

Rejestracja chmur punktów: komponenty systemu

Tomasz Kornuta

IBM Research – Almaden, 650 Harry Rd, San Jose, CA 95120, Stany Zjednoczone

Marta Jolanta Łępicka

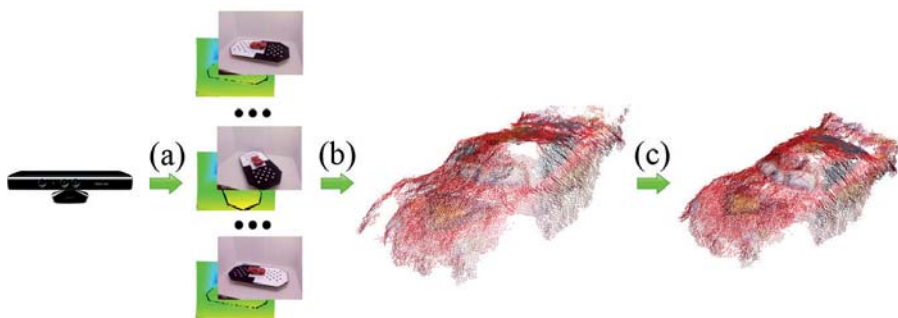
Politechnika Warszawska, Instytut Automatyki i Informatyki Stosowanej, ul. Nowowiejska 15/19, 00-665 Warszawa

Streszczenie: Dwuczęściowy artykuł dotyczy problemu rejestracji obrazów RGB-D. W robotyce problem ten znany jest pod pojęciem V-SLAM (ang. Visual Simultaneous Localization and Mapping). W poniższej, pierwszej części artykułu omówiono pokrótce główne komponenty typowego systemu rejestracji, a następnie zawężono uwagę do algorytmu ICP (ang. Iterative Closest Point), służącego do wzajemnej rejestracji chmur punktów. W drugiej części artykułu uwagę skupiono na asocjacji chmur punktów, różnego rodzaju atrybutach punktów, które mogą być wykorzystane podczas znajdowania dopasowań oraz szeregu metryk operujących na tych atrybutach. Pokrótce omówiono zastosowaną metodykę badań, zaprezentowano eksperymenty mające na celu porównanie wybranych odmian algorytmu ICP oraz omówiono otrzymane wyniki.

Słowa kluczowe: obraz RGB-D, chmura punktów, V-SLAM, rejestracja, wzajemne łączenie, ICP

1. Wprowadzenie

Łączenie wielu widoków danej sceny w jeden spójny, trójwymiarowy model zwane jest rejestracją (ang. *registration*) [13]. W przypadku wykorzystania czujników RGB-D [16, 31] problem rejestracji może być sformułowany jako znalezienie względnych transformacji między kolejnymi chmurami punktów oraz przekształcenie współrzędnych wszystkich punktów do wspólnego układu odniesienia. Istnieje szereg potencjalnych aplikacji rejestracji w robotyce [24], wśród których najważniejszymi są: budowanie trójwymiarowej mapy otoczenia (np. [2]) oraz generowanie trójwymiarowych modeli obiektów (np. [15]) na potrzeby ich późniejszego wykorzystania podczas rozpoznawania obiektów znajdujących się na scenie. Ideę procesu rejestracji modelu obiektu zaprezentowano na rys. 1. W kroku (a) następuje akwizycja zestawu widoków danego obiektu z różnych stron – w przypadku czujnika Kinect widok sta-



Rys. 1. Poglądowy schemat procesu rejestracji modelu obiektu

Fig. 1. Image presenting the idea of the registration of a model of an object

nowi para obrazów (obraz RGB oraz odpowiadająca mu mapa głębi). Celem kroku (b) jest przekształcenie tych widoków do postaci chmur punktów oraz dopasowanie par chmur do siebie. Ponieważ otrzymany w ten sposób model jest często niespójny, niezbędne jest wykonanie kroku (c), którego celem jest optymalizacja modelu (tzw. domknięcia pętli), w wyniku czego otrzymany zostaje lepszy, bardziej spójny model obiektu.

Teoretycznie w przypadku eksteroreceptorów (np. kamera, lidar lub czujnik RGB-D) zintegrowanych z robotem, transformacja między kolejnymi odczytami mogłaby być oszacowana na podstawie odczytów z czujników proprioceptywnych, np. enkoderów oraz tzw. odometrii (roboty mobilne) lub przez rozwiązanie prostego zagadnienia kinematyki (manipulatory). Niestety, zakłócenia w postaci poślizgu kół, tarcia czy niedokładności parametrów kinematycznych utrudniają prawidłowe dopasowanie kolejnych widoków i znacząco obniżają spójność trajektorii ruchu oraz wynikowej mapy otoczenia. W przypadku, gdy estymacja ruchu robota bazuje na informacji wizyjnej, mamy

Autor korespondujący:

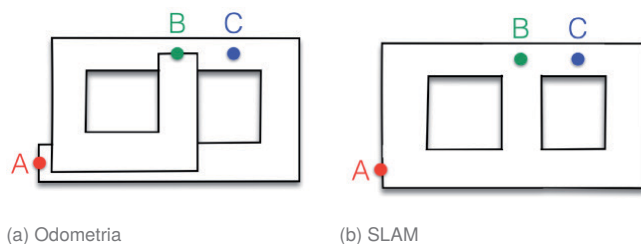
Tomasz Kornuta, tkornuta@gmail.com

Artykuł recenzowany

nadesłany 22.02.2017 r., przyjęty do druku 29.03.2017 r.



