

SEGMENTACJA DANYCH MLS Z UŻYCIEM PROCEDUR POINT CLOUD LIBRARY

MLS DATA SEGMENTATION USING POINT CLOUD LIBRARY PROCEDURES

Magdalena Grochowska

Zakład Fotogrametrii, Instytut Geodezji i Kartografii

SŁOWA KLUCZOWE: segmentacja, Point Cloud Library, mobilny skanowanie laserowe, chmura punktów, RANSAC

STRESZCZENIE: Mobilny skanowanie laserowe jest dynamicznie rozwijającą się technologią pomiarową, która staje się coraz bardziej powszechna w zakresie pozyskiwania trójwymiarowej informacji przestrzennej. W artykule przedstawiono wstępne wyniki segmentacji danych pozyskanych techniką MLS. Dane użyte w niniejszej pracy pozyskano w ramach pomiaru inwentaryzacyjnego infrastruktury linii kolejowej na odcinku Warszawa Zachodnia – Rembertów. Pomiar chmury punktów zrealizowano za pomocą zespołu skanerów profilowych zainstalowanych na platformie kolejowej. Proces segmentacji wykonano przy zastosowaniu narzędzi programu ‘open source’ Point Cloud Library. Wykorzystano szablony biblioteki PCL, która zawiera algorytmy segmentacji do odrębnych klastrów. Algorytmy te są najlepiej dostosowane do przetwarzania chmury punktów, składającej się z określonej liczby izolowanych przestrzennie regionów. Szablon biblioteki wykonuje ekstrakcję klastra w oparciu o dopasowanie modelu poprzez metody konsensusu próbek dla różnych parametrycznych modeli (płaszczyzny, walce, kule, linie, itp.). Większość operacji matematycznych jest realizowana na podstawie biblioteki Eigen, zbioru szablonów algebry liniowej. W niniejszym referacie przedstawiono szereg zagadnień związanych z zastosowaniem oprogramowania PCL i jego możliwościami.

1. WPROWADZENIE

Znaczny rozwój technologii pomiarowej MLS w ostatnich latach spowodował duże zainteresowanie rynku produktami otrzymywanymi z mobilnego skanowania laserowego. System ten jest najchętniej wykorzystywany do mapowania obiektów liniowych, w tym szlaków linii kolejowej. Bezinwazyjny pomiar, który nie zakłóca pracy, ani płynności w ruchu pasażerskim i towarowym jest nad wyraz istotny na tego typu obiektach. Dane użyte w niniejszej pracy uzyskano dzięki uprzejmości firmy Gispro Sp. z o.o. ze Szczecina, która wykonała pomiar i udostępniła wyniki w celach badawczo-naukowych Instytutowi Geodezji i Kartografii. Dane pozyskano w ramach pomiaru inwentaryzacyjnego infrastruktury fragmentu linii kolejowej na odcinku Warszawa Zachodnia – Rembertów (rys. 1). Pomiar chmury punktów zrealizowano za pomocą zespołu skanerów profilowych zainstalowanych na platformie kolejowej. W celu ograniczenia czasochłonności pomiaru,

samochód wraz z zainstalowanymi trzema skanerami oraz systemem pozycjonowania GPS/IMU, wjechał na platformę kolejową, a całość poruszana była przez lokomotywę. Trasa o łącznej długości 2.3 km przechodzi przez tunel średnicowy, co przysporzyło duże trudności podczas pomiaru. W zamkniętych przestrzeniach tunelu występował przez długi czas problem z odczytem sygnału GPS. Pomiar odbywał się przy prędkości wahającej się od 20 do 50 km/h, uwarunkowane było to głównie harmonogramem ruchu pociągów, a zarazem sposobnością przejazdu. Dokładność względna na całym odcinku linii kolejowej jest wyrażona błędem średnim ± 5 mm, natomiast 90% obserwacji wynosi średnio ± 7 mm. Dokładność bezwzględna oscyluje w przedziale od ± 3 cm do ± 50 cm, co jest spowodowane trudnością odczytu sygnału GPS w głębi tunelu. Otrzymane wyniki są w dużej mierze spowodowane brakiem osnowy, cały pomiar opierał się jedynie na obserwacjach GPS/IMU, a w tunelu nawet na samej jednostce inercyjnej.

