

## TEKSTUROWANIE MODELI OBIEKTÓW O ZŁOŻONEJ GEOMETRII NA PODSTAWIE DANYCH Z NAZIMENEGO SKANINGU LASEROWEGO

### COMPLEX OBJECTS TEXTURING BASED ON TERRESTRIAL LASER SCANNER DATA

Jakub Kolecki, Małgorzata Słota

AGH w Krakowie,  
Katedra Geoinformacji, Fotogrametrii i Teledetekcji Środowiska

SŁOWA KLUCZOWE: naziemny skaningu laserowy, przetwarzanie chmury punktów, modelowanie 3D, inwentaryzacja, teksturowanie

STRESZCZENIE: Obrazy pozyskane podczas rejestrowanej w trakcie skanowania laserowego chmury punktów pozwalają na tworzenie tekstur modelowanego obiektu, co zwiększa jego zawartość informacyjną. Generowanie tekstur bezpośrednio z pozyskanych obrazów wymaga znajomości parametrów odwzorowania, w którym powstaje zdjęcie. Jednak także informacja o kolorze zapisana jako atrybuty punktów chmury może być wykorzystana do tworzenia tekstur modelowanego obiektu. W takim przypadku chmura punktów może pośredniczyć w tworzeniu obrazów tekstur a cały proces teksturowania odbywa się bez bezpośredniego udziału zdjęć. Celem niniejszych badań było opracowanie metody teksturowania modeli obiektów o złożonej geometrii na podstawie kolorowej chmury punktów pochodzącej z naziemnego skaningu laserowego. Wynikiem pracy jest autorski program do tworzenia tekstur, bezpośrednio w oparciu o kolory RGB chmury punktów. Danymi wejściowymi do programu są chmury punktów w formacie tekstowym oraz obiekty 3D w formacie VRML. W celu przyspieszenia obliczeń w pierwszym kroku wykonywana jest automatyczna segmentacja chmur punktów. Następnie pozyskiwana jest informacja o geometrii płaszczyzn obiektu na podstawie pliku VRML. W efekcie analiz przestrzennych pomiędzy położeniem pikseli na teksturowanych płaszczyznach i chmurą punktów, pozyskiwane są informacje o kolorze pikseli oraz tworzone są tekstury obiektu. Uzyskane wyniki pokazują, że tworzone w ramach prac testowych tekstury mogą posiadać artefakty, będące efektem niedopasowania radiometrycznego zdjęć pozyskiwanych z różnych stanowisk skanowania.

## 1. WPROWADZENIE

### 1.1. Teksturowanie modeli 3D

Obecnie standardem przedstawiania wyników prac fotogrametrycznych stają się teksturowane modele 3D. Wszystko zaczęło się od prostych wizualizacji 2.5D niewielkich obiektów ze sztucznie wygenerowanymi teksturami, imitującymi rzeczywisty wygląd powierzchni. Obecnie wraz z rozwojem systemów pomiarowych, zwiększyły się

oczekiwania co do produktów inwentaryzacji obiektów. Zamawiający oprócz wysokiej dokładności pomiarów cenią sobie jakość wiernej, najlepiej multimedialnej prezentacji wyników. Coraz więcej badań poświęconych jest aktualnie tworzeniu fotorealistycznych wizualizacji obiektów, składających się z milionów punktów, w czasie rzeczywistym (Hanusch, 2008).

W grafice komputerowej istnieje wiele metod fotorealistycznego wyświetlania obiektów 3D. Przedstawiane obiekty reprezentowane są zazwyczaj w postaci siatki trójkątów i/lub w postaci modelu typu CAD, składającego się z trójwymiarowych obiektów elementarnych tj. płaszczyzny. Nakładanie na trójwymiarowy obiekt fotorealistycznej lub sztucznej (syntetycznej) tekstury nosi nazwę „teksturowania”. Tekstura służy więc do odwzorowania rzeczywistego wyglądu obiektów za pomocą technik komputerowych. Teksturowanie umożliwia przedstawienie zmiennego koloru modelowanej powierzchni jak również umożliwia wizualizację skomplikowanych detali obiektu bez potrzeby szczegółowego modelowania, co znacznie wydłużyłoby proces opracowywania wyników.