Zezwala się na korzystanie z artykułu na warunkach licencji Creative Commons Uznanie autorstwa 3.0



Rys. 2. Poglądowy rysunek obrazujący różnicę między mapami zbudowanymi z wykorzystaniem odometrii oraz SLAM (źródło: [4])
Fig. 2. An illustration of the difference between maps build relying on odometry and SLAM (source: [4])

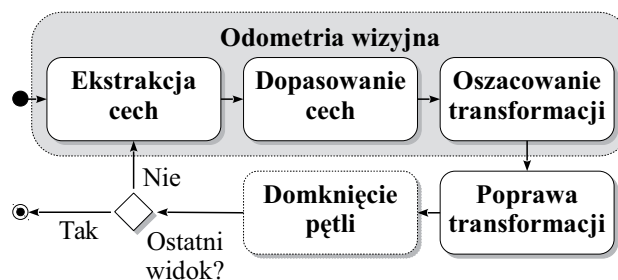
do czynienia z tzw. odometrią wizyjną VO (ang. *Visual Odometry*) [23]. Odometrię wizyjną można zdefiniować jako określanie względnego ruchu obserwatora (np. kamery, czujnika RGB-D) na podstawie kolejnych obrazów, gdzie ważna jest głównie lokalna spójność otrzymanej trajektorii ruchu. Robot wykorzystujący wizualną odometrię wykonuje „inkrementalną rejestrację”, w pewnym sensie porusza się w świecie „nieskończonego korytarza”. Skupia przy tym uwagę na możliwie najlepszym dopasowaniu nowego odczytu do poprzedniego, jednocześnie pomijając ważne aspekty topologiczne, jak np. ponowne odwiedzenie tego samego obszaru (rys. 2a). W konsekwencji prowadzi to do kumulacji błędów (tzw. dryfu), analogicznie jak w przypadku zwykłej odometrii.

Konieczne jest opracowanie algorytmu, który poza integracją kolejnych widoków mógłby jednocześnie śledzić trajektorię poruszającego się czujnika oraz analizować topologię środowiska w celu zmniejszenia dryfu (rys. 2b). W robotyce problem ten znany jest jako jednoczesna lokalizacja i mapowanie SLAM (ang. *Simultaneous Localization and Mapping*) [4, 32]. Celem SLAM jest uzyskanie estymacji ruchu obserwatora spójnej globalnie, co wymaga ciągłego budowania i przechowywania mapy, która wykorzystywana jest do znajdowania odwiedzonych już miejsc, czyli tzw. domknięcia pętli (ang. *Loop Closure*). Wykrycie domknięcia pętli oraz wykorzystanie tej informacji do poprawy spójności trajektorii oraz budowanej mapy są głównymi problemami związanymi ze SLAM.

W pracach [11, 29] zaproponowano dwa podstawowe kryteria podziału metod odometrii wizyjnej. Pierwsze kryterium dotyczy podziału na systemy mono- bądź stereo kamerowe, przy czym do drugiej kategorii zaliczano również szereg rozwiązań wykorzystujących pojedynczą kamerę umieszczoną na szynie (ang. *slider stereo*, zaproponowany w pracy [20]), co umożliwił akwizycję szeregu klatek z różnych ujęć oraz – w konsekwencji – wyznaczenie głębi. Kryterium to dotyczy wykorzystania informacji na temat głębi, a nie liczby kamer zastosowanych w systemie.

Drugie kryterium dotyczy podziału metod wyznaczania względnego ruchu kamery na podstawie dwóch kolejnych obrazów. Wyróżnia się tu metody wykorzystujące całe obrazy (ang. *Appearance-based methods*) oraz metody operujące na cechach (ang. *Feature-based methods*). W przypadku drugiej kategorii jest mowa o cechach punktowych, tzn. ekstrahowanych w pewnym otoczeniu wybranych punktów obrazu (tzw. punktów kluczowych). W przypadku uwzględniania głębi podział ten dotyczy wykorzystania gęstych bądź rzadkich chmur punktów – gęste chmury tworzone są na podstawie wszystkich pikseli obrazów, natomiast chmury rzadkie tworzone są na podstawie znalezionych punktów kluczowych.

