Instrukcja laboratoryjna 5

RealityScan i cyfrowa reprezentacja obiektów rzeczywistych

1. Cel i kontekst ćwiczenia

Nowoczesne aplikacje interaktywne, od gier AAA po przemysłowe cyfrowe bliźniaki i coraz częściej zawierają obiekty wprost zaczerpnięte z rzeczywistego świata. Fotogrametria cyfrowa pozwala przetransformować dowolny fizyczny przedmiot w pełnoprawny model 3D, zachowując zarówno geometrię, jak i wygląd wizualny powierzchni.

W tym laboratorium przeprowadzimy pełny **pipeline fotogrametryczny**, od wykonania dokumentacji zdjęciowej telefonem komórkowym, przez generowanie modelu 3D w aplikacji RealityScan, aż po integrację zeskanowanego obiektu ze środowiskiem Unity. Przy okazji omówimy temat **cyfrowych bliźniaków** (ang. digital twins), jednego z kluczowych trendów współczesnej informatyki przemysłowej.

Po zakończeniu tego laboratorium student:

- rozumie zasadę działania fotogrametrii i tłumaczy, jak zdjęcia 2D prowadzą do rekonstrukcji 3D,
- potrafi samodzielnie wykonać dokumentację fotograficzną obiektu spełniającą wymagania fotogrametryczne,
- potrafi przygotować model 3D z użyciem RealityScan i wyeksportować go do Unity,
- potrafi zintegrować model fotogrametryczny ze sceną wirtualną i skonfigurować materiały,
- potrafi przeprowadzić krytyczną porównawczą analizę modelu fotogrametrycznego i klasycznego modelu 3D,
- rozumie pojęcie cyfrowego bliźniaka i wskazuje przykłady jego zastosowania w informatyce i inżynierii.

2. Wprowadzenie teoretyczne

2.1. Fotogrametria, matematyczne podstawy

Fotogrametria (gr. **photos**, światło, **gramma**, zapis, **metron**, miara) to nauka o pozyskiwaniu informacji o kształtach i położeniu obiektów na podstawie zdjęć fotograficznych [3]. W wersji cyfrowej (ang. **Image-Based Modelling**, IBM) algorytm przetwarza zespół zdjęć wykonanych z różnych punktów widokowych i odtwarza pełną geometrię 3D oraz teksturę powierzchni obiektu.

Proces rekonstrukcji opiera się na algorytmie **Structure from Motion (SfM)** [4], który w uproszczeniu wykonuje następujące kroki:

1. Detekcja cech charakterystycznych (ang. keypoints) na każdym zdjęciu, np. naroża, krawędzie, charakterystyczne plamy tekstury.
2. Dopasowanie cech między zdjęciami (ang. feature matching), wyznaczenie, które punkty na zdjęciu A odpowiadają punktom na zdjęciu B.
3. Estymacja położenia kamery dla każdego zdjęcia względem obiektu (ang. camera pose estimation).
4. Generowanie chmury punktów (ang. point cloud), zbioru punktów 3D odpowiadających cechom widocznym na co najmniej dwóch zdjęciach.
5. Tworzenie siatki polygonowej (ang. mesh) na podstawie chmury punktów (algorytmy: Poisson Surface Reconstruction, Delaunay triangulation).
6. Generowanie tekstury, rzutowanie pikseli zdjęć na powierzchnię siatki i tworzenie mapy UV.

□ **MIEJSCE NA GRAFIKĘ / DIAGRAM** □

Rysunek 1. Pipeline algorytmu Structure from Motion (SfM)

Schemat blokowy: zdjęcia wejściowe → detekcja cech → dopasowanie → estymacja pozycji kamery → chmura punktów → siatka 3D → tekstura. Każdy etap ilustrowany schematyczną grafiką.

Sugerowana wysokość: ~ jakieś 9 cm

Każdy punkt siatki 3D powstaje tylko wtedy, gdy algorytm potrafi znaleźć ten sam fragment obiektu na kilku zdjęciach. Z tego powodu zdjęcia muszą mieć duże zakładkowanie, stałą ostrość i podobną ekspozycję. Jeśli student wykona zdjęcia zbyt rzadko lub zmieni oświetlenie w połowie akwizycji, SfM może oszacować błędne pozycje kamer, a model będzie miał dziury albo lokalne deformacje.

