\sigma_\eta^2}] \tag{3}$$

Ponieważ $\frac{\eta_i}{\sigma_\eta} \sim N(0,1)$ zatem zmienna losowa $\sum_{i=1}^{n} (\frac{\eta_i}{\sigma_\eta})^2$ posiada rozkład chi-kwadrat o n stopniach swobody. Ostatecznie, rozwijając równanie (3), przy uwzględnieniu powyższych zależności, wartość tolerancji (δ) obliczana jest zgodnie z relacją (4):

$$\delta = \sigma_\eta \sqrt{F^{-1}_{\chi^2_2}(P)} \tag{4}$$

gdzie:
$F^{-1}_{\chi^2_2}$ - wartość krytyczna rozkładu chi-kwadrat z liczbą stopni swobody równą dwa na poziomie istotności P.

W ten sposób wyznaczenie wartości progowej (δ) sprowadza się do podania prawdopodobieństwa P oraz wartości szumu pomiarowego σ_η.

Proces inicjalizacji i testu powtarzany jest iteracyjnie dla nowo wylosowanej próbki zbiorów minimalnych, a liczba powtórzeń (Iter) wyznaczana jest zgodnie z równaniem (5) (Zuliani, 2012). W przypadku gdy liczebność punktów card(CS) w kolejnym znalezionym zbiorze CS jest większa od poprzedniego, zastępuje on dotychczasowy zbiór.

$$(1-P) = (1-W^3)^{Iter} = Iter = \frac{\log(1-P)}{\log(1-W^3)} \tag{5}$$

Powyższa zależność pokazuje, że liczba iteracji obliczana jest na postawie znajomości prawdopodobieństwa P identyfikacji „najlepszej płaszczyzny" oraz prawdopodobieństwa W, że wszystkie punkty losowej próbki należą do ostatecznego modelu. Dobór tych parametrów odbywa się w sposób empiryczny, stąd też prawdopodobieństwo P przyjmuje wartość z przedziału 0.95 ÷ 0.99, z kolei W powinno być oszacowane w funkcji danych (6).

$$W = \frac{card(CS)}{card(N)} \tag{6}$$

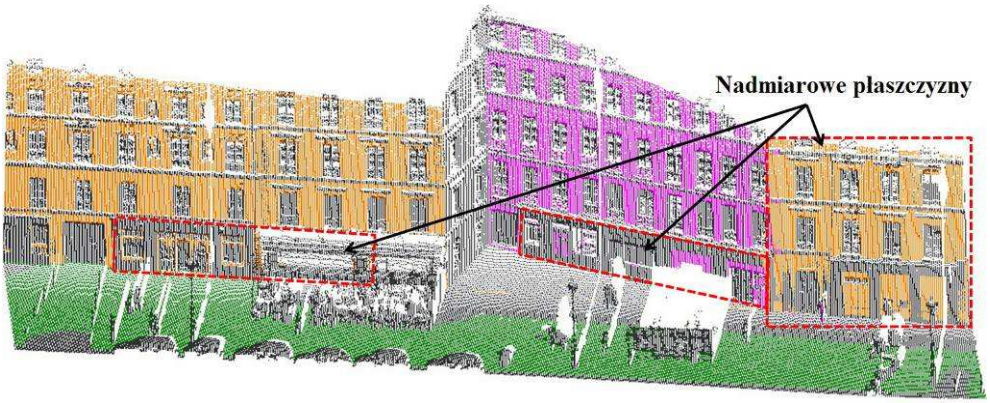
gdzie:
card(N) – liczba wszystkich punktów wejściowego zbioru danych.

Z praktycznego punktu widzenia, określenie parametru W może przysporzyć problemów dlatego też algorytm RANSAC w wersji bazowej inicjalizowany jest poprzez zgrubną estymacje tej wartości. Rozwiązanie zaproponowane przez (Hartley *et al.*, 2003) zmniejsza czas przetwarzania pozwalając na dynamiczny dobór maksymalnej liczby iteracji, aktualizowanej przed rozpoczęciem następnej sekwencji - inicjalizacji i testu.