Proces teksturowania obiektów 3D można przeprowadzić ręcznie lub automatycznie. W teksturowaniu manualnym użytkownik sam musi wskazać odpowiadające sobie punkty na obrazie jak i na teksturowanym obiekcie. Takie podejście jest zaimplementowane między innymi w programach Maxon Cinema 4D, Autodesk 3Ds Max czy Maya. Teksturowanie automatyczne polega na rzutowaniu obrazu na obiekt z wykorzystaniem równania kolinearności. W automatycznym podejściu elementy orientacji wewnętrznej i zewnętrznej wszystkich zdjęć muszą być znane w układzie współrzędnych tożsamym z układem obiektu. W celu uzyskania najlepszej dokładności geometrycznej teksturowania, w automatycznej metodzie wykorzystuje się najczęściej matematyczne modele kamery obliczone w procesie ścisłego wyrównania sieci zdjęć (Kersten *et al.*, 2012).

## 1.2. Dotychczasowe prace

W literaturze spotkać można bardzo dużo prac poświęconych rekonstrukcji i modelowaniu trójwymiarowych obiektów na podstawie danych fotogrametrycznych jak również danych pochodzących z laserowego skaningu naziemnego. W coraz większej ilości publikacji poruszane są kwestie teksturowania. Beraldin (Beraldin *et al.*, 2002) oraz Kadobayashi (Kadobayashi *et al.*, 2004) w swoich badaniach wykorzystują zdjęcia o znanych elementach orientacji w celu teksturowania trójwymiarowych modeli pozyskanych na podstawie danych ze skaningu laserowego. Przegląd technik teksturowania na podstawie zdjęć wraz z przekrojem przez różne programy do tworzenia fotorealistycznych tekstur prezentuje Kersten (Kersten *et al.*, 2012). W publikacji Hanuscha (Hanusch, 2008) zaprezentowano rozszerzenie metody teksturowania z 2.5D do 3D. Przeprowadzono wektorową analizę widoczności i zaproponowano algorytm do teksturowania obiektów na podstawie wielu obrazów o różnej rozdzielczości i różnych poziomach jasności. Problem teksturowania modeli z naziemnego skaningu laserowego na podstawie niemetrycznych zdjęć został opisany przez Hu (Hu *et al.*, 2008) – jako rozwiązanie proponowane jest wyznaczenie elementów orientacji zdjęć na podstawie punktów wiążących chmurę punktów i zdjęcia. Ciekawą iteracyjną metodę do teksturowania chmury punktów, wykorzystującą drzewo KD do tworzenia zależności sąsiedztwa pomiędzy punktami oraz pozwalającą na ominięcie etapu budowania siatki TIN

przedstawia Zeng (Zeng *et al.*, 2011). Zupełnie inne podejście, podobne do proponowanego w tym artykule, przedstawia Oniga (Oniga, 2012), która prezentuje półautomatyczną metodę teksturowania modeli 3D na podstawie kolorowej chmury punktów. Do każdego punktu z chmury przypisywana jest płaszczyzna, do której odległość od punktu jest najmniejsza. Następnie odrzucane są punkty, które leżą w większej niż zdefiniowana przez użytkownika odległości od płaszczyzny. Kolejnym krokiem jest budowa diagramu Voronoi dla zakwalifikowanych do płaszczyzny punktów i przypisywanie koloru do odpowiednich poligonów Voronoi z chmury punktów.

Większość proponowanych metod teksturowania zakłada istnienie modelu 3D w postaci siatki trójkątów oraz znanych lub wyznaczanych elementów orientacji zdjęć. Obecnie na rynku istnieją programy do automatycznego modelowania i teksturowania na podstawie kolorowej chmury punktów (np. Geomagic), lecz w wyniku działania zaimplementowanych algorytmów tworzona jest najczęściej bardzo gęsta siatka TIN. Większość analiz przestrzennych i baz danych zorientowanych obiektowo nie aprobuje takiego typu reprezentacji obiektów 3D. Ponieważ sama idea generowania tekstur zakłada minimalizację złożoności modelu bez wizualnej utraty detali obiektu, automatyczne teksturowanie bardzo złożonych modeli TIN (opartych często bezpośrednio na milionach punktów ze skaningu) wydaje się zaprzeczać idei teksturowania.