2.2. Wymagania dotyczące materiału fotograficznego

Jakość końcowego modelu 3D jest wprost uzależniona od jakości zdjęć wejściowych. Najważniejsze błędy fotogrametrii powstają jeszcze przed uruchomieniem przetwarzania. RealityScan może odrzucić słabe zdjęcia albo wygenerować model z artefaktami, ale nie naprawi brakującej informacji geometrycznej, której nie ma w materiale wejściowym. Najczęściej popełnianie błędy fotograficzne i ich konsekwencje dla rekonstrukcji przedstawia poniższa tabela:

Błąd fotograficzny	Skutek dla modelu	Rozwiązanie
Zbyt małe zakładkowanie zdjęć (< 60%)	Luki w siatce, artefakty geometryczne	Fotografuj z zachodzeniem > 70–80%
Rozmycie (ruch aparatu, zbyt długa ekspozycja)	Błędna detekcja cech, zniekształcenia siatki	Stabilizacja, krótki czas nasświetlania
Jednolita, bezstrukturalna powierzchnia	Brak cech do dopasowania, dziury	Naklej tymczasowe naklejki / użyj matowej powłoki
Zbyt błyszcząca powierzchnia (refleksje)	Fałszywe cechy, zniekształcenia tekstury	Matowy spray, miękkie światlo, polaryzator
Ruch obiektu między zdjęciami	Rozmywanie się geometrii, artefakty	Stabilizacja obiektu, statyw, turntable
Niewystarczająca liczba zdjęć (< 30)	Brakujące strony, słabe pokrycie siatki	Minimum 40–60 zdjęć dla małych obiektów
Zmienne oświetlenie (słońce/cienie)	Niespójna tekstura, artefakty koloru	Oświetlenie rozproszone, chmurny dzień

Zakładkowanie 70-80% daje algorytmowi wiele wspólnych punktów między sąsiednimi zdjęciami, a krótki czas naświetlania ogranicza rozmycie, które niszczy keypointy. Matowe, teksturowane powierzchnie są preferowane, ponieważ zapewniają stabilne cechy obrazu; refleksy i przezroczystość zmieniają wygląd punktów w zależności od kąta patrzenia, więc algorytm może uznać je za różne lub fałszywe fragmenty obiektu.

□ **MIEJSCE NA GRAFIKĘ / DIAGRAM** □

Rysunek 2. Schematy prawidłowego i błędnego ustawienia zdjęć

Widok z góry: trzy wzorce ścieżki fotograficznej wokół obiektu (obwód + pas górny + zeni t), zaznaczone zakładkowanie zdjęć. Obok: przykład błędnego podejścia (zdjęcia tylko z jednej strony).

Sugerowana wysokość: 7 cm

2.3. RealityScan, charakterystyka narzędzia

RealityScan to mobilna aplikacja fotogrametryczna opracowana przez Epic Games we współpracy z firmą Capturing Reality (twórcy oprogramowania RealityCapture) [6]. Działa na urządzeniach iOS (iPhone/iPad z LiDARem lub bez) oraz Android. Aplikacja prowadzi użytkownika przez proces akwizycji zdjęć w czasie rzeczywistym, wyświetla podgląd pokrycia obiektu i ostrzega o niewystarczającym zachodzeniu zdjęć.

Na laboratorium wybieramy RealityScan, ponieważ skraca czas między wykonaniem zdjęć, a uzyskaniem modelu i pozwala skoncentrować się na pełnym przepływie pracy w ciągu jednych zajęć. Narzędzia desktopowe dają większą kontrolę i często wyższą jakość, ale wymagają więcej czasu na konfigurację, mocniejszego sprzętu i znajomości parametrów rekonstrukcji.