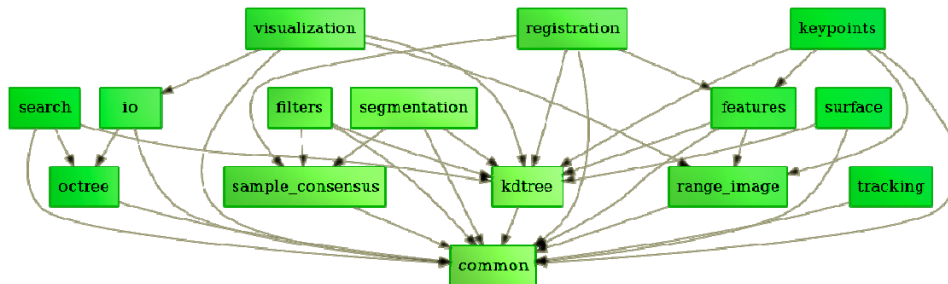


Rys. 1. Widok chmury punktów w tunelu średnicowym (Gispro Sp. z o.o.)

2. POINT CLOUD LIBRARY

Do przetworzenia danych wykorzystano narzędzia programu 'open source' Point Cloud Library. Narzędzia te umożliwiają wykorzystanie szablonów bibliotek programistycznych. PCL jest otwartym, samodzielny projektem, działającym na dużą skalę, służącym do przetwarzania 2D/3D obrazu oraz chmury punktów. Oprogramowanie PCL jest zgodne z warunkami licencji BSD (Berkeley Software Distribution License), co oznacza, że jest darmowym narzędziem do użytku komercyjnego i badań naukowych. W celu ułatwienia rozwoju, PCL jest podzielone na szereg mniejszych bibliotek kodu, które mogą być tworzone oddzielnie. Ta modularność jest ważna dla dystrybucji PCL na platformach z obniżonymi możliwościami obliczeniowymi. Struktura PCL podzielna jest na liczne moduły, w tym filtrację, estymację cech charakterystycznych, rekonstrukcję płaszczyzn, rejestrację, wpasowywanie modeli czy segmentację. Moduły te mogą być stosowane, na przykład, do filtracji odstających danych, łączenia ze sobą chmur punktów 3D, segmentacji odpowiadających sobie części sceny, wyodrębniania punktów strategicznych, obliczania deskryptorów w celu rozpoznawania obiektów na podstawie ich kształtu geometrycznego, tworzenia powierzchni oraz wizualizacji (<http://www.pointclouds.org>).

Aby móc dostrzec złożoność i funkcjonalność narzędzi PCL, przedstawiono schemat możliwości połączeń między poszczególnymi modułami (rys. 2).



Rys. 2. Graf modułów PCL (<http://www.pointclouds.org>)

Segmentacja danych przy zastosowaniu narzędzi Point Cloud Library odbywa się w oparciu o zastosowanie szablonów biblioteki `pcl_segmentation`, która zawiera algorytmy segmentacji do odrębnych klastrów. Algorytmy te są najlepiej dostosowane do przetwarzania chmury punktów, składającej się z określonej liczby izolowanych przestrzennie regionów. Szablon biblioteki wykonuje ekstrakcję klastra w oparciu o dopasowanie modelu poprzez metody konsensusu próbek dla różnych parametrycznych modeli (płaszczyzny, walce, kule, linie, itp.). Większość operacji matematycznych jest realizowana na podstawie biblioteki Eigen, zbioru szablonów algebry liniowej.