## **2. BEZPOŚREDNIE WYKORZYSTANIE ZDJĘĆ DO GENEROWANIA OBRAZÓW TEKSTUR – POTENCJALNE PROBLEMY**

W praktyce teksturowanie gotowych modeli obiektów 3D realizowane jest z wykorzystaniem zapisanych w trakcie skanowania obrazów. Aby mogło się ono odbywać automatycznie, wymagane jest określenie odwzorowania pomiędzy punktami przestrzeni obiektu a punktem leżącym na zdjęciu. Najczęściej system skanujący określa elementy orientacji zewnętrznej zdjęć znając wzajemną orientację kamery w układzie skanera. Jednak w pewnych okolicznościach generowanie obrazów tekstur na podstawie zarejestrowanych przez skaner zdjęć może stanowić pewien problem w produkcji modelu skanowanego obiektu. Wymienić tu można następujące przypadki:

- użytkownik skanera nie ma dostępu do elementów orientacji zewnętrznej zdjęć, a ich wyznaczenie w oparciu o rozwiązania analityczne (np. wcięcie wstecz) jest zbyt pracochłonne.
- geometria chmury punktów została zmieniona w wyniku transformacji (obrót, przesunięcie), tak że elementy orientacji zewnętrznej zdjęć w układzie obiektu nie są już znane.

Oprogramowanie do kompleksowego opracowania danych otrzymanych w wyniku skanowania najczęściej dostosowane jest do pracy z konkretnym typem skanera i zazwyczaj umożliwia znaczną automatyzację procesu teksturowania modelu 3D obiektu powstałego na podstawie chmury punktów, wykorzystując do tego celu zdjęcia. Zaznaczyć jednak należy, że na rynku dostępnych jest relatywnie dużo programów umożliwiających tworzenie modeli 3D obiektów na podstawie chmur punktów. Programy te nie posiadają jednak możliwości generowania tekstur. Problem teksturowania mógłby być jednak rozwiązany przez zewnętrzną aplikację, realizującą algorytm generowania tekstur w sposób pośredni, wykorzystując pokolorowaną chmurę punktów.

### 3. ALGORYTM TEKSTUROWANIA BAZUJĄCEGO NA PUNKTACH

Próba rozwiązania wyżej wymienionych problemów jest stworzenie programu generującego tekstury obiektów, wykorzystującego wprost informację o składowych R, G, B zapisaną łącznie ze współrzędnymi punktów chmury. Możliwość eksportu gotowych modeli 3D do popularnych formatów (języków) modelowania (VRML, OBJ) podobnie jak możliwość eksportu chmur punktów do otwartych formatów zapisu sprawia, że program taki może działać niezależnie od dostępnych już zaawansowanych pakietów do tworzenia modeli i pracy z danymi pochodzącymi ze skaningu.

Algorytm generowania obrazów tekstur na podstawie chmur punktów z informacją o kolorze (R, G, B) jest w znacznej części zbieżny z typowym algorytmem generowania tekstur na podstawie zdjęć cyfrowych. W pierwszej kolejności określana jest transformacja z lokalnego układu wieloboku modelu do układu terenowego. Następnie określić należy transformację z układu rastra (wiersz i kolumna) tekstury do układu teksturowanego wieloboku. Parametry tej transformacji uzależnione są od rozdzielczości rastra. Najczęściej wymiary obrazu tekstury stanowią potęgę liczby 2 (np.  $256 \times 256$  pikseli), co jest związane z wymogami kart graficznych. Kolejnym etapem algorytmu jest interpolacja jasności dla każdego z 3 kanałów rastra (R, G, B) w oparciu o kolory punktów chmury. Dla danego piksela (wiersz, kolumna) obliczane są zatem jego współrzędne w lokalnym układzie wieloboku i następnie współrzędne terenowe. Kolejnym etapem jest interpolacja jasności na podstawie nieregularnie rozmieszczonych punktów chmury leżących w sąsiedztwie punktu o obliczonych współrzędnych. Można wykorzystać tu interpolację metodą najbliższego sąsiada lub interpolować jasność z danych nieregularnych np. obliczać wartości jasności w każdym z trzech kanałów jako średnie ważone z jasności kilku punktów leżących w pobliżu. W badanej implementacji algorytmu wartości składowych barwnych interpolowane były jako średnie ważone ze składowych przypisanych do trzech najbliższych punktów chmury.

