



URB3D

METODA TRÓJWYMIAROWEGO MODELOWANIA OBSZARÓW URBANISTYCZNYCH Z WYKORZYSTANIEM METOD FOTOGRAMETRII



DANIEL BORKOWSKI, JULIA FARGANUS, RAFAŁ MIELNICZUK, KATARZYNA WOCHAL, PROMOTOR MAREK KRÓTKIEWICZ

ABSTRAKT



Projekt obejmował stworzenie aplikacji do modelowania trójwymiarowych scen miejskich na podstawie zdjęć fotogrametrycznych, wykorzystującej metody *structure from motion* i *gaussian splatting* oraz segmentację semantyczną.

Słowa kluczowe: reconstruction, 3D modeling, urban modeling, semantic segmentation, gaussian splatting.

AKWIZYCJA I REKONSTRUKCJA

W projekcie zastosowano metody fotogrametryczne do pozyskania zdjęć z drona i gruntu na terenie Politechniki Wrocławskiej, obejmujących budynki C5, C7 oraz Strefę Kultury Studenckiej. Zdjęcia wykonano z wielu kątów, zapewniając odpowiednie nakładanie się ujęć do poprawnej rekonstrukcji 3D. Następnie, przy użyciu techniki **Structure from Motion** (SfM), wygenerowano chmurę punktów 3D, określając rozmieszczenie punktów w przestrzeni oraz pozycje i orientacje kamer. Uzyskana chmura punktów została dodatkowo poddana filtracji z wykorzystaniem metod opartych na analizie sąsiedztwa każdego punktu.