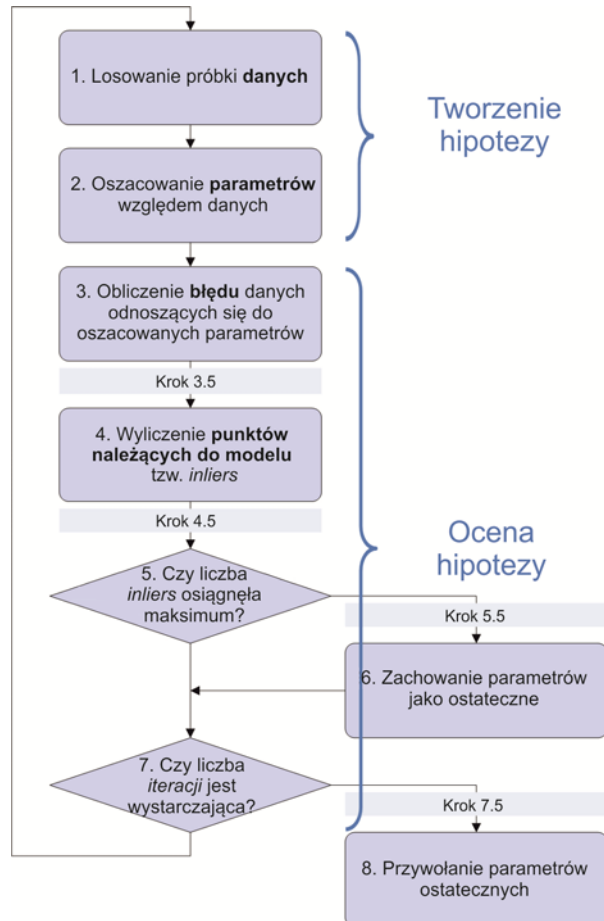
3. ALGORYTM RANSAC

Algorytm RANSAC (Random Sample Consensus) po raz pierwszy został opublikowany w 1981 roku przez Fischler'a i Bolles'a i jest jednym z najczęściej stosowanych algorytmów, które mają na celu skuteczne wykrycie kształtu chmury punktów, nawet z dużego zbioru punktów. RANSAC jest w stanie wyodrębnić różne rodzaje podstawowych kształtów, przy zachowaniu korzystnych właściwości paradygmatu jak solidność, ogólność i prostota (Schnabel *et al.*, 2007). W szczególności jest powszechnie stosowany w systemach widzenia komputerowego.

Konsensus próby losowej jest iteracyjną metodą pozwalającą na oszacowanie parametrów szukanego modelu matematycznego obiektu opierając się na zestawie danych, spośród których tylko część poprawnie przedstawia modelowaną powierzchnię. Głównym założeniem algorytmu jest występowanie w danych wejściowych zarówno punktów należących do modelu tzw. *inliers*, jak również wartości odstających (błędnych) tzw. *outliers*. Źródłem punktów, które odbiegają od próbki statystycznej mogą być błędne pomiary, szumy lub zakłócenia czy jeszcze inne elementy.

Działanie algorytmu RANSAC (rys. 3) polega na najlepszym dopasowaniu modelowanej powierzchni spośród punktów 3D chmury. W tym celu algorytm losowo wybiera podzbiór danych oraz oblicza parametry tak, aby pasowały do próbki. Wielkość próby zależy od modelu matematycznego (linia, płaszczyzna, walec, kula, itp.), który chcemy znaleźć. Zazwyczaj, rozmiar próbki jest najmniejszą liczbą punktów wystarczającą do określenia parametrów modelu (Deschaud, Goulette, 2010). Następnie, w trakcie

realizacji algorytmu wielkość podzbioru ulega modyfikacjom, aż do momentu kiedy metoda wykryje wszystkie dane pasujące do szukanego modelu zgodnie z określonymi parametrami.



Rys. 3. Schemat algorytmu RANSAC (Choi *et al.*, 2009)

Do uruchomienia algorytmu potrzebne są dane wejściowe jak:

- zbiór 3D punktów obserwacji $[X, Y, Z]$ tworzące macierz,
- próg tolerancji, zakres odległości pomiędzy wybraną płaszczyzną, a pozostałymi punktami (t),
- liczba punktów potrzebnych do jednoznacznego dopasowania szukanego modelu geometrycznego (s),
- prawdopodobieństwo zdefiniowania co najmniej jednego zestawu obserwacji należących do płaszczyzny (p).