Krytycznym z punktu widzenia czasochłonności proponowanego algorytmu jest sposób implementacji wyszukiwania najbliższego danemu pikselowi tekstury punktu w chmurze. Aby skrócić czas przeszukiwania, chmura punktów dzielona jest na mniejsze chmury. Każda z takich chmur zawiera punkty leżące wewnątrz sześciangu o wymiarach  $1 \times 1 \times 1$  m (woksela). Przeszukiwanie odbywać się może teraz jedynie w obrębie chmury leżącej w sześciangu, w którym leży również dany piksel tekstury.

Przedstawiony algorytm zaimplementowano w języku C++. Do obsługi plików graficznych wykorzystano bibliotekę OpenCV (OpenCV, 2012). Zbudowano interfejs graficzny programu. Program wczytuje chmurę punktów w postaci pliku tekstowego zawierającego współrzędne oraz składowe R, G, B. W celu szybszego przetwarzania chmura ta zamieniana jest na postać binarną. Istnieje możliwość ustalenia rozdzielczości obrazu tekstury na poziomach od  $128 \times 128$  do  $1024 \times 1024$  piksele.

Kolorowanie chmury punktów odbywa się na podstawie zdjęć wykonywanych przez system skanujący. Generowanie tekstur na podstawie barw rozumianych jako atrybuty chmury punktów wiąże się zatem z wtórnym wykorzystaniem informacji o kolorze. Wartości składowych barwnych poszczególnych pikseli tekstury mogą zatem być wyznaczone na podstawie nie jednego ale kilku zdjęć. Może to powodować zaszumienie

obrazu tekstury, gdyż każde ze zdjęć wykonywane jest podczas innych warunków oświetleniowych.

Kolejnym problemem proponowanego rozwiązania jest konieczność implementacji algorytmu interpolacji wartości jasności z nieregularnego zbioru danych, jakim jest chmura punktów. W przypadku generowania tekstur w oparciu o zdjęcia mamy do czynienia z interpolacją jasności w siatce regularnej, które jest algorytmicznie prostsze do realizacji (np. metoda biliniowa) i szybsze w działaniu. W odróżnieniu od interpolacji w siatce regularnej, interpolacja z nieregularnego zbioru danych wymaga odszukania punktów leżących w sąsiedztwie danego piksela tekstury. Wymaga to wdrożenia algorytmów indeksowania chmury punktów, tak by skrócić czas jej przeszukiwania.

#### **4. PRACE EKSPERYMENTALNE**

##### **4.1. Dane – chmura punktów**

Chmurę punktów pozyskano dla dźwigu portowego o wysokości ok. 30 m (rys. 1). W wyniku pomiarów wykonanych za pomocą naziemnego skanera laserowego FARO Focus 3D pozyskano osiem chmur punktów, które zostały przetransformowane do wspólnego, lokalnego układu współrzędnych oraz „pokolorowane” w oprogramowaniu FARO Scene.

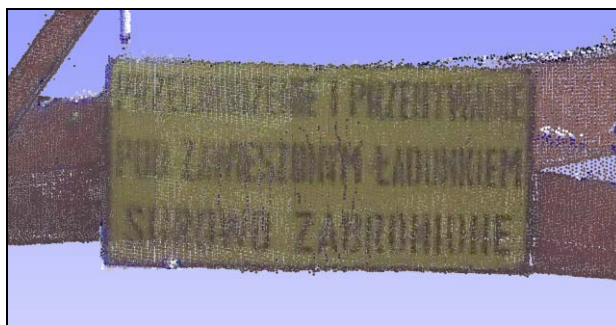


Rys. 1. Połączone i pokolorowane chmury punktów skanowanego dźwigu

##### **4.2. Testy**

W celu sprawdzenia działania algorytmu utworzono dwa wieloboki składające się na model obiektu testowego. Pierwszy wielobok utworzono na drodze wpasowania płaszczyzny w chmurę punktów reprezentującą tablicę, przedstawioną na rysunku 2. Drugi wielobok utworzono w podobny sposób dla ściany widocznej na rysunku 3. Proces

modelowania przeprowadzono w oprogramowaniu Leica Cyclone v. 7.2, a następnie model został wyeksportowany do formatu VRML.