3. PROPONOWANY ALGORYTM DETEKCJI PŁASZCZYZN

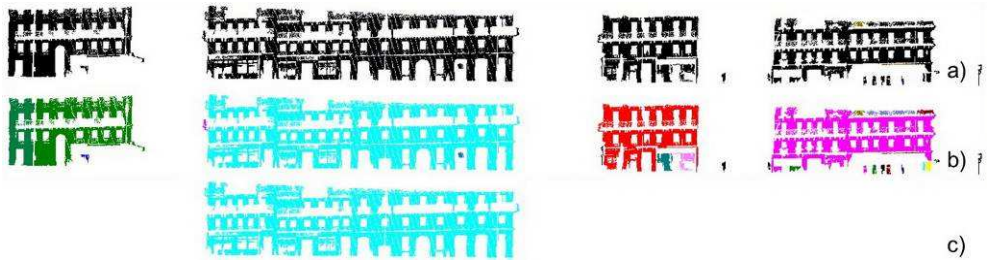
W niniejszej pracy algorytm RANSAC (Zuliani, 2012) z modyfikacjami wprowadzonymi przez (Hartley *et al.,* 2003) zaimplementowano sekwencyjnie, a więc punkty przynależące do każdej kolejno zidentyfikowanej płaszczyzny są wyłączane z dalszych poszukiwań. Algorytm kończy swoje działanie, gdy wykryta zostanie wstępnie założona liczba płaszczyzn (N_{pl}) lub liczba punktów nie pozwala już na poprawną estymację modelu. Rysunek 1 ilustruje rezultat ekstrakcji trzech kolejnych płaszczyzn w chmurze punktów (N). Cechą specyficzną algorytmu RANSAC jest fakt, iż generuje on dwa rodzaje błędów: powierzchniowe i liniowe, formowane przez punkty ułożone w sekwencje linii lub pól rozproszonych wokół powierzchni dominującej. Punkty te tworzą jedną płaszczyznę matematyczną lecz z punktu widzenia semantycznego nie są zgodne z rzeczywistością, gdyż należą do różnych obiektów. Podstawowym wyzwaniem

postawionym przed rozwijanym algorytmem, decydującym o jego operacyjności, jest zdolność detekcji dla każdego elementu architektonicznego odrębnej płaszczyzny modelującej.



Rys. 1. Rezultat detekcji trzech płaszczyzn w chmurze punktów za pomocą algorytmu RANSAC w wersji bazowej (dany kolor odpowiada jednej płaszczyźnie)

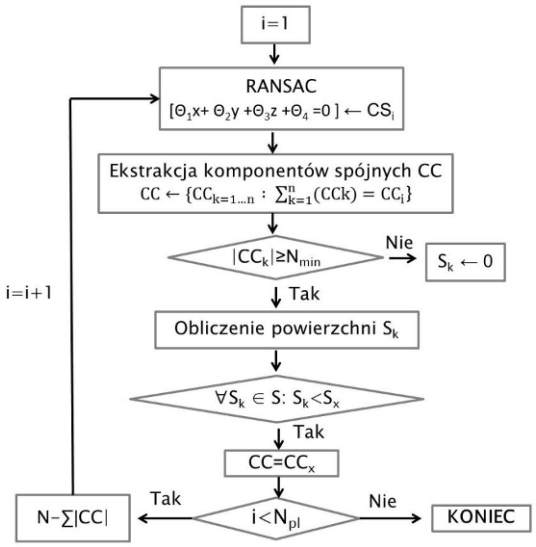
Aby zoptymalizować działanie algorytmu RANSAC wprowadzono dodatkowo ograniczenie dotyczące łączności pomiędzy punktami wyodrębnionymi dla poszczególnych powierzchni. W rezultacie dla każdego podzbioru reprezentującego pojedynczą płaszczyznę weryfikowane jest czy stanowi on jeden ciągły element, czy może zbiór rozłącznych powierzchni. Zaproponowane rozwiązanie opiera się na teorii grafów traktując tym samym chmurę punktów jako graf nieskierowany, dla którego poszukiwane są spójne składowe. Takie podejście umożliwia pracę bezpośrednio na chmurze punktów.