Kryteria te w naturalny sposób przekładają się na klasyfikację zarówno całych systemów rejestracji, jak i ich poszczególnych komponentów. Zakres prezentowanej pracy został ograniczony do systemów rejestracji opartych na czujnikach RGB-D



Rys. 3. Główne komponenty typowego systemu trójwymiarowej rejestracji wykorzystującego odometrię wizyjną bazującą na cechach
Fig. 3. Major components of a typical 3D registration system relying on feature-based visual odometry

wykorzystujących zarówno głębię, jak i informacje o kolorze. W następnej sekcji przedstawiono z kolei komponenty systemu rejestracji, gdzie odometria wizyjna operuje na cechach (rzadkich chmurach punktów) – co jest obecnie dominującym rozwiązaniem. Natomiast w sekcji trzeciej skupiono uwagę na algorytmie ICP służącym do wzajemnej rejestracji par chmur punktów oraz przedstawiono poszczególne kroki jego działania niezależnie od rodzaju chmur (rzadkich bądź gęstych) podawanych na jego wejście. Celem tej pracy jest bowiem wprowadzenie do problemu rejestracji i zaprezentowanie szerszego kontekstu dla wyników badań nad rejestracją wzajemną, którym poświęcono drugą część artykułu.

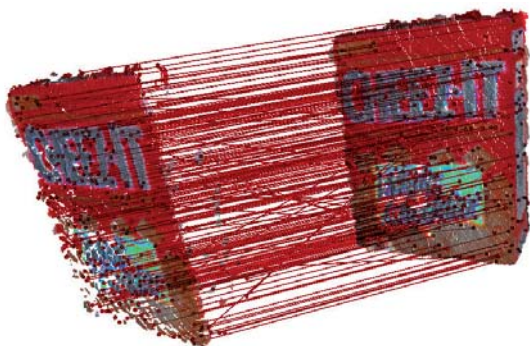
2. System rejestracji

Diagram stanów typowego systemu rejestracji pokazano na rys. 3. W kolejnych podsekcjach omówiono przeznaczenie oraz działanie poszczególnych elementów takiego systemu.

2.1. Ekstrakcja cech

Celem odometrii wizyjnej jest estymacja ruchu kamery względem otoczenia. W typowym systemie (np. [12]) jest to realizowane przez wyznaczenie transformacji między dwoma kolejnymi widokami w oparciu o dopasowania znalezione między cechami wyekstrahowanymi z obu widoków. Cechy ekstrahowane z obrazów dzielą się na dwie główne kategorie: cechy lokalne i globalne. Globalne cechy obliczane są na podstawie całego obrazu lub całej chmury punktów (np. histogram koloru), mogą być również związane z konkretnym segmentem obrazu (np. momenty opisujące kształt). Cechy lokalne (lub cechy punktowe) związane są z kolei z pojedynczym pikselem obrazu (lub punktem chmury) i opisują pewną lokalną charakterystykę tego punktu. Charakterystyka ta wyznaczana jest dla określonego sąsiedztwa punktu i zapisywana w postaci deskryptora. Ponieważ obliczenie deskryptora dla każdego punktu jest czasochłonne, dlatego też obliczenia te przeprowadzane są tylko dla wybranych punktów, zwanych punktami kluczowymi.

W systemach rejestracji zwykle wykorzystywane są cechy lokalne, gdyż cechy globalne ekstrahowane z tego samego obiektu/fragmentu sceny dla dwóch różnych widoków mogą być diametralnie różne. Cechy lokalne natomiast mogą być wykorzystane do znalezienia dopasowań między różnymi widokami sceny bądź obiektu nawet w przypadku znacznego przesłonięcia czy dużych różnic w wyglądzie. Mimo iż istnieją algorytmy umożliwiające znalezienie dopasowań i wyznaczanie transformacji między cechami lokalnymi bazującymi tylko na obrazach kolorowych, dodatkowa informacja w postaci mapy głębi znacznie je usprawnia. Połączenie cech i informacji o głębi umożliwia generowanie tzw. rzadkich chmur cech.



Rys. 4. Dwie przykładowe chmury z zaznaczonymi cechami SIFT oraz znalezionymi dopasowaniami

Fig. 4. Two exemplary point clouds along with the extracted SIFT features and found correspondences

Efektywność wyznaczonych lokalnych cech zależy od użytego detektora punktów kluczowych oraz rodzaju deskryptora. Istnieje szereg cech posiadających swoje własne detektory i deskryptory, na przykład SIFT (ang. *Scale Invariant Feature Transform*) [18]. Dość popularne są binarne deskryptory będące w istocie wektorami zero-jedynkowymi, co pozwala na szybkie ich porównywanie (miara Hamminga). Przykładami tych cech są BRIEF (ang. *Binary Robust Independent Elementary Features*) [5], BRISK (ang. *Binary Robust Invariant Scalable Keypoints*) [17] oraz FREAK (ang. *Fast Retina Keypoint*) [1].