Rys. 2. Widok pokolorowanej chmury punktów reprezentującej pierwszy obiekt testowy



Rys. 3. Widok pokolorowanej chmury punktów reprezentującej drugi obiekt testowy

Dla obu obiektów testowych wygenerowano obrazy tekstur stosując zaproponowany algorytm. Ponieważ obiekty testowe zostały zeskanowane z różnych stanowisk, teksturowanie dla każdego z nich przeprowadzono w dwóch wariantach: wykorzystując połączoną chmurę punktów oraz wykorzystując jedną, wybraną chmurę zawierającą punkty reprezentujące dany obiekt testowy. Rozdzielczości chmur punktów dla poszczególnych obiektów testowych przedstawiono w tabeli 1. Obrazy tekstur tworzono w rozdzielczości od  $128 \times 128$  do  $1024 \times 1024$  pikseli. Wartości wielkości piksela terenowego dla obrazów o różnej rozdzielczości przedstawiono w tabeli 2.

Tabela 1. Średnie odległości sąsiednich punktów w chmurach [mm]

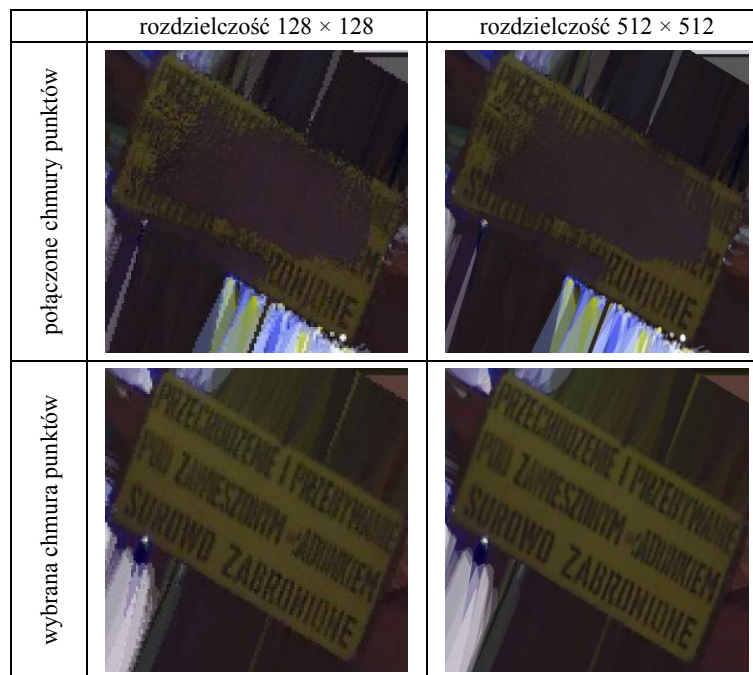
| Obiekt testowy | Połączona chmura punktów | Wybrana chmura punktów |
|----------------|--------------------------|------------------------|
| 1              | 4                        | 7                      |
| 2              | 12                       | 17                     |

Tabela 2. Wielkość piksela terenowego w [mm] dla obrazów tekstur o różnej rozdzielczości

| Obiekt testowy | Rozdzielczość tekstury |           |           |             |
|----------------|------------------------|-----------|-----------|-------------|
|                | 128 × 128              | 256 × 256 | 512 × 512 | 1024 × 1024 |
| 1              | 12.3                   | 6.1       | 3.1       | 1.5         |
| 2              | 31.7                   | 15.9      | 7.9       | 4.0         |

### 4.3. Wyniki – obrazy tekstur

Wyniki przeprowadzonych testów potwierdziły sprawność proponowanego algorytmu jak również wskazały na kilka problemów przy tworzeniu tekstur z chmur punktów. Na rysunkach 4 i 5 pokazano wynikowe obrazy tekstur dla 2 obiektów testowych. Zauważyć można, iż ze wzrostem rozdzielczości generowane tekstury mają lepszą jakość (do momentu, gdy rozdzielczość nie przekracza odległości terenowej pomiędzy punktami skanu). Na rysunku 4 przedstawiono tekstury obiektu o małym przekroju poprzecznym (metalowa tablica). Obiekt ten został zeskanowany z dwóch stron – ponieważ odległość pomiędzy przednią i tylną ścianką tablicy jest bardzo mała, dla części pikseli na obrazie tekstury został przypisany kolor ze skanów zarejestrowanych dla tylnej części tablicy.