Rys. 2. Etapy ekstrakcji płaszczyzn: a) Detekcja płaszczyzny za pomocą RANSAC; b) Wyodrębnienie spójnych komponentów; c) Płaszczyzna zachowana po uwzględnieniu kryterium łączności

W rozwijanym algorytmie wyróżnia się dwa główne etapy: sekwencyjną detekcję płaszczyzny za pomocą algorytmu RANSAC oraz analizę spójności sprowadzającą się do ustalenia listy sąsiedztwa dla wszystkich punktów, a następnie etykietowania spójnych komponentów (Rys. 2). Ostatecznie dla każdego wyznaczonego spójnego komponentu obliczana jest powierzchnia (S_k) jako pole otoczki wypukłej, czyli najmniejszego poligonu

zamkniętego obejmującego zbiór. Region o największym polu stanowią punkty reprezentujące „najlepszą powierzchnię”, pozostałe punkty włączane są ponownie do początkowego zbioru danych i uczestniczą w dalszych poszukiwaniach.



Rys. 3. Diagram czynności – zaproponowany algorytm detekcji płaszczyzn

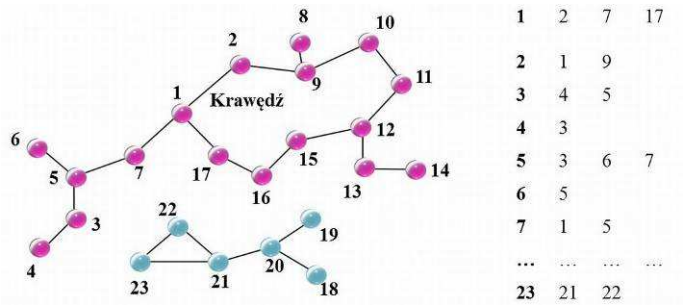
Rysunek 3 przedstawia schemat zaproponowanego algorytmu detekcji płaszczyzn wzbogaconego o analizę spójnych komponentów. Wyodrębnienie spójnych składowych wymaga określenia trzech parametrów: promienia sfery R oraz maksymalnej liczby sąsiadów K, których znaczenie zostanie szczegółowo omówione w następnym paragrafie. Ograniczenie liczby wyodrębnianych komponentów sprowadza się do zdefiniowania parametru N_{min} określającego minimalną liczebność.

3.1. Lista sąsiedztwa

W uproszczeniu graf to zbiór wierzchołków S, wśród których niektóre połączone są za pomocą krawędzi $E \subseteq \{\{u,v\} : u,v \in S, u \neq v\}$. Z kolei każda krawędź zaczyna się i kończy w wierzchołku mogąc tym samym obrazować relacje między obiektami jak na przykład sąsiedztwo dwóch wybranych wierzchołków. Aby usystematyzować relacje pomiędzy punktami chmury punktów – grafu, najwygodniej jest przedstawić chmurę w postaci listy sąsiedztwa, która dla każdego wierzchołka przyporządkowuje listę jego najbliższych sąsiadów – krawędzi wychodzących z punktu (Rys. 4). Skonstruowanie wspomnianej listy wymaga zaimplementowania algorytmu analizującego sąsiedztwo lokalne punktów.