Istnieją również cechy, detektory punktów kluczowych oraz deskryptory wykorzystujące informacje o głębi lub operujące na chmurach punktów. Dobrym przykładem cech ekstrahowanych z map głębi są cechy NARF (ang. *Normal Aligned Radial Feature*) [30], natomiast przykładami cech lokalnych ekstrahowanych z chmury punktów mogą być cechy SHOT (ang. *Signature of Histograms of Orientation*) [33] lub FPFH (ang. *Fast Point Feature Histogram*) [28]. Warto zaznaczyć, że istnieje szereg rozwiązań, w których rzadka chmura cech tworzona jest na podstawie cech lokalnych wyekstrahowanych z obrazu RGB, których współrzędne są następnie przetwarzane do współrzędnych kartezjańskich przez wykorzystanie informacji z mapy głębi skojarzonej z danym obrazem RGB (np. [14]).

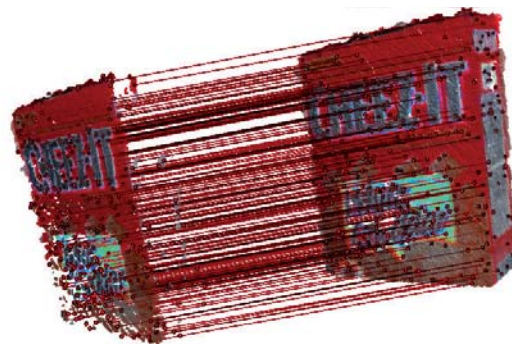
2.2. Dopasowanie cech

Rozwiązanie problemu dopasowania dwóch zbiorów punktów jest zazwyczaj realizowane za pomocą algorytmu k -najbliższych sąsiadów (ang. *k-Nearest Neighbors*, k -NN). Ponieważ deskryptory cech punktowych są wektorami o wielu wymiarach (dla przykładu deskryptor cech SIFT jest 128-wymiarowy), problem dopasowania cech staje się wielowymiarowy i wyszukiwanie najbliższego sąsiada może być problematyczne oraz czasochłonne.

Jednym z popularniejszych rozwiązań algorytmu k -najbliższych sąsiadów jest FLANN (ang. *Fast Library for Approximate Nearest Neighbors*) [21]. Biblioteka ta bazuje na dwóch podstawowych strukturach danych: losowych k -wymiarowych drzewach (ang. *Randomized kd-trees*) oraz hierarchicznych drzewach k -średnich (ang. *Hierarchical k-means Trees*), jednocześnie oferując mechanizmy do automatycznego dopasowania struktury do domeny danego problemu oraz doboru (pseudo) optymalnych parametrów. Warto przypomnieć, że autorzy FLANN opracowali również rozwiązanie przystosowane do wyszukiwania dopasowań dla deskryptorów binarnych [22].

2.3. Oszacowanie transformacji

Wybranie co najmniej trzech dopasowań między punktami należącymi do danej pary widoków umożliwia wyznaczenie transformacji (tj. translacji oraz rotacji) między tymi widokami. Pojawiają się jednak dwa problemy. Po pierwsze – należy przyjąć, iż część ze znalezionych w poprzednim kroku dopa-



Rys. 5. Dopasowania pozostałe po odrzuceniu dopasowań odstających za pomocą algorytmu RANSAC

Fig. 5. The remaining correspondences after filtering with the RANSAC algorithm

sowań jest nieprawidłowa. Po drugie – kartezjańskie pozycje punktów obliczone na podstawie map głębi obarczone są szumem pomiarowym czujnika. Dlatego też niezbędne jest odrzucenie dopasowań odstających i obliczenie transformacji na podstawie dopasowań przystających.

Jednym z klasycznych rozwiązań tego problemu jest iteracyjny algorytm RANSAC (ang. *RANdom SAMple Consensus*) [10]. W każdej iteracji tego algorytmu obliczana jest transformacja między losowo wybranymi trzema dopasowaniami. Następnie sprawdzane są pozostałe dopasowania pod kątem przystawiania do znalezionej, hipotetycznej transformacji. Sprawdzenie to powoduje utworzenie dwóch zbiorów: pasujących do transformacji (przystających – ang. *inliers*) i niepasujących (odstających – ang. *outliers*). Powyższa procedura powtarzana jest zadaną liczbę iteracji lub dopóki błąd znalezionej transformacji nie będzie poniżej zadanego progu. Przykład dopasowań wyznaczonych za pomocą algorytmu FLANN pokazano na rys. 4, natomiast na rys. 5 przedstawiono dopasowania pozostałe po odfiltrowaniu za pomocą algorytmu RANSAC.