Cecha	RealityScan	Alternatywy
Platforma	iOS 15+ / Android 10+	3DF Zephyr (PC), Meshroom (PC, open-source), Polycam (iOS/Android)
Przetwarzanie	Chmurowe (serwery Epic Games)	Lokalne (Meshroom, RealityCapture Desktop)
Eksport	FBX, OBJ, GLTF, USD	Zależy od oprogramowania
Limit bezpłatny	200 skanów/miesiąc	Meshroom: bez limitu (lokalnie)
LiDAR	Przyspiesza akwizycję na wspieranych urządzeniach	Wymagany przez niektóre aplikacje AR
Jakość siatki	Średnio-wysoka (dobra na cele real-time)	RealityCapture Desktop: bardzo wysoka (producja)

RealityScan narzuca konkretny sposób pracy: akwizycję mobilną, przetwarzanie chmurowe i eksport do formatów użytecznych w silnikach 3D. Student powinien rozumieć zarówno zalety tego wyboru, jak i ograniczenia wynikające z automatyzacji oraz zależności od serwerów.

> **Dostępność aplikacji na zajęciach**
> RealityScan wymaga smartfona z systemem iOS 15+ lub Android 10+. Jeśli urządzenie mobilne nie jest dostępne lub skan nie uda się z powodów technicznych, prowadzący udostępni gotowy plik modelu (.fbx + tekstury) do dalszej pracy w Unity.

2.4. Cyfrowe bliźniaki, koncepcja i zastosowania

W praktyce przemysłowej model 3D ma wartość dopiero wtedy, gdy może reprezentować, monitorować lub symulować rzeczywisty obiekt w systemie informatycznym. Dlatego tworzenie modelu fotogrametrycznego nie jest celem samym w sobie.

Cyfrowy bliźniak (ang. **digital twin**) to dynamiczna, zsynchronizowana wirtualna replika fizycznego obiektu, procesu lub systemu [7]. W odróżnieniu od statycznego modelu 3D, bliźniak cyfrowy jest podłączony do swojego fizycznego odpowiednika przez strumień danych z czujników, co pozwala na monitorowanie stanu w czasie rzeczywistym, symulację scenariuszy i predykcyjne utrzymanie ruchu.

Branża	Przykład zastosowania	Korzyść
Przemysł 4.0	Cyfrowy bliźniak turbiny wiatrowej z czujnikami wibracji	Predykcyjne utrzymanie ruchu, redukcja przestojów
Budownictwo / BIM	Wirtualna kopia budynku aktualizowana przez IoT	Monitorowanie zużycia energii, diagnostyka
Medycyna	Model organu stworzony z danych tomograficznych	Planowanie operacji, trening lekarzy

Urbanistyka	Cyfrowy bliźniak miasta (np. Singapur Virtual Singapore)	Symulacje ruchu, planowanie przestrzenne
Lotnictwo i kosmonautyka	Bliźniak statku powietrznego aktualizowany w locie	Diagnostyka, optymalizacja zużycia paliwa
Gry i symulacje	Trening personelu na wiernej kopii maszyny	Redukcja kosztów szkoleń, bezpieczeństwo

W kontekście tego laboratorium tworzymy uproszczony bliźniak cyfrowy: statyczny model fotogrametryczny integrowany z wirtualnym środowiskiem Unity. Pełnoprawne bliźniaki wymagają dodatkowo warstwy integracji danych i infrastruktury IoT, co jest przedmiotem specjalistycznych kursów inżynierskich [8].

Model 3D pokazuje wygląd i kształt obiektu, natomiast cyfrowy bliźniak utrzymuje relację z realnym obiektem przez dane, aktualizacje i logikę analityczną. W tym laboratorium celowo zaczynamy od statycznego modelu, ponieważ poprawna geometria, skala i materiały są fundamentem, na którym można później budować warstwę danych.

3. Przygotowanie do ćwiczenia

Odpowiednie przygotowanie ogranicza ryzyko przerwania ćwiczenia z powodów organizacyjnych. Fotogrametria wymaga jednocześnie sprawnego telefonu, konta, odpowiedniego obiektu, dobrego oświetlenia i projektu Unity; brak jednego elementu może zablokować cały pipeline.