Istnieją dwie najczęściej wykorzystywane strategie określania sąsiedztwa punktu. Pierwsza, polega na wyszukaniu, dla każdego punktu w zbiorze, ustalonej liczby K najbliższych sąsiadów położonych w odległości minimalnej. Według innej definicji, są to

wszystkie punkty zawarte w sferze o promieniu R (Deschaud, 2010). Takie rozwiązanie, w porównaniu do pierwszego, jest niezależne od rozmieszczenia punktów, przy czym dedykowane jest gęstym chmurom punktów. W niniejszej pracy połączono zalety obu metod definiując sąsiedztwo lokalne punktu jako zbiór punktów zawartych w sferze o promieniu R przy jednoczesnym ograniczeniu liczby sąsiadów do K.



Rys. 4. Reprezentacja grafu w postaci listy sąsiedztwa

3.2. Ekstrakcja spójnych komponentów

Wyszukanie spójnych składowych grafu sprowadza się do określenia jego maksymalnych podgrafów, czyli takich, które można wydzielić z całego grafu bez usuwania krawędzi. Jedną z metod eksploracji grafów jest przeszukiwanie w głąb (ang. DFS Depth-first search). Algorytm ten może być zastosowany do wyznaczania spójnych komponentów grafu. Idea jest prosta – procedura rozpoczyna działanie w wybranym wierzchołku grafu oznaczanym jako odwiedzony i nadaje mu odpowiednią etykietę. Następnie przechodzi wzdłuż krawędzi do kolejnego, nieodwiedzonego jeszcze sąsiada. Eksploracja wykonywana jest do momentu natrafienia na wierzchołek, który nie posiada już nieodwiedzonych sąsiadów. Wtedy procedura wraca do poprzedniego odwiedzonego wierzchołka i kontynuuje przeszukiwanie w głąb. Po wykonaniu pełnego przejścia (wszystkie wierzchołki osiągalne z wierzchołka startowego zostały odwiedzone) wybierany jest nowy wierzchołek początkowy, zwiększany numer etykiety, a procedura ponownie wykonuje rekurencyjne poszukiwania. W przeciwnym wypadku algorytm kończy działanie, gdyż wszystkie wierzchołki grafu zostały już odwiedzone, a badany graf jest spójny i stanowi jeden komponent.

Poszukując bardziej wydajnej metody etykietowania spójnych składowych grafu zastosowana została dekompozycja Dulmage –Mendelsohn (DM) macierzy rzadkiej A zawierającej listę wszystkich par wierzchołków sąsiednich. W teorii grafów umożliwia ona w unikalny sposób rozkład grafu dwudzielnego na spójne podgrafy. Z punktu widzenia praktycznego dekompozycja DM odpowiada permutacji wierszy (p) i kolumn (q) macierzy niekwadratowej A do postaci górnej blokowo trójkątnej (7):

$$C = A(p,q) = \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ 0 & 0 & A_{23} & A_{24} \\ 0 & 0 & 0 & A_{34} \\ 0 & 0 & 0 & A_{44} \end{bmatrix} \tag{7}$$

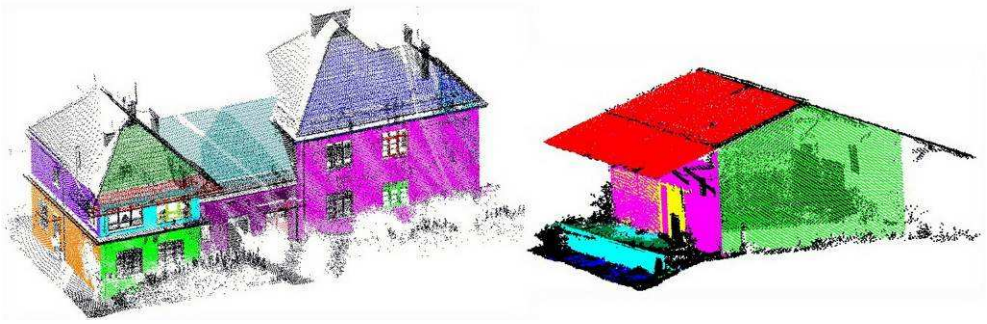
Ostatecznie spójne komponenty obliczane są w oparciu o poniższe równanie:

$$p(r(i) : r(i+1) - 1) \tag{8}$$

gdzie: r- lista wierzchołków przynależących do tego samego spójnego komponentu.