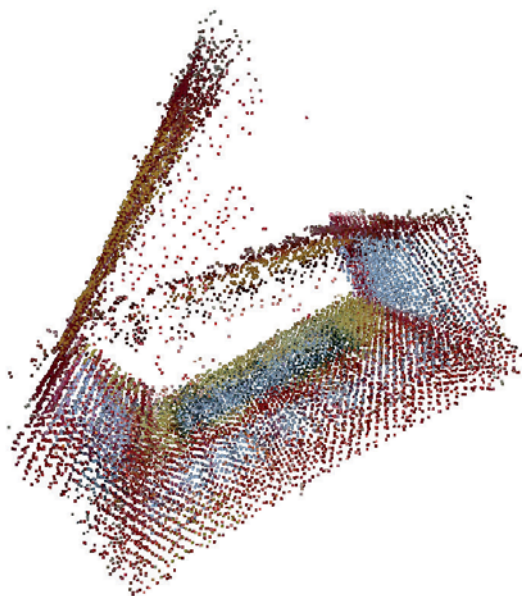
Oczywiście oprócz RANSAC istnieje szereg innych, alternatywnych rozwiązań problemu oszacowania transformacji, jak również szereg ulepszeń samego RANSAC. Zainteresowanego czytelnika odsyłamy do pracy [26], gdzie poza przeglądem różnych rozwiązań autorzy zaproponowali USAC (ang. *Universal RANSAC*), będący rozszerzeniem podstawowego algorytmu RANSAC o szereg praktycznych usprawnień.

2.4. Poprawa transformacji

Wyznaczona w opisany sposób transformacja wstępna jest zwykle bardzo zgrubna, dlatego też z reguły musi być poprawiana. W tym celu zazwyczaj stosowany jest algorytm Iterative Closest Point (ICP) [3, 7, 34]. Główną ideą tego algorytmu jest iteracyjne poprawianie oszacowanej transformacji przez ponowne znalezienie dopasowania punktów pochodzących z dwóch różnych chmur (jednak w tym przypadku mogą to być również gęste chmury punktów) oraz minimalizację odległości między nimi. Ponieważ zarówno samo dopasowywanie punktów może przebiegać według różnych reguł, jak i różne mogą być miary odległości, w przeszłości zaproponowano szereg ciekawych wersji algorytmu ICP, z których część wykorzystuje nie tylko informacje o położeniu punktów w przestrzeni kartezjańskiej, ale również dodatkowe informacje o punktach, np. wektory normalne [6] czy kolor [19]. Szczegółowy opis algorytmu ICP umieszczono w sekcji 3.

2.5. Domknięcie pętli

Po zakończeniu dopasowywania dla każdego widoku i połączeniu wszystkich widoków w jeden model może się okazać, że nie jest on spójny globalnie. Wynika to z faktu, iż w każdym kroku porównywane są tylko dwie kolejne chmury, co sprawia, że kumuluje się błąd dopasowania, widoczny głównie w tzw.



Rys. 6. Przykład niedomkniętej pętli modelu
 Fig. 6. An example of registered model with loop closure step missing

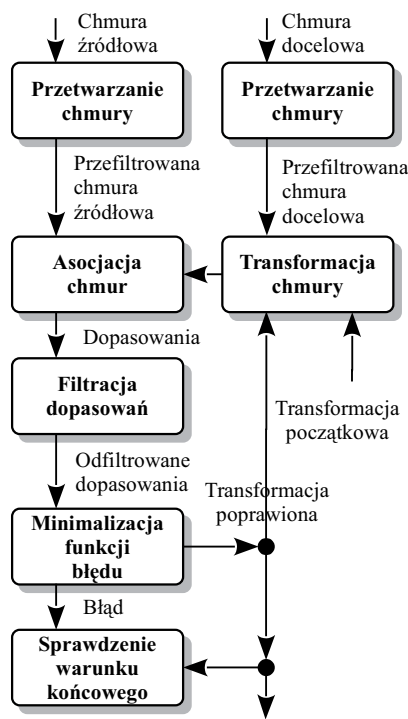
niedomkniętej pętli – w momencie, w którym dany widok z sekwencji ponownie będzie zawierał ten sam fragment sceny lub modelu. Przykład niedomkniętej pętli modelu pokazano na rys. 6. Dla poprawienia spójności można skorzystać z tzw. algorytmów domykających pętle. Algorytmy te równocześnie poprawiają współrzędne punktów kolejnych widoków obraz estymację względnego ruchu kamery. Typowym podejściem jest wykorzystanie globalnej optymalizacji za pomocą grafów. Wierzchołki grafu reprezentują pozycje czujników skojarzone z poszczególnymi widokami. Krawędzie natomiast reprezentują ograniczenia/powiązania i tworzone są tylko wtedy, gdy znaleziono dopasowania między danymi dwoma chmurami. Dla zwiększenia efektywności podczas optymalizacji z reguły brane są pod uwagę jedynie rzadkie chmury cech. Utworzony w ten sposób graf jest optymalizowany w celu znalezienia zbioru najlepszych znalezionych transformacji między zapamiętanymi chmurami. Przykładowe rozwiązania bazujące na tej idei to optymalizacja „TORO” [12] lub optymalizacja „g2o” zaprezentowana w [9].

