



URB3D

METODA TRÓJWYMIAROWEGO MODELOWANIA OBSZARÓW URBANISTYCZNYCH Z WYKORZYSTANIEM METOD FOTOGRAMETRII

DANIEL BORKOWSKI, JULIA FARGANUS, RAFAŁ MIELNICZUK, KATARZYNA WOCHAL



ABSTRAKT

Przedmiotem projektu było wykonanie aplikacji wykorzystującej metody fotogrametrii do modelowania trójwymiarowych scen miejskich. Zaimplementowany program umożliwia użytkownikowi zrekonstruowanie chmury punktów i modelu 3D na podstawie podanych na wejściu zdjęć fotogrametrycznych za pomocą dostosowanych do specyfiki problemu w toku eksperymentów rozwiązań *structure from motion* i *gaussian splatting*. Otrzymana w ten sposób chmura może być przez użytkownika poddana segmentacji semantycznej, przeprowadzanej przez wytrenowany do tego celu model sztucznej inteligencji w postaci sieci neuronowej. Aplikacja oferuje również wizualizację wykonanych obliczeń, która możliwa jest dzięki użyciu przystosowanego dla większej wydajności mechanizmu renderowania.

Wytworzony produkt informatyczny ze względu na integrację wielu rozwiązań i efektywną implementację procesu *end-to-end* przejawia potencjał w zastosowaniach biznesowych począwszy od branż takich jak gry wideo, przez architekturę, robotykę, pojazdy autonomiczne, skończywszy na modelowaniu urbanistycznym.

Słowa kluczowe: reconstruction, 3D modeling, urban modeling, semantic segmentation, structure from motion, 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.

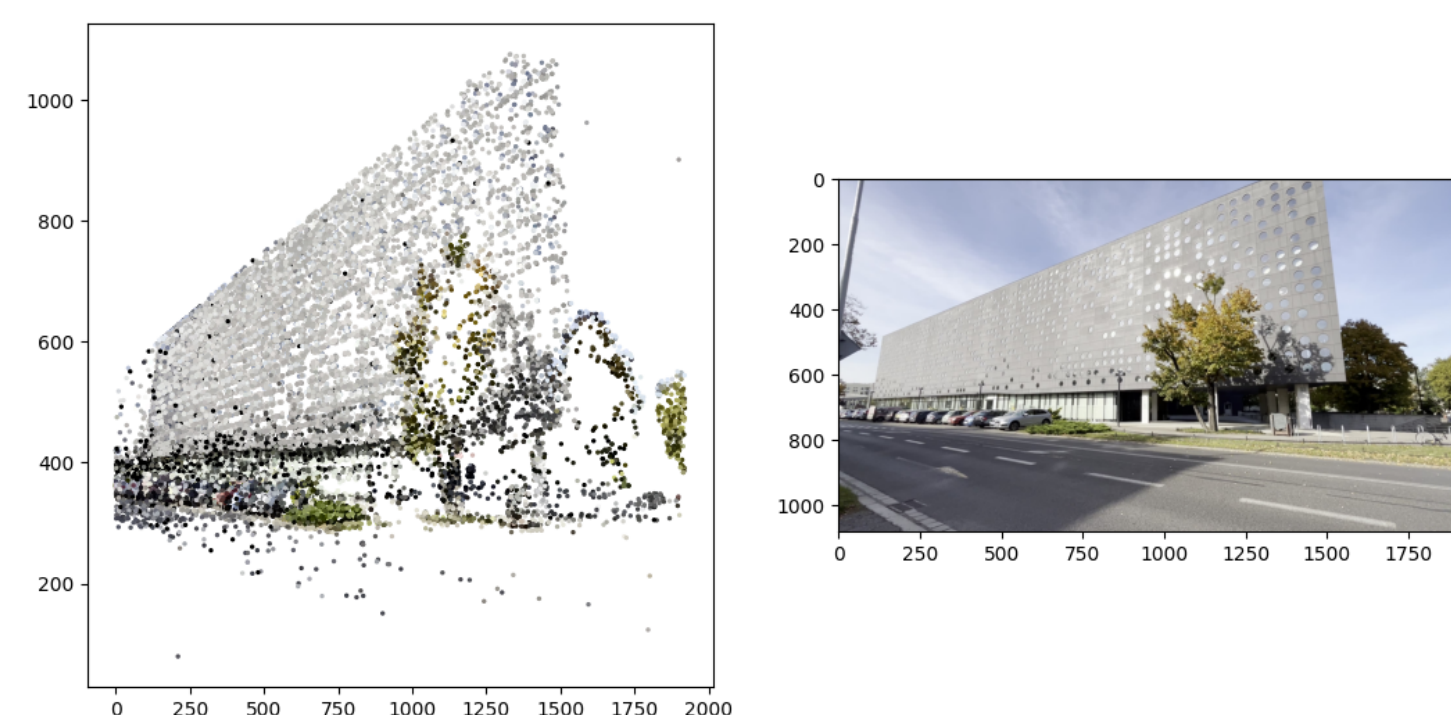


Figure 1: Projektcja przykładowej chmury punktów na płaszczyznę porównana do zdjęcia

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), umożliwiając efektywną pracę nawet przy scenach z milionami obiektów.