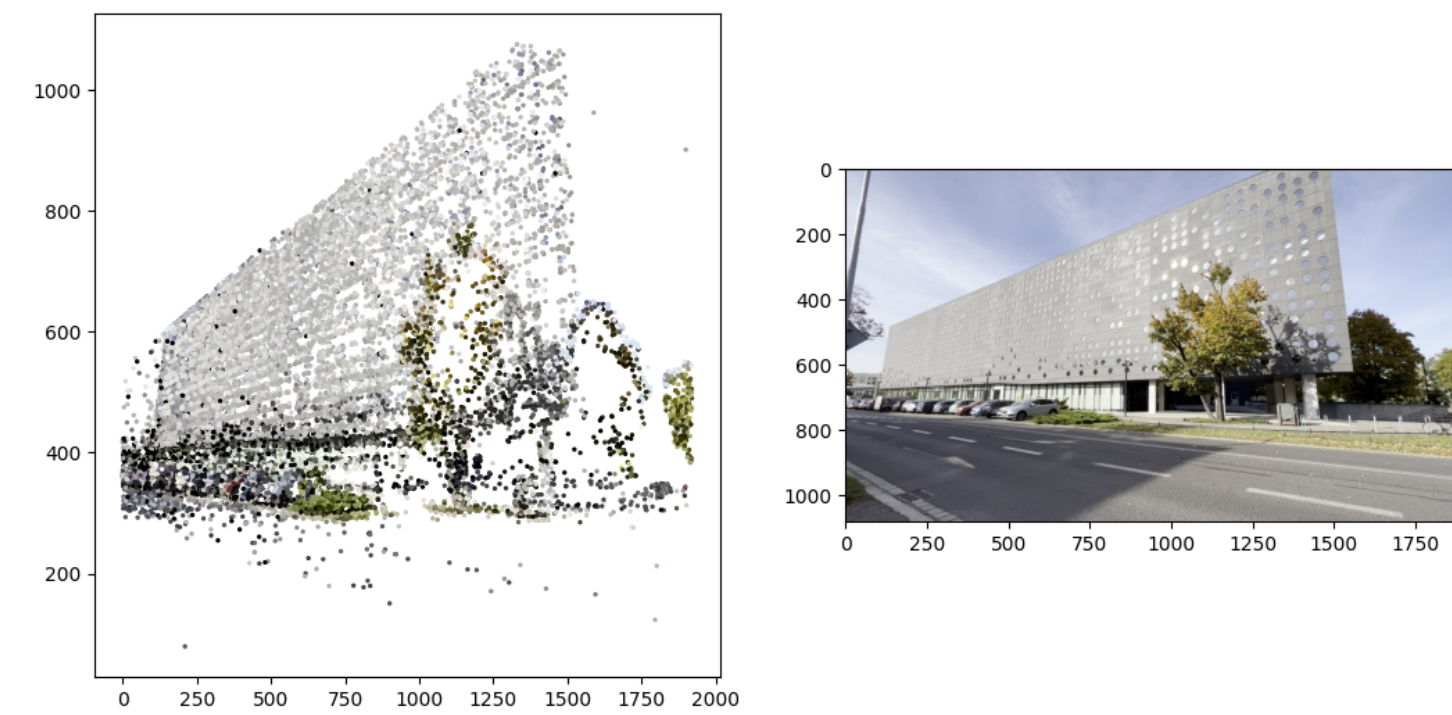


Figure 1: Projektcja przykładowej chmury punktów na płaszczyznę porównana do zdjęcia

RENDEROWANIE

Zaprojektowano intuicyjny interfejs w *PyQt* i *QML*, zintegrowany z wydajnym systemem renderowania *GPU* opartym na *OpenGL*, *OpenCL* i języku *C*, oferującym także wsparcie dla *VisPy*. Aplikacja zapewnia spójne środowisko do obsługi modeli 3D, obejmujące procesy takie jak generowanie chmury punktów, segmentacja i wizualizacja danych. Rendering wykorzystuje pliki .ply oraz Shader Storage Buffer Object (SSBO). Rysunek obok przedstawia główny widok aplikacji, w tym własny render.



PODSUMOWANIE

Opracowane oprogramowanie umożliwia kompleksowe modelowanie obszarów miejskich z wykorzystaniem fotogrametrii. Wśród oferowanych użytkownikowi funkcjonalności znajdują się m.in. dostosowywanie parametrów, wczytywanie zdjęć, uruchamianie poszczególnych etapów oraz przeglądanie rezultatów. Odpowiednio dobrane i dostrojone algorytmy zapewniają jakościowe wyniki, które mogą być wykorzystywane razem lub oddzielnie.

PRZEBIEG PROCESU



Figure 2: Diagram pokazujący kolejne etapy procesu wraz z danymi wejściowymi i wyjściowymi

GAUSSIAN SPLATTING

Wykorzystano bibliotekę *gsplat* do testowania różnych hiperparametrów algorytmu **Gaussian Splatting**, optymalizując trenowanie pod względem czasu i pamięci. Kluczowe były liczba gaussianów, strategia i częstość adaptacji, liczba iteracji i stopień zmiennych harmonicznych.

scena	PSNR	LPIPS	SSIM	LG	Czas
SKS	22.03	0.71	0.25	2,9e+6	2:53
C5	21.98	0.71	0.26	4,5e+6	10:40
C7	22.63	0.72	0.29	3,0e+6	15:15

Table 1: Metryki PSNR, SSIM oraz LPIPS, liczba gaussianów oraz czas trenowania dla testowych scen.



Figure 3: Przykładowe wizualizacje dla SKS oraz C5 (od lewej do prawej: prawdziwe zdjęcie i widok modelu)

SEGMENTACJA

Przeprowadzono segmentację semantyczną chmury punktów, przypisując każdemu punktowi kategorię, taką jak budynek, droga czy zieleń miejska. Wykorzystano do tego implementację sieci **PointNet** w bibliotece *Pytorch* oraz zbiór danych SensatUrban. Kluczowe wyzwania obejmowały próbkowanie danych, radzenie sobie z niebalansowanymi kategoriami oraz unikanie wycieku danych, co rozwiązano m.in. przez ważenie funkcji straty i odpowiednie przygotowanie zestawów treningowych. Poprzez podział wejściowej chmury na tzw. *chunki* model jest w stanie przeprowadzić inferencję na zbiorze punktów o wielkości nawet paru milionów elementów.