3. Iterative Closest Point

Nazwa *Iterative Closest Point* pierwszy raz została podana w pracy [3], jednak sam problem rejestracji obrazów był poruszany wcześniej [7]. Od tego czasu powstał szereg prac opisujących różne modyfikacje i udoskonalenia tego algorytmu. W celu uporządkowania głównych kroków algorytmu skorzystano z taksonomii zaproponowanej w pracy [27], połączonej z krokami opisanymi w [25]. Opracowany diagram przepływu danych dla algorytmu ICP pokazano na rys. 7.

Danymi wejściowymi algorytmu ICP są dwie chmury punktów: chmura źródłowa (ang. *Source Point Cloud*), do której będziemy dopasowywać, oraz nowo wczytana chmura docelowa (ang. *Target Point Cloud*), która będzie dopasowywana. W szczególności chmura odniesienia może być chmurą sklejoną z poprzednio wczytanych chmur (np. [8]).

Po wczytaniu obie chmury mogą być poddane przetworzeniu. Operacja ta może mieć na celu pozbycie się szumu, zakładamy dodatkowo, że w kroku tym może być zrealizowane wzbogacenie chmury o dodatkową informację, np. obliczenie dodatkowych cech geometrycznych dla każdego punktu typu wektory normalne.



Rys. 7. Diagram przepływu danych algorytmu ICP
 Fig. 7. Data flow diagram of ICP algorithm

Chmura źródłowa poddawana jest następnie transformacji – w pierwszej iteracji wykorzystywana jest transformacja zadana z zewnątrz (może to być np. transformacja wstępna znaleziona za pomocą wizualnej odometrii lub macierz jednostkowa), w pozostałych iteracjach wykorzystywana jest transformacja wyznaczona w poprzedniej iteracji.

W kolejnym kroku realizowana jest asocjacja dopasowań między punktami chmury źródłowej oraz docelowej. Asocjacja ta polega na sparowaniu punktów obu chmur według zadanego kryterium, tzn. na znalezieniu dla danego punktu jednej chmury najbliższego sąsiada drugiej chmury, według pewnej ustalonej metryki, na przykład najmniejszej odległości euklidesowej między ich współrzędnymi w przestrzeni. Parowanie to realizowane jest w każdym kroku, stąd też znalezione dopasowania w każdej iteracji algorytmu mogą być różne. Warto zaznaczyć tutaj jest, iż podczas asocjacji można wykorzystać nie tylko współrzędne punktu, ale również wektory normalne, kolor czy wspomniane wcześniej deskryptory.

Znalezione dopasowania mogą być następnie poddane różnego rodzaju filtracjom w celu odrzucenie dopasowań, które nie pasują do pozostałych (odstających).

Odfiltrowane dopasowania są następnie wykorzystywane w minimalizacji funkcji błędu. Funkcja błędu może być minimalizowana według innego kryterium niż te wykorzystane do znalezienia dopasowań. Po sprawdzeniu warunku końcowego, algorytm kończy swoje działanie lub rozpoczyna się kolejna iteracja. Warunkami końcowymi algorytmu ICP mogą być:

- maksymalna liczba iteracji,
- minimalna różnica między kolejnymi wyznaczonymi transformacjami,
- minimalna suma odległości między punktami.

4. Podsumowanie

Niniejsza, pierwsza część artykułu miała charakter przeglądowy i jej celem było wprowadzenie czytelnika do problemu rejestracji chmur punktów, począwszy od ogólnego opisu działania i struktury systemu, przez przybliżenie roli poszczególnych

komponentów i wykorzystywanych w nich algorytmów, aż po szczegółowy opis algorytmu ICP przeznaczonego do iteracyjnej rejestracji wzajemnej. Druga część artykułu będzie poświęcona asocjacji punktów należących do różnych chmur, w szczególności na wykorzystywanych podczas asocjacji metrykach. Zaprezentowane zostaną także wyniki eksperymentów, których celem jest porównanie różnych odmian algorytmu ICP.