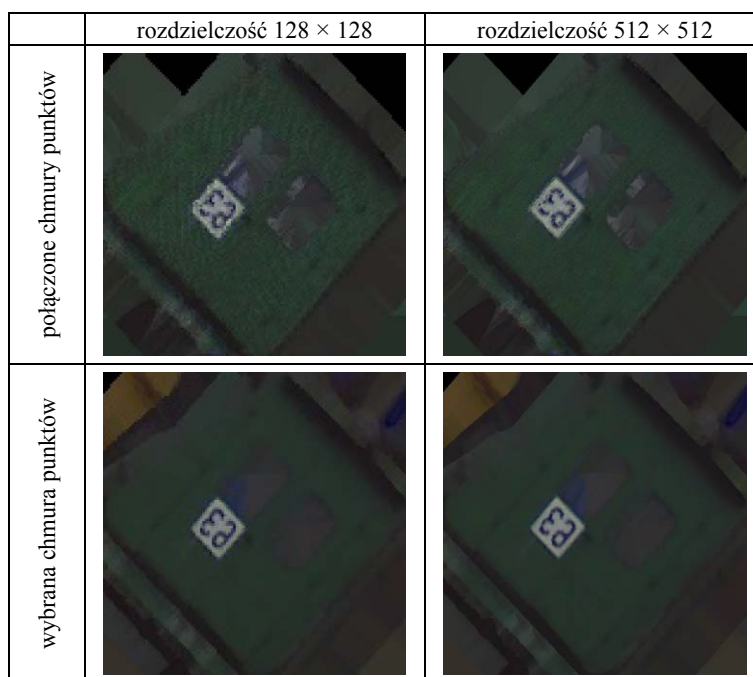


Rys. 4. Obrazy tekstur wygenerowane dla pierwszego obiektu testowego

Najprostszym sposobem na uniknięcie tego błędu jest wykorzystanie w procesie teksturuowania tylko wybranych, odpowiednich chmur punktów. Kolejnym zaobserwowanym w badaniach problemem jest szum, pojawiający się na teksturach wygenerowanych z kilku kolorowych chmur punktów (rys. 5). Ponieważ skany oraz zdjęcia

rejestrowane są z poszczególnych stanowisk w pewnym odstępie czasowym i w nieco innych warunkach oświetlenia, odfotografowane obiekty na zdjęciach, na podstawie których chmura jest „kolorowana”, różnią się kolorystycznie między sobą. W przypadku więc gdy algorytm poszukuje koloru tekstury na połączonej z kilku skanów chmurze punktów, wartości pikseli obrazu mogą być pozyskane z pierwotnie różnych chmur punktów, co generuje szum.

Dodatkowo na przedstawionych rysunkach zauważyć można, iż orientacja tworzonych obrazów tekstur nie odpowiada obecnie optymalnemu rozmieszczeniu tekstury względem teksturowanej płaszczyzny. W przyszłości planuje się kontynuować pracę nad algorytmem, między innymi nad rozwiązaniem problemu orientacji obrazu i metodami, przyspieszającymi obliczenia na dużym zbiorze danych.



Rys. 5. Obrazy tekstur wygenerowane dla drugiego obiektu testowego

#### 4. PODSUMOWANIE

Zaproponowana w tym artykule metoda generowania tekstur na podstawie kolorowej chmury punktów wydaje się być interesującą alternatywą w przypadkach, gdy użytkownik nie ma dostępu do elementów orientacji zdjęć i odtworzenie tych parametrów jest zbyt czasochłonne lub gdy operator nie dysponuje odpowiednim oprogramowaniem do tworzenia teksturowanych modeli 3D na podstawie zdjęć. Oczywiście tekstury powstałe na podstawie kolorowych chmur punktów będą miały niższą jakość od tekstur generowanych bezpośrednio na podstawie zdjęć o wysokiej rozdzielczości, lecz dla wielu zastosowań, np.



GIS, jakość generowanych na podstawie proponowanego algorytmu tekstur jest wystarczająca.