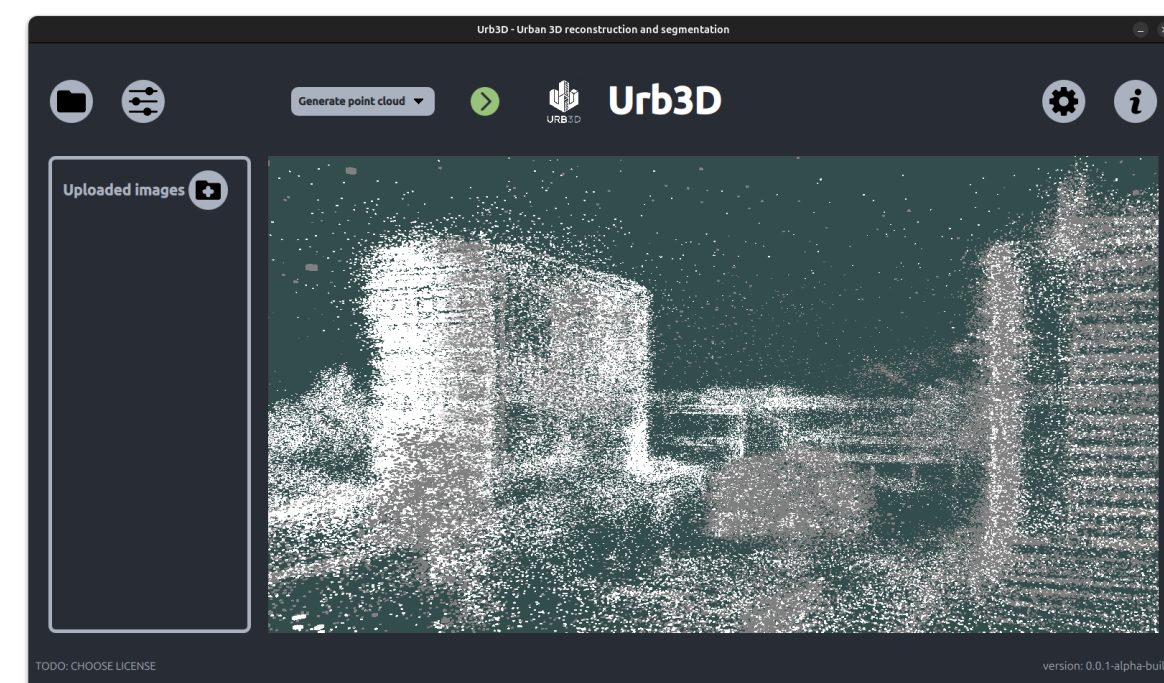


Figure 5: Główny widok aplikacji, w tym własny renderer

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

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

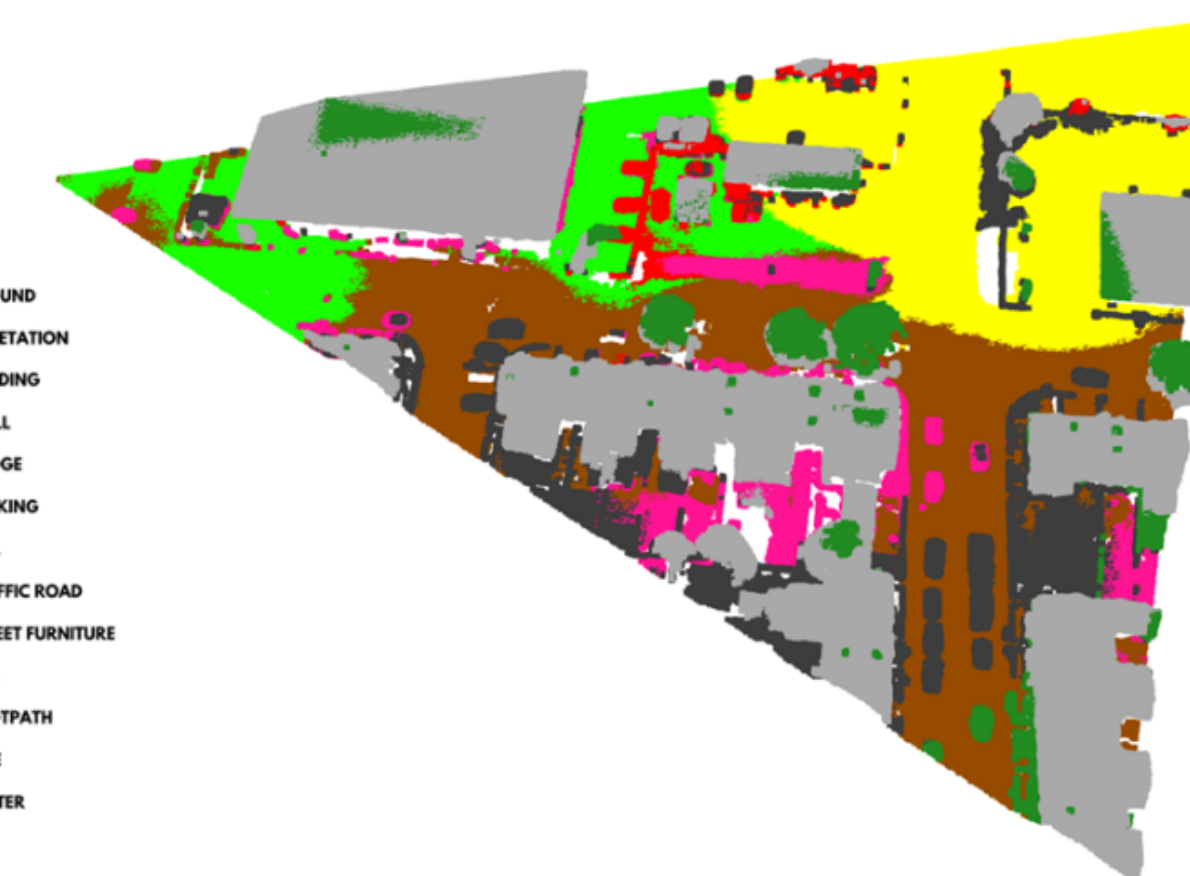


Figure 3: Segmentacja na jednym z bloków ze zbioru SensatUrban

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