Bibliografia

- Alahi A., Ortiz R., Vandergheynst P., *FREAK: Fast Retina Keypoint*. 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 510–517, IEEE, 2012.
- Belter D., Łabęcki P., Fankhauser P., Siegwart R., *RGB-D terrain perception and dense mapping for legged robots*. “International Journal of Applied Mathematics and Computer Science”, 26(1):81–97, 2016.
- Besl P., McKay N., *A method for registration of 3-D shapes*. “IEEE Transactions on Pattern Analysis and Machine Intelligence”, 14(2):239–256, 1992.
- Cadena C., Carlone L., Carrillo H., Latif Y., Scaramuzza D., Neira J., Reid I., Leonard J.J., *Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age*. “IEEE Transactions on Robotics”, 32(6):1309–1332, 2016.
- Calonder M., Lepetit V., Strecha C., Fua P., *BRIEF: Binary Robust Independent Elementary Features*. Computer Vision – ECCV 2010, 778–792, Springer, 2010.
- Censi A., *An ICP variant using a point-to-line metric*. 2008 IEEE International Conference on Robotics and Automation, ICRA 2008, 19–25, DOI: 10.1109/ROBOT.2008.4543181.
- Chen Y., Medioni G., *Object modelling by registration of multiple range images*, 1991 IEEE International Conference on Robotics and Automation, Proceedings Vol. 3, 2724–2729.
- Dryanovski I., Valenti R., Xiao J., *Fast visual odometry and mapping from RGB-D data*. 2013 IEEE International Conference on Robotics and Automation (ICRA), 2305–2310, DOI: 10.1109/ICRA.2013.6630889.
- Endres F., Hess J., Engelhard N., Sturm J., Cremers D., Burgard W., *An evaluation of the RGB-D SLAM system*. 2012 IEEE International Conference on Robotics and Automation (ICRA), 1691–1696, May 2012.
- Fischler M.A., Bolles R.C., *Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography*. “Communications of the ACM”, 24(6):381–395, 1981, DOI: 10.1145/358669.358692.
- Fraundorfer F., Scaramuzza D., *Visual odometry: Part II: Matching, robustness, optimization, and applications*. “IEEE Robotics & Automation Magazine”, 19(2):78–90, 2012, DOI: 10.1109/MRA.2012.2182810.
- Henry P., Krainin M., Herbst E., Ren X., Fox D., *RGB-D Mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments*. “International Journal of Robotics Research”, 31(5):647–663, 2012.
- Holz D., Ichim A.E., Tombari F., Rusu R.B., Behnke S., *Registration with the point cloud library – A modular framework for aligning in 3-D*. “IEEE Robotics & Automation Magazine”, 22(4):110–124, 2015, DOI: 10.1109/MRA.2015.2432331.
- Kornuta T., Laszkowski M., *Perception subsystem for object recognition and pose estimation in RGB-D images*. Challenges in Automation, Robotics and Measurement Techniques, 597–607. Springer, 2016, DOI: 10.1007/978-3-319-29357-8_52.
- Kornuta T., Stefanczyk M., *Utilization of textured stereo-vision for registration of 3D models of objects*. 21st International Conference on Methods and Models in Automation and Robotics (MMAR), 2016, 1088–1093. IEEE, DOI: 10.1109/MMAR.2016.7575289.
- Kornuta T., Stefanczyk M., *Akwizycja obrazów RGB-D: czujniki*. „Pomiary Automatyka Robotyka”, 18(2):92–99, 2014.
- Leutenegger S., Chli M., Siegwart R., *BRISK: Binary Robust invariant scalable keypoints*. 2011 IEEE International Conference on Computer Vision (ICCV), 2548–2555, DOI: 10.1109/ICCV.2011.6126542.
- Lowe D., *Object recognition from local scale-invariant features*. The Proceedings of the Seventh IEEE International Conference on Computer Vision, 1999, Vol. 2, 1150–1157.
- Men H., Gebre B., Pochiraju K., *Color point cloud registration with 4D ICP algorithm*. 2011 IEEE International Conference on Robotics and Automation (ICRA), 1511–1516, DOI: 10.1109/ICRA.2011.5980407.
- Moravec H.P., *Obstacle avoidance and navigation in the real world by a seeing robot rover*. Memo AIM-340, Stanford Artificial Intelligence Laboratory, 1980.
- Muja M., Lowe D.G., *Fast approximate nearest neighbors with automatic algorithm configuration*. International Conference on Computer Vision Theory and Application VSSAPP '09, 331–340. INSTICC Press, 2009.
- Muja M., Lowe D.G., *Fast matching of binary features*. 2012 Ninth Conference on Computer and Robot Vision (CRV), 404–410. IEEE, 2012.
- Nistér D., Naroditsky O., Bergen J., *Visual odometry*. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Vol. 1, 1–8. IEEE, 2004.
- Pomerleau F., Colas F., Siegwart R., *A review of point cloud registration algorithms for mobile robotics*. Foundations and Trends in Robotics, 4(1):1–104, 2015, DOI: 10.1561/23000000035.
- Pomerleau F., Colas F., Siegwart R., Magnenat S., *Comparing ICP variants on real-world data sets*. “Autonomous Robots”, 34(3):133–148, 2013, DOI: 10.1007/s10514-013-9327-2.
- Raguram R., Chum O., Pollefeys M., Matas J., Frahm J.-M., *USAC: a universal framework for random sample consensus*. IEEE transactions on pattern analysis and machine intelligence, 35(8):2022–2038, 2013, DOI: 10.1109/TPAMI.2012.257.
- Rusinkiewicz S., Levoy M., *Efficient variants of the ICP algorithm*. Third International Conference on 3-D Digital Imaging and Modeling, Proceedings, 145–152, IEEE, 2001, DOI: 10.1109/IM.2001.924423.
- Rusu R.B., Blodow N., Beetz M., *Fast point feature histograms (FPFH) for 3D registration*. IEEE International Conference on Robotics and Automation, ICRA '09, 3212–3217. IEEE, 2009, 10.1109/ROBOT.2009.5152473.
- Scaramuzza D., Fraundorfer F., *Visual odometry: Part I: The first 30 years and fundamentals*. IEEE Robotics & Automation Magazine, 18(4):80–92, 2011.
- Steder B., Rusu R.B., Konolige K., Burgard W., *Point feature extraction on 3D range scans taking into account object boundaries*. 2011 IEEE International Conference on Robotics and Automation (ICRA), 2601–2608. IEEE, 2011, DOI: 10.1109/ICRA.2011.5980187.
- Stefanczyk M., Kornuta T., *Akwizycja obrazów RGB-D: metody*. „Pomiary Automatyka Robotyka”, 18(1):82–90, 2014.
- Thrun S., Leonard J.J., *The Handbook of Robotics*, rozdział Simultaneous Localization and Mapping, 871–890. Springer, June 2008.
- Tombari F., Salti S., Di Stefano L., *Unique signatures of histograms for local surface description*. Proceedings of the 11th European Conference on Computer Vision Conference on Computer Vision: Part III, ECCV '10, 356–369, Berlin, Heidelberg, 2010. Springer-Verlag.
- Zhang Z., *Iterative point matching for registration of free-form curves and surfaces*. “International Journal of Computer Vision”, 13(2):119–152, 1994.