Urządzenie mobilne	Smartfon z iOS 15+ lub Android 10+. Aplikacja RealityScan zainstalowana i zalogowana (konto Epic Games, bezpłatne).
Obiekt do skanowania	Wybrany przez studenta przedmiot fizyczny: butelka, but, zabawka, artefakt, fragment narzędzia. Unikamy obiektów lustrzanych, półprzezroczystych i jednolicie białych.
Oświetlenie	Dostęp do równomiernie oświetlonego miejsca (okno północne bez bezpośredniego słońca, lub sztuczne światło rozproszone). Unikamy oświetlenia kierunkowego tworzącego ostre cienie.
Projekt Unity	Projekt z poprzednich laboratoriów lub nowy projekt 3D (Core). Nowa scena ScenaLab05 w folderze Scenes projektu.
Konto Epic Games	Rejestracja na unrealengine.com (bezpłatna) jest wymagana do eksportu modelu z RealityScan.

Lista ma charakter kontrolny, ponieważ czas laboratorium jest ograniczony. Jeżeli konto Epic Games, instalacja aplikacji lub scena Unity będą przygotowywane dopiero w trakcie zajęć, student może nie zdążyć wykonać analizy porównawczej, która jest najważniejszą częścią oceny jakości modelu.

4. Ćwiczenia praktyczne

Część praktyczna jest ułożona liniowo: wybór obiektu, zdjęcia, rekonstrukcja, import, analiza i dokumentacja. Takie uporządkowanie odzwierciedla profesjonalny pipeline produkcyjny i ułatwia diagnozowanie błędów, ponieważ wiadomo, na którym etapie powstał problem.

4.1. Selekcja i przygotowanie obiektu do skanowania

Nie każdy obiekt nadaje się do fotogrametrii. Pierwszym i często pomijanym krokiem jest ocena skanowalności obiektu oraz jego optyczne przygotowanie. Inwestycja kilku minut na tym etapie ma przełożenie na jakość finalnego modelu, gdyż niektóre materiały są fizycznie trudne dla fotogrametrii. Lepiej wybrać prostszy obiekt i poprawnie przejść cały pipeline niż tracić czas na obiekt, którego refleksy, przezroczystość albo brak tekstury uniemożliwią rekonstrukcję.

Krok 1. Ocena obiektu

Ocena obiektu przed skanowaniem zapobiega sytuacji, w której student dopiero po kilkunastu minutach przetwarzania odkrywa, że model jest bezużyteczny. Kryteria doboru wynikają bezpośrednio z potrzeby wykrywania stabilnych cech obrazu. Przed przyciśnięciem przycisku skanowania oceniamy nasz obiekt według poniższej listy:

- **Dobry kandydat:** obiekt o bogatej, matowej teksturze, wyraźnie geometrycznie zdefiniowany, o rozmiarze 10–50 cm, nieruchomy.
- **Zły kandydat:** lustrzane lub połyskliwe powierzchnie, przezroczysty plastik/szkło, jednokolorowe materiały bez wzoru (biała kula, stalowy walec).
- **Wymaga przygotowania:** metalowe narzędzia, błyszcząca ceramika, należy naświetlić matowym aerozolem (np. Aesub Scanning Spray) lub użyć pułku kredowego.

Krok 2. Ustawienie stanowiska

1. Umieszczamy obiekt na neutralnym tle (szara mata, karton), z dala od błyszczących powierzchni.
2. Zapewniamy równomierne oświetlenie, najlepiej dwa źródła po obu stronach obiektu, bez bezpośredniego słońca.
3. Opcjonalnie: układamy wokół obiektu arkusz papieru z wydrukowanymi znacznikami (ang. coded targets), poprawiają dokładność rekonstrukcji skali.
4. Jeśli posiadamy obrotnicę (ang. turntable), ustawiamy obiekt na niej, aparat pozostaje nieruchomy. Jeśli nie, sam chodzi wokół obiektu.

□ **MIEJSCE NA GRAFIKĘ / DIAGRAM** □

Rysunek 3. Przykłady prawidłowo i nieprawidłowo przygotowanego stanowiska

Dwa zdjęcia lub ilustracje: lewa strona, obiekt na matowym tle, równomierne oświetlenie, znaczniki wokół. Prawa strona, obiekt na błyszczącej podłodze, okno za nim (kontrświatło), brak neutralnego tła.