W celu wyliczenia minimalnej liczby (I) prób potrzebnych do osiągnięcia prawdopodobieństwa (p), aby znaleźć co najmniej jeden dobry zestaw obserwacji (Boulaassal *et al.*, 2007), można posłużyć się wzorem (1).

$$I = \frac{\log(1-p)}{\log(1-n)} \quad (1)$$

gdzie n można wyliczyć na podstawie równania (2).

$$n = \left(\frac{N_i}{N} \right)^s \quad (2)$$

W powyższym wzorze (N_i) charakteryzuje liczbę punktów, które należą do wykrytej płaszczyzny, (N) jest liczbą wszystkich punktów reprezentujących dane wejściowe, natomiast (s) jest minimalną liczbą punktów koniecznych do obliczenia parametrów modelu. W przypadku płaszczyzn parametr ten wynosi trzy punkty.

4. SEGMENTACJA DANYCH

W tej części pracy pokazano rezultat segmentacji danych z mobilnego skaningu laserowego, jaki uzyskano podczas wstępnych działań z narzędziami Point Cloud Library. Opracowano program oparty o język C++ w celu automatycznej segmentacji i ekstrakcji płaszczyzn. Proces klasyfikacji danych odbywa się w oparciu o zastosowanie szablonów biblioteki PCL, która zawiera algorytmy grupowania do odrębnych klastrów. Algorytmy te są najlepiej dostosowane do przetwarzania chmury punktów, składającej się z określonej liczby izolowanych przestrzennie regionów. W celu skrócenia czasu procesu wyodrębniania płaszczyzn, ograniczono ilość danych i z odcinka liczącego 2.3 km, wybrano punkty reprezentujące fragment tunelu średnicowego.

W analizowanym przypadku, zastosowane moduły umożliwiają wyodrębnienie ścian tunelu, zarówno w płaszczyźnie pionowej jak i w płaszczyźnie poziomej. W celu wyodrębnienia poszczególnych płaszczyzn, zastosowano z biblioteki PCL moduł PCL_SAMPLE_CONSENSUS, wykorzystujący algorytm RANSAC. Do wyznaczenia powierzchni poziomej wybrano SACMODEL_PLANE, natomiast do powierzchni pionowej SACMODEL_PERPENDICULAR_PLANE.

W przestrzeni euklidesowej R^3 płaszczyznę można opisać za pomocą równania ogólnego o czterech parametrach A, B, C, D (3).

$$Ax + By + Cz + D = 0 \quad (3)$$

Trzy pierwsze wartości nie mogą być jednocześnie równe zero i określają jednostkowy wektor normalny, równanie (4).

$$(A^2 + B^2 + C^2 = 1) \quad (4)$$

Czwarta wartość jest odległością płaszczyzny od hipotetycznej płaszczyzny. Wszystkie punkty (x, y, z) spełniające równanie 3 należą do tej samej płaszczyzny.

W rzeczywistości jednak, dane uzyskane przez skaniny laserowy nie są bezpośrednio zgodne z modelem matematycznym. Innymi słowy, nie występują idealnie płaskie ściany, nie ma idealnie prostych krawędzi. W związku z tym, aby uzyskać płaszczyznę reprezentującą ściany, nakłada się wartość tolerancji (tzw. próg tolerancji) opisującą grubość płaszczyzny. W ten sposób badana płaszczyzna jest rozpatrywana jako równoległościan, ale jest to konieczne w celu uzyskania znaczących segmentów.