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 (M)	Czas
SKS	22.03	0.71	0.25	2,9	2:53
C5	21.98	0.71	0.26	4,5	10:40
C7	22.63	0.72	0.29	3,0	15:15

Table 2: Metryki PSNR, SSIM oraz LPIPS, liczba gaussianów oraz czas trenowania dla testowych scen.

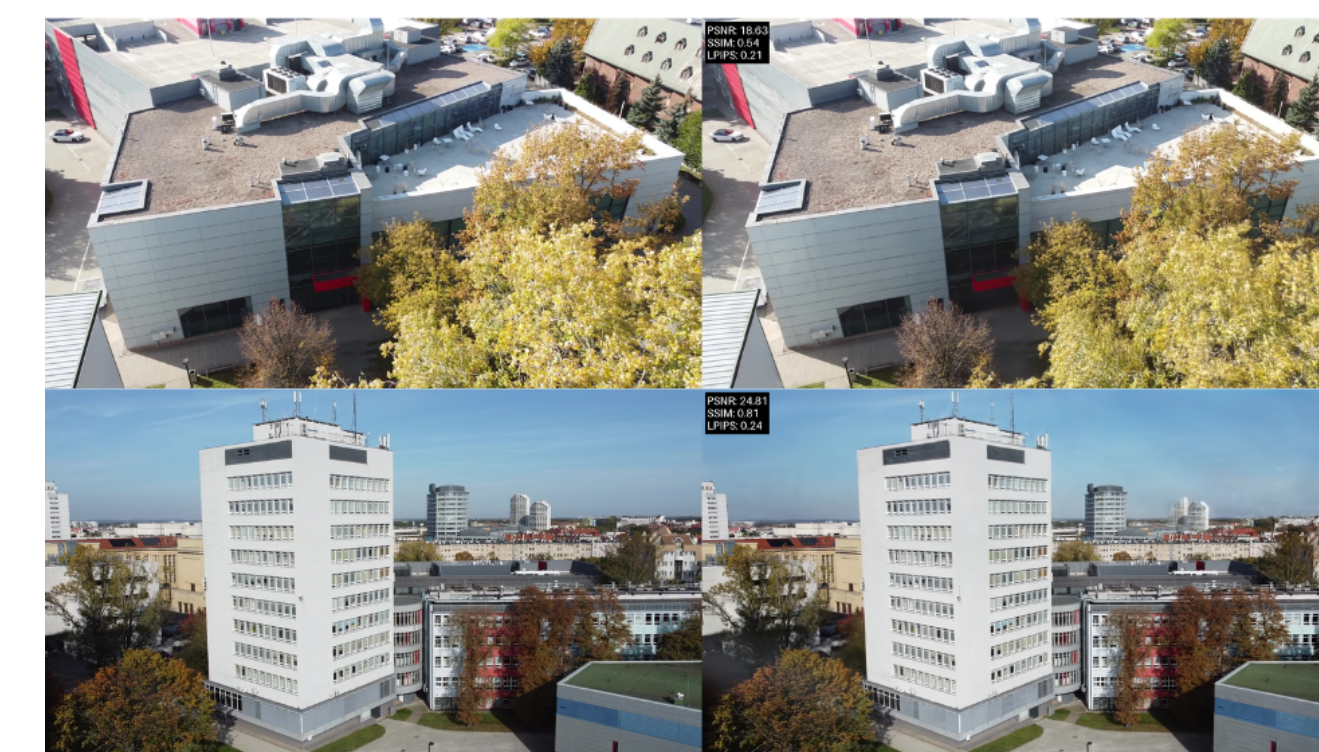


Figure 4: Przykładowe wizualizacje dla SKS oraz C5 (od lewej do prawej: prawdziwe zdjęcie i widok modelu)