4. PODSUMOWANIE

Testy numeryczne wykonano na chmurach punktów o różnej rozdzielczości, pochodzących z mobilnego i statycznego skaningu laserowego. Uzyskane wyniki pokazują, że algorytm RANSAC wzbogacony o analizę spójnych komponentów poprawnie rozdziela punkty odpowiadające poszczególnym obiektom, leżącym w tej samej płaszczyźnie matematycznej (Rys. 5). Wykorzystanie, nierozpowszechnionej dotąd dekompozycji Dulmage-Mandelsohn przyniosło oczekiwane rezultaty, pozwalając na efektywne wyodrębnienie i zaetykietowanie spójnych komponentów. Najbardziej czasochłonnym etapem metody jest analiza sąsiedztwa punktu, a czas realizacji zadania uzależniony jest od gęstości chmury punktów. Zaproponowany algorytm pozwala na automatyzację procesu detekcji płaszczyzn, wymagając od operatora wprowadzenie jedynie kilku parametrów wejściowych, dostosowanych do przetwarzanych danych. Algorytm jest odpowiedni dla segmentacji nieskomplikowanych modeli, w szczególności głównych powierzchni fasad budynków czy jezdni.



Rys. 5. Rezultat detekcji płaszczyzn w chmurze pozyskanej przez system mobilny Riegl VMX-250. Ustalone parametry: σ=0.01m, P=0.99; R=0.2m, K=25; N_min=200

Istotną wadą RANSAC pozostaje ciągle jego wrażliwość na niską gęstość chmury i nieregularne rozmieszczenie punktów. Fakt ten sprawia, że RANSAC może wyodrębniać „fałszywe" płaszczyzny, które nie znajdują pokrycia w rzeczywistości. Przeprowadzone badania pokazują, że problem ten ujawnia się głównie podczas pracy na surowych chmurach punktów pozyskanych przez systemy mobilne. Możliwe przestoje platformy pomiarowej powodują bowiem nagromadzenie w jednym rejonie znacznej ilości punktów.

Należałoby zatem w przyszłości przeznaczyć dodatkowy czas na ujednorodnienie i filtracje danych wejściowych. Kolejnym kierunkiem ulepszenia opisanego algorytmu będzie optymalizacja czasu poszukiwania sąsiedztwa lokalnego punktu poprzez strukturyzację chmury punktów za pomocą np. kd-drzewa (ang. *kd-tree*).

5. LITERATURA

Awwad T. M., Zhu Q., Du Z., Zhang Y., 2010. An improved segmentation approach for planar surfaces from unstructured 3D point clouds. *The Photogrammetric Record*, 25(129), s. 5-23.

Bauer J., Karner K., Schindler K., Klaus A., Zach C., 2005. Segmentation of building from dense 3D point-clouds. *In Proceedings of the ISPRS. Workshop Laser scanning Enschede* Netherlands, September 12-14.

Boulaassal, H., 2010. Segmentation et modélisation géométrique de façades de bâtiments à partir de relevés laser terrestres. *PhD Thesis*, Université de Strasbourg.

Delmerico J. A., David P., Corso J. J., 2011.Building Facade Detection, Segmentation, and Parameter Estimation for Mobile Robot Localization and Guidance. *In Proceedings of IROS International Conference on Intelligent Robots and Systems.*

Deschaud, J-E., 2010. Traitements de nuages de points denses et modélisation 3D d'environnement par système mobile LIDAR/Caméra. *PhD Thesis*, MINES ParisTech.

Fischler, M. A., Bolles, R. C., 1981. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6), s.381-395.