4.1. Segmentacja płaszczyzn poziomych

Operację segmentacji w celu wyodrębnienia płaszczyzny poziomej za pomocą algorytmu RANSAC wykonano według niżej przedstawionego algorytmu (rys. 4). Wykorzystano model SACMODEL_PLANE. W prawdzie złożoność tego modelu jest dosyć mała, jednak wystarczająca do nieskomplikowanych obiektów.

```
// Wstawianie danych chmury punktów
pcl::pcdReader reader;
reader.read (D:/projekty/segmentacja_płaszczyzn/tunel.pcd", *cloud);

pcl::ModelCoefficients::Ptr coefficients (new pcl::ModelCoefficients);
pcl::PointIndices::Ptr inliers (new pcl::PointIndices);
// Tworzenie obiektu segmentacji
pcl::SACSegmentation<pcl::PointXYZ> seg;
// Ustawienia opcjonalne
seg.setOptimizeCoefficients (true);
// Ustawienia obowiązkowe
seg.setModelType (pcl::SACMODEL_PLANE);
seg.setMethodType (pcl::SAC_RANSAC);
seg.setDistanceThreshold (0.70);

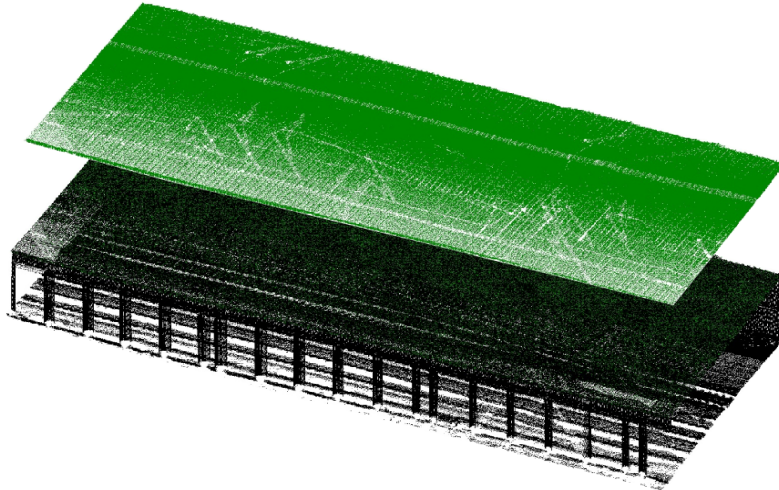
seg.setInputCloud (cloud.makeShared ());
seg.segment (*inliers, *coefficients);
```

Rys. 4. Fragment utworzonego algorytmu do segmentacji płaszczyzn poziomych

Wynik przeprowadzonej segmentacji pokazano na rysunku 5. Kolorem czarnym oznaczone są punkty wejściowe, natomiast uzyskaną powierzchnię płaską zaprezentowano kolorem zielonym.

Wadą danego modelu, a zarazem algorytmu RANSAC jest włączanie punktów błędnych, które nie należą do danej płaszczyzny np. punktów należących do ścian bocznych. Spowodowane jest to występowaniem tych punktów w zakresie tolerancji, którą ustala użytkownik. W związku z tym wartość progu należy dobrać bardzo starannie.

Po wielu doświadczeniach, okazało się, że wartość tolerancji w określonym przypadku, w celu uzyskania znaczącej płaszczyzny sufitu tunelu średnicowego powinna być ustawiona na $t = 70$ cm. Mniejsze progi wyznaczały bardzo liczne segmenty, które nie reprezentowały powierzchni w całości i posiadały niewystarczającą liczbę punktów. Natomiast wartości powyżej zadanej łączyły podczas segmentacji więcej płaszczyzn i były rozpatrywane jako jedna płaska powierzchnia.



Rys. 5. Prezentacja uzyskanej płaszczyzny poziomej w PCL

4.2. Segmentacja płaszczyzn pionowych

Operację segmentacji w celu wyodrębnienia płaszczyzny pionowej, również za pomocą algorytmu RANSAC wykonano według przedstawionego algorytmu (rys. 6). Wykorzystano model SACMODEL_PERPENDICULAR_PLANE. Wspomniany model jest bardziej złożony od SACMODEL_PLANE i w przeciwieństwie do niego bierze pod uwagę jeszcze jedno dodatkowe założenie. Wypadkową płaszczyzną jest powierzchnia prostopadła do zadanych osi.

```
// Wstawianie danych chmury punktów
pcl::pcdReader reader;
reader.read (D:/projekty/segmentacja_płaszczyzn/tunel.pcd", *cloud);

pcl::ModelCoefficients::Ptr coefficients (new pcl::ModelCoefficients);
pcl::PointIndices::Ptr inliers (new pcl::PointIndices);
// Tworzenie obiektu segmentacji
pcl::SACSegmentation<pcl::PointXYZ> seg;
// Ustawienia opcjonalne
seg.setOptimizeCoefficients (true);
// Ustawienia obowiązkowe
seg.setModelType (pcl::SACMODEL_PLANE);
seg.setMethodType (pcl::SAC_RANSAC);
seg.setDistanceThreshold (0.30);
seg.setAxis (Eigen::Vector3f (0.0, 0.0, 1.0));
seg.setEpsAngle (pcl::deg2rad (3.0));

seg.setInputCloud (cloud.makeShared ());
seg.segment (*inliers, *coefficients);
```