Table 2: Metryki: średnia ważona dokładność, średni ważony F1-score, F1-score dla klas: budynki i zieleń miejska

Metryki	Unweighted Model	Weighted Model
OwA	43	51
OwF	38	46
budynkiF	87	84
zieleńF	0	42

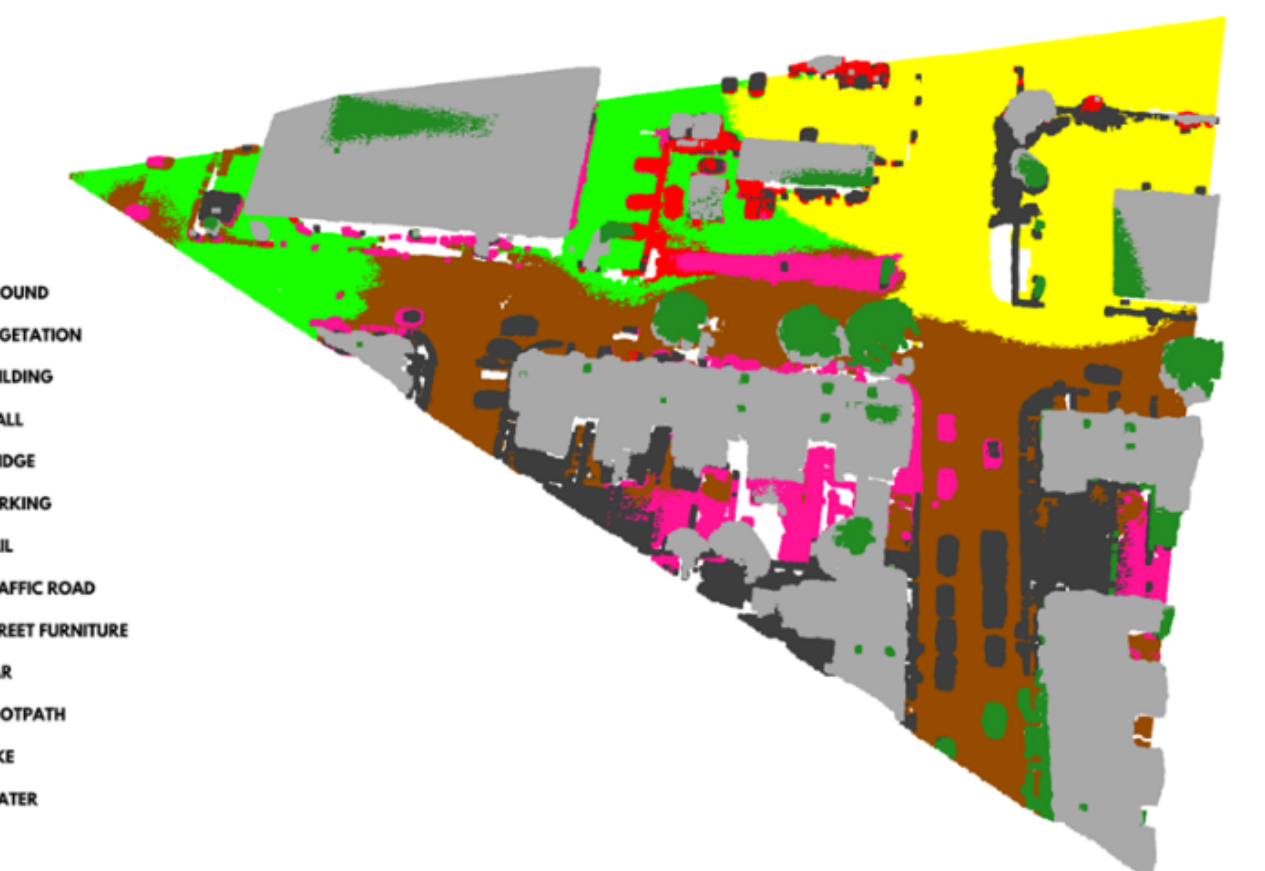


Figure 4: Segmentacja na jednym z bloków ze zbioru SensatUrban