Przedstawiony w artykule algorytm zaimplementowano w języku C++ z wykorzystaniem biblioteki OpenCV. W wyniku prac powstał program, który wczytuje chmurę punktów w postaci pliku tekstowego zawierającego współrzędne oraz składowe R, G, B oraz generuje obrazy tekstur o rozdzielczości zdefiniowanej przez użytkownika (od  $128 \times 128$  do  $1024 \times 1024$  pikseli). Podczas przeprowadzonych testów zwrócono uwagę na problemy, pojawiające się przy tworzeniu tekstur z kilku połączonych chmur punktów m.in. kwestia szumu czy obiektów o niewielkich przekrojach.

Metoda naziemnego skaningu laserowego wraz z proponowanym algorytmem wydają się odpowiednio z punktu widzenia potrzeb systemów GIS i szybkich opracowań 3D.

## 5. LITERATURA

Beraldin J-A., Picard M., El-Hakim SF., Godin G., Valzano V., Bandiera A., Lataouche D., 2002. Virtualizing a Byzantine crypt by combining high-resolution textures with laser scanner 3D data. *Proceedings of VSMM 2002*, Gyeongju, Korea, s. 25-27 września.

Hanusch T., 2008. A new texture mapping algorithm for photorealistic reconstruction of 3D objects. *Int. Archives Photogram. Remote Sensing Spatial Information Science*, XXXVII, B5-2, Pekin, Kongres ISPRS, s. 699-705

Hu C., Wang Y., Yu W., 2008. Mapping Digital Image Texture onto 3D Model from LiDAR data, *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 37(Part B5), s. 611-614.

Kadobayashi, R., Kochi, N., Otani, H. and Furukawa, R., 2004. Comparison and evaluation of laser scanning and Photogrammetry and their Combined Use for Digital Recording of Cultural Heritage. *IAPRS&SIS*, 35(5), İstanbul, Turcja, s. 401-406

Kersten T. P., Stallmann D., 2012. Automatix Texture Mapping of Architectural and Archeological 3D Models, *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Volume XXXIX-B5, 2012 XXII ISPRS Congress, 25 sierpnia – 1 września 2012, Melbourne, Australia.

Oniga E., 2012, A New Approach for the Semi-Automatic Texture Generation of the Buildings Facades from Terrestrial Laser Scanner Data, *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Volume XXXIX-B6, 2012 XXII ISPRS Congress, 25 August - 01 September 2012, Melbourne, Australia.

OpenCV, 2012, <http://opencv.willowgarage.com/wiki/>

Zeng J., Li T., 2011. An Algorithm of 3D Texture Synthesis based on Point Cloud, *Journal of Computational Information Systems* 7:5, s. 1592-1599.

Praca naukowa finansowana ze środków na naukę w latach 2010-2012 jako projekt rozwojowy.

## COMPLEX OBJECTS TEXTURING BASED ON TERRESTRIAL LASER SCANNER DATA

KEY WORDS: terrestrial laser scanning, point cloud processing, 3D modeling, stocktaking, texture mapping

### Summary

Images taken during point cloud acquisition using laser scanning can be subsequently utilized for generating textures of 3D models. As a result the final amount of information associated with produced model is increased. Automatic texture generation using captured images directly, demands the knowledge about parameters describing image projection. However using the information about color stored as point cloud attributes allows texture generation without using the images directly. The addressed researches aim to develop a method of model texturing. As the final result a simple GUI application has been created in C++. Point clouds in text format and VRML models are used as the input data. In order to speed up the calculation process, in the first step the automatic segmentation of the point clouds is performed. Secondly the information about the object surfaces is obtained based on VRML file and then textures are defined. After performing spatial analysis between pixels position on textured surfaces and point cloud, the pixels color information is computed and texture images are generated. The results show that the test objects textures may be affected by noise resulting from radiometric discrepancies between images acquired from different standpoints.

Dane autorów:

Mgr inż. Jakub Kolecki  
e-mail: kolecki@agh.edu.pl  
telefon: 12 617 39 93  
fax: 12 617 39 93

Mgr inż. Małgorzata Słota  
e-mail: mslota@agh.edu.pl  
telefon: 12 617 39 93  
fax: 12 617 39 93