Registration of RGB-D Images: Components of the System

Abstract: The two-part article focuses on the problem of registration of RGB-D images, a problem that in the robotics domain is known as Visual Simultaneous Localization and Mapping, or V-SLAM in short. The following, first part of the article presents a bird's eye view on the main components of V-SLAM systems and focuses on the ICP (Iterative Closest Point), an algorithm for a pairwise registration of point clouds. In the second part we present different types of attributes of points that can be used during the association step along with different metrics that operate on those attributes and that can be employed during the registration. We also describe the methodology used in the conducted experiments and discuss the results of comparison of selected flavours of ICP.

Keywords: RGB-D image, point cloud, V-SLAM, pairwise registration, ICP

dr inż. Tomasz Kornuta

tkornuta@gmail.com

Absolwent Wydziału Elektroniki i Technik Informatycznych Politechniki Warszawskiej. W 2003 r. uzyskał tytuł inżyniera, w 2005 r. tytuł magistra inżyniera, a w 2013 r. stopień doktora nauk technicznych. W latach 2008–2015 pracował w Instytucie Automatyki i Informatyki Stosowanej, początkowo jako asystent, potem jako adiunkt, dodatkowo od 2009 r. pełni funkcję kierownika Laboratorium Podstaw Robotyki. Od października 2015 r. przebywa na stażu post doktorskim w ośrodku IBM Research – Almaden, prowadząc badania nad wykorzystaniem technik sztucznej inteligencji i uczenia maszynowego w percepcji wizyjnej. Jego zainteresowania naukowe obejmują metody programowania robotów, architektury układów sterowania robotów, percepcję wizyjną (w szczególności trójwymiarową z wykorzystaniem czujników RGB-D) oraz sztuczną inteligencję i uczenie maszynowe. Autor/współautor 70+ publikacji dotyczących powyższych tematów. Organizator szeregu sesji specjalnych poświęconych tym tematom, recenzent krajowych oraz międzynarodowych konferencji i czasopism. Uczestnik kilkunastu projektów badawczych, w tym dwóch grantów europejskich oraz grantu finansowanego przez agencję DARPA.



mgr inż. Marta Jolanta Łepicka

marta.lepicka@gmail.com

Absolwentka kierunku Informatyka na Wydziale Elektroniki i Technik Informatycznych Politechniki Warszawskiej. W ramach prac inżynierskiej i magisterskiej zajmowała się badaniami nad algorytmami łączenia chmur punktów wykorzystującymi zarówno informację geometryczną (kształt), jak i fotometryczną (tekstura). Jej aktywność naukowa w trakcie studiów zaowocowała udziałem w szeregu grantów badawczych oraz otrzymaniem stypendium Rektora Politechniki Warszawskiej za Wyjątkowe Osiągnięcia Naukowe, a także dwoma Głównymi Nagrodami w organizowanym przez Instytut PIAP ogólnopolskim konkursie „Młodzi Innowacyjni”: w ramach kategorii „Najlepsza Praca Dyplomowa Inżynierska” w 2015 r. oraz „Najlepsza Praca Dyplomowa Magisterska” w 2017 r. Jej zainteresowania naukowe obejmują wizję komputerową, przetwarzanie obrazów, uczenie maszynowe oraz systemy inteligentne.