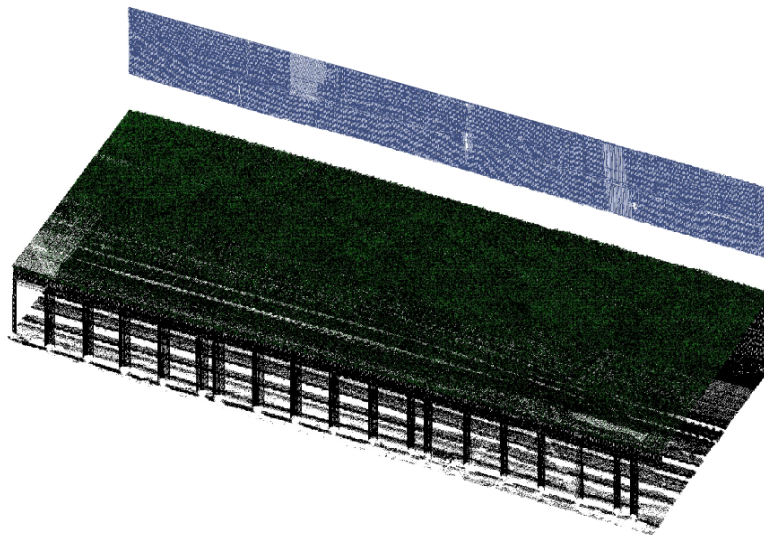
Rys. 6. Fragment utworzonego algorytmu do segmentacji płaszczyzn poziomych

W danym przypadku okazało się, że wartość tolerancji, w celu uzyskania znaczącej płaszczyzny ściany bocznej tunelu (druga ściana jest w postaci cylindrycznych kolumn),

powinna być ustawiona na $t = 30$ cm. Dodatkowo ustalono wartość kąta odchylenia (3°), który określa maksymalną dopuszczalną różnicę położenia punktów wyznaczonej płaszczyzny w stosunku do osi. Jest ona istotna w szczególności w momentach kiedy zbiór punktów wejściowych nie jest położony idealnie równoległe względem osi i występują odchylenia.

Podobnie jak w poprzednim modelu, zarówno mniejsze, jak i większe progi tolerancji, nie wyznaczały poprawnie powierzchni w całości i posiadały niewystarczającą liczbę punktów lub kilka płaszczyzn było rozpatrywanych jako jedna. Dodatkowo należało przetransformować zbiór punktów do położenia równoległego względem osi. Tylko w takim przypadku wykorzystany model SACMODEL_PERPENDICULAR_PLANE wyodrębni poprawnie płaszczyznę pionową.

Wynik przeprowadzonej segmentacji pokazano na rysunku 7. Kolorem czarnym oznaczone są punkty wejściowe, natomiast uzyskaną powierzchnię płaską zaprezentowano kolorem fioletowym.



Rys. 7. Prezentacja uzyskanej płaszczyzny poziomej w PCL

5. PODSUMOWANIE I WNIOSKI

Pisanie kodów źródłowych przy zastosowaniu bibliotek PCL jest stosunkowo łatwym sposobem na osiągnięcie wyznaczonego celu. Oprogramowanie zawiera wiele algorytmów do przetwarzania obrazu w postaci chmury punktów, a dodatkowo w bibliotekach internetowych umieszczone są liczne przykłady, ćwiczenia oraz instrukcje krok-po-kroku.

