

# URB3D



## Metoda trójwymiarowego modelowania obszarów urbanistycznych z wykorzystaniem metod fotogrametrii



**Autorzy:** Daniel Borkowski · Julia Farganus · Rafał Mielniczuk · Katarzyna Wochal

**Opiekun:** dr hab. inż. Marek Krótkiewicz, prof. PWr

### Streszczenie

Celem pracy jest wykonanie aplikacji, która wykorzystuje metody fotogrametrii do modelowania miejskich scen 3D. Dane wejściowe stanowią zdjęcia obszarów miejskich, które są przetwarzane w celu stworzenia modelu 3D, a następnie segmentowane na obiekty przestrzeni miejskiej, takie jak budynki, tereny zielone, itp.. Aplikacja będzie wizualizować model oraz wyniki segmentacji semantycznej.

Innowacyjność tego projektu polega na połączeniu, adaptacji i udoskonaleniu najlepszych dostępnych rozwiązań, takich jak Gaussian Splatting i PointNet, aby stworzyć nowy, kompleksowy produkt.

Przetwarzanie dużych scen miejskich jest wyzwaniem dla obecnie istniejących rozwiązań, które skupiają się głównie na pojedynczych obiektach lub zamkniętych scenach. Typowa scena miejska natomiast może obejmować setki zdjęć, a powstała chmura może zawierać miliony punktów. Dodatkowym wyzwaniem jest niebalansowana reprezentacja kategorii semantycznych. Nasze rozwiązanie ma na celu efektywne przetwarzanie dużych zbiorów danych przy rozsądnym zużyciu zasobów czasowych i pamięciowych.

Zastosowania biznesowe otrzymywanych w ten sposób modeli 3D są szerokie: od gier wideo, przez architekturę, robotykę, pojazdy autonomiczne, po modelowanie urbanistyczne.

## 1 WPROWADZENIE

Nasz projekt skupia się na problemie rekonstrukcji trójwymiarowej scen urbanistycznych, jej klasyfikacji oraz wizualizacji. Warto podkreślić, że obszar naszej pracy jest relatywnie nowy i stawia wyzwania związane z efektywnością przetwarzania dużego zbioru danych - w naszym przypadku chmury punktów, która może składać się nawet z paru milionów punktów. Na rynku dostępne są rozwiązania które możemy wykorzystać, więc naszym głównym celem jest zbadanie ich użyteczności w naszym problemie i ich ewentualna adaptacja.

Nasze rozwiązanie będzie umożliwiało przeprowadzenie rekonstrukcji do modelu trójwymiarowego na podstawie odpowiednio przygotowanego zbioru zdjęć, klasyfikację otrzymanej sceny na zbiór predefiniowanych klas istotnych w kontekście scen urbanistycznych, oraz wizualizację wykonanych obliczeń.

Jako zespół stawiamy następujące cele, które chcemy zrealizować:

1. Skomponowanie własnego zbioru danych
2. Wykorzystanie algorytmu Gaussian Splatting do rekonstrukcji sceny 3D
3. Filtracja chmury punktów przy użyciu różnych technik
4. Zastosowanie architektur sieci neuronowych takich jak PointNet do klasyfikacji chmury punktów
5. Adaptacja istniejących bibliotek do wizualizacji wyników
6. Implementacja własnego algorytmu do renderowania gaussianów

## 2 STAN WIEDZY

Unikalność naszego projektu wynika z połączenia wielu rozwiązań które istnieją samodzielnie na rynku. Algorytm Structure-from-Motion jest popularną fotogrametryczną techniką otrzymywania chmury punktów ze zbioru zdjęć i jego implementacja oferowana jest m. in. przez oprogramowanie COLMAP.

W przypadku modelu 3D często stosowaną techniką są siatki, jednak ich wadą jest niekompatybilność z algorytmami sztucznej inteligencji. Popularne są też rozwiązania wykorzystujące sieci neuronowe

jak np. NeRF, ale długi czas trenowania jest nieefektywny. Z tego powodu zdecydowaliśmy się na wykorzystanie algorytmu Gaussian Splatting, który buduje model sceny z tzw. gaussianów, które można interpretować jako punkty rozmyte. Można napotkać różne adaptacje tego algorytmu, jak np. CityGaussian.

Ważnym krokiem jest również filtracja chmury punktów w celu usunięcia odstających punktów lub tych nieistotnych dla wyników klasyfikacji. W tym obszarze znajdują się np. techniki statystyczne czy oparte na sąsiedztwie.

Istniejące architektury sieci neuronowych dla zadania segmentacji są głównie przeznaczone dla scen zamkniętych lub pojedynczych obiektów. Popularnym rozwiązaniem jest PointNet oraz jego następnik PointNet++ oparte na wielowarstwowym perceptronie, jak i również bardziej skomplikowane rozwiązania jak KPConv wykorzystujące konwolucje. Wyzwaniem dla naszego projektu będzie dostosowanie takich architektur do chmury punktów o wielkości rzędu milionów punktów.

W przypadku renderowania istnieją rozwiązania przeznaczone zarówno do chmur punktów, jak i do splatów, jednak wyzwaniem jest ...

### **Wykorzystane oprogramowanie**

Główymi językami programowania są C/C++ oraz Python.

## **3 WYNIKI**

Proponuję tutaj wrzucać zdjęcia, tabele, raczej nie dużo tekstu WYNIK - oprogramowanie i to co wychodzi jako wynik oprogramowania

### **3.1 Gaussian Splatting**

Przy pomocy biblioteki *gsplat* implementującej Gaussian Splatting w Pythonie wykonaliśmy eksperymenty polegające na uruchomieniu algorytmu dla różnych wartości hiperparametrów w celu znalezienia optymalnych dla nich wartości. Poniżej znajdują się wnioski dotyczące znaczenia hiperparametrów na podstawie obserwacji procesów trenowania dla różnych ustawień.

1. Liczba Gaussianów: w przypadku scen urbanistycznych w celu oddania odpowiedniej szczegółowości potrzebne jest parę milionów Gaussianów, dla naszych scen było to zwykle 3 mln.
2. Strategia adaptacji: wybór strategii jest istotny gdyż określa sposób dodawania i usuwania Gaussianów oraz wpływa na jakość sceny.
3. Liczba iteracji: zwykle im dłużej trenowana jest scena tym lepsze wyniki otrzymujemy, jednak zależy to również od przyjętej strategii.
4. Stopień zmiennych harmonicznych: zmienne harmoniczne wyrażają kolor, im większy stopień tym lepsza jakość sceny, ale też zwiększone zużycie pamięci.

tutaj zdjęcia dla oczków



Rysunek 1: Scena SKS (od lewej do prawej: prawdziwe zdjęcie i widok modelu)

## **4 PODSUMOWANIE**