Hartley R., Zisserman A., 2003. Multiple view geometry in computer vision. *Cambrige University Press,* Second edition, s.117-121.

Jarząbek-Rychard M., Borkowski A., 2010. Porównanie algorytmów RANSAC oraz rosnących płaszczyzn w procesie segmentacji danych z lotniczego skaningu laserowego. *Archiwum Fotogrametrii, Kartografii i Teledetecji*, vol. 21, s.119-129.

Tarsha Kurdi, F., 2008. Extraction et reconstruction de bâtiments en 3D à partir de relevés lidar aéroportés. *PhD Thesis*, Université de Strasbourg.

Tuttas S., Stilla U., 2011. Window Detection In Sparse Point Clouds Using Indoor Points. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 38 (3/W22), s.131-136.

Vosselman G., Gorte B.G.H., Sithole G., Rabbani T., 2004. Recognising structure in laser scanner point clouds. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XXXVI, s. 33-38.

Yang M.Y., Forstner W., 2010. Plane Detection in Point Cloud Data, *Technical Report Nr.1*, Department of Photogrammetry, University of Bonn.

Zuliani M., 2012. RANSAC for Dummies, Technical Report.

http://vision.ece.ucsb.edu/~zuliani/Research/RANSAC/docs/RANSAC4Dummies.pdf

RANSAC ALGORITHM AND ELEMENTS OF GRAPH THEORY FOR AUTOMATIC PLANE DETECTION IN 3D POINT CLOUD

KEY WORDS: point cloud, segmentation, RANSAC, graph, k-nearest neighbour algorithm, labelling, connected component

Summary

Laser scanning techniques play very important role in acquiring of spatial data. Once the point cloud is available, the data processing must be performed to achieve the final products. The segmentation is an inseparable step in point cloud analysis in order to separate the fragments of the same semantic meaning. Existing methods of 3D segmentation are divided into two categories. The first family contains algorithms functioning on principle of fusion, such as surface growing approach or split-merge algorithm. The second group consists of techniques making possible the extraction of features defined by geometric primitives i.e.: sphere, cone or cylinder. Hough transform and RANSAC algorithm (RANdom SAmple Consensus) are classified to the last of aforementioned groups.

This paper studies techniques of point cloud segmentation such as fully automatic plane detection. Proposed method is based on RANSAC algorithm providing an iterative plane modelling in point cloud affected by considerable noise. The algorithm is implemented sequentially, therefore each successive plane represented by the largest number of points is separated. Despite all advantages of RANSAC, it sometimes gives erroneous results. The algorithm looks for the best plane without taking into account the particularity of the object. Consequently, RANSAC may combine points belonging to different objects into one single plane. Hence, RANSAC algorithm is optimized by analysing the adjacency relationships of neighbouring points for each plane. The approach based on graph theory is thus proposed, where the point cloud is treated as undirected graph for which connected components are extracted. Introduced method consists of three main steps: identification of k-nearest neighbours for each point of detected plane, construction of adjacency list and finally connected component labelling.

Described algorithm was tested with raw point clouds, unprocessed in sense of filtration. All the numerical tests have been performed on real data, characterized by different resolutions and derived from both mobile and static laser scanning techniques. Obtained results show that proposed algorithm properly separates points for particular planes, whereas processing time is strictly dependent on number of points within the point cloud. Nevertheless, susceptibility of RANSAC algorithm to low point cloud density as well as irregular points distribution is still an important problem.

This paper contains literature review in subject of existing methods for plane detection in data set. Moreover, the description for proposed algorithm based on RANSAC, its principle, as well as the results is also presented.

Dane autorów:

Mgr inż. Martyna Poręba
e-mail: martyna.poreba@mines-paristech.fr
telefon: 12 617 39 93

Prof. dr hab. inż. François Goulette
e-mail: francois.goulette@mines-paristech.fr
telefon: (+33) 14 051 92 35