Przy zastosowanym algorytmie można dostrzec zarówno zalety, jak i ograniczenia w zakresie jego zastosowania. Algorytm w krótkim czasie wykonał segmentację na danych testowych, jednak przy źle dobranych parametrach łączy ze sobą odrębne obiekty leżące

w tej samej płaszczyźnie. Stopień poprawności poszukiwania modelu jest różny dla każdej danej, a jego wartość jest nieznaną i niemożliwą do jednoznacznego określenia. Optymalny wynik może być uzyskany jedynie w empiryczny sposób, który jest silnie uzależniony od rodzaju badanego obiektu, jego właściwości oraz celu badania.

W przeprowadzonym badaniu oceniono dokładność wykrywania uzyskanych płaszczyzn przez prezentowane podejścia. W tym celu potrzebny był model referencyjny, który otrzymano poprzez przeprowadzenie manualnej segmentacji na podstawie oryginalnych danych wejściowych. Nałożenie na siebie dwóch płaszczyzn dało możliwość analizy jakościowej, która wykazała, że są one podobne. Natomiast analiza ilościowa wykazała skuteczność rzędu 95% w przypadku modelu SACMODEL_PLANE oraz 93% w przypadku modelu SACMODEL_PERPENDICULAR_PLANE.

6. LITERATURA

Boulaassal H., Landes T., Grussenmeyer P., Tarsha-Kurdi F., 2007. Automatic segmentation of building facades using terrestrial laser data. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Espoo, Finland, Vol. XXXVI, Part 3 / W52, s. 65-70.

Choi S., Kim T., Yu W., 2009. Performance Evaluation of RANSAC Family. *The British Machine Vision Conference*, London, UK.

<http://www.bmva.org/bmvc/2009/Papers/Paper355/Paper355.pdf>

Fischler, A., Bolles, 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.

Schnabe R., Wahl R., Klein R., 2007. Efficient RANSAC for point-cloud shape detection. *Computer Graphics Forum*, 26 (2), s. 214–226.

Deschaud J.E., Goulette F., 2010. A fast and accurate plane detection algorithm for large noisy point clouds using filtered normals and voxel growing. Proceedings of the Fifth International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT'10), Paris, France.

https://diglib.org/EG/DL/PE/3DPVT/3DPVT2010Papers/29_paper111.pdf

PointCloudLibrary, 2012: <http://www.pointclouds.org>

MLS DATA SEGMENTATION USING POINT CLOUD LIBRARY PROCEDURES

KEY WORDS: segmentation, Point Cloud Library, mobile laser system, point cloud, RANSAC

Summary

Mobile laser scanning is dynamically developing measurement technology, which is becoming increasingly widespread in acquiring three-dimensional spatial information. Continuous technical progress based on the use of new tools, technology development, and thus the use of existing resources in a better way, reveals new horizons of extensive use of MLS technology. Mobile laser scanning system is usually used for mapping linear objects, and in particular the inventory of roads, railways, bridges, shorelines, shafts, tunnels, and even geometrically complex urban spaces. The

measurement is done from the perspective of use of the object, however, does not interfere with the possibilities of movement and work.

This paper presents the initial results of the segmentation data acquired by the MLS. The data used in this work was obtained as part of an inventory measurement infrastructure railway line. Measurement of point clouds was carried out using a profile scanners installed on the railway platform. To process the data, the tools of 'open source' Point Cloud Library was used. These tools allow to use templates of programming libraries. PCL is an open, independent project, operating on a large scale for processing 2D/3D image and point clouds. Software PCL is released under the terms of the BSD license (Berkeley Software Distribution License), which means it is a free for commercial and research use. The article presents a number of issues related to the use of this software and its capabilities.

Segmentation data is based on applying the templates library `pcl_segmentation`, which contains the segmentation algorithms to separate clusters. These algorithms are best suited to the processing point clouds, consisting of a number of spatially isolated regions. Template library performs the extraction of the cluster based on the fit of the model by the consensus method samples for various parametric models (planes, cylinders, spheres, lines, etc.). Most of the mathematical operation is carried out on the basis of Eigen library, a set of templates for linear algebra.

Dane autora:

Mgr inż. Magdalena Grochowska
e-mail: magdalena.grochowska@igik.edu.pl