Sugerowana wysokość: 7 cm

4.2. Dokumentacja fotograficzna, wykonanie zdjęć

Dokumentacja fotograficzna jest najważniejszym etapem technicznym, ponieważ zdjęcia są jedynym źródłem danych dla rekonstrukcji. Zastosowanie trzech pasów zapewnia pokrycie bocznych, górnych i skośnych powierzchni, co redukuje dziury w siatce.

Celem jest zebranie co najmniej **60 zdjęć** pokrywających obiekt z wszystkich stron. Przyjmujemy zasadę trzech pasów fotografowania: pas równikowy (poziomy), pas górny (nachylony) oraz zdjęcia zenitu (od góry). Fotogrametria oparta na zdjęciach tylko z jednego poziomu **zawsze** daje modele z brakującym stropem lub spodami [5].

Krok 1. Tryb ręczny aparatu

Tryb ręczny ogranicza zmienność między zdjęciami. Algorytm SfM zakłada, że różnice między zdjęciami wynikają głównie ze zmiany pozycji kamery, a nie z automatycznej zmiany ekspozycji, ostrości, kontrastu czy przetwarzania HDR.

1. Ustawiamy czas naświetlania: `1/120 s lub krótszy`, eliminuje rozmycie od drżeń ręki.
2. **ISO**: **`100–400`**, unikamy wysokiego ISO powodującego szum (ziarno) na teksturze.
3. **Ostrość**: **`manual focus lub AF-lock`**, zapobiegamy zmianie ostrości się aparatu pomiędzy zdjęciami.
4. **Wyłączamy HDR i filtry** aparatu, chcemy fotografię techniczną, nie artystyczną.

Krok 2. Schemat wykonywania zdjęć

Stały dystans i regularny kąt obrotu zapewniają przewidywalne zakładkowanie. Fotografowanie chaotyczne może pozostawić niewidoczne sektory, a wtedy nawet duża liczba zdjęć nie poprawi modelu. Poruszamy się wokół obiektu w stałej odległości, wykonując zdjęcia co ok. 10–15 stopni obrotu (co daje ok. 24–36 zdjęć na pas):

- **Pas 1**, równikowy: aparat na wysokości środka obiektu, lekko nachylony ku środkowi (kąt ok. 15–20° w dół). Liczymy co najmniej 24 zdjęcia.
- **Pas 2**, górny: aparat uniesiony, kąt nachylenia ku obiektowi ok. 45–60° w dół. Kolejne 24 zdjęcia.
- **Pas 3**, zenit: 4–8 zdjęć bezpośrednio od góry (o ile obiekt ma widoczną górną powierzchnię).
- **Pas 4**, spody (opcjonalny): jeśli obiekt ma interesującą dolną powierzchnię, unosimy obiekt na podstawce i fotografujemy od dołu.

□ MIEJSCE NA GRAFIKĘ / DIAGRAM □

Rysunek 4. Schemat trójpasmowego fotografowania obiektu

Widok izometryczny obiektu z naniesionymi torami kamery: pas równikowy (niebieski), pas górny (zielony), zdjęcia z zenitu (czerwony). Strzałki pokazują kierunek ruchu i zachodzenie zdjęć (overlap).

Sugerowana wysokość: 9 cm

Kontrola „zakładkowania”

Każde zdjęcie powinno pokrywać co najmniej 70–80% obszaru widocznego na poprzednim zdjęciu. Aplikacja RealityScan wyświetla na żywo mapa ciepła pokrycia, należy fotografować aż wszystkie obszary obiektu zielenią się na mapie.

Kontrola zakładkowania w trakcie robienia zdjęć jest skuteczniejsza, niż próba naprawy modelu po rekonstrukcji. Jeżeli obszar nie został uchwycony z wystarczającą liczbą wspólnych punktów, program nie ma danych, z których mógłby odtworzyć geometrię.

4.3. Generowanie modelu w RealityScan

Po zebraniu zdjęć aplikacja przetwarza je w chmurze obliczeniowej Epic Games i generuje model 3D. Czas przetwarzania zależy od liczby zdjęć i obciążenia serwerów, zwykle wynosi 5-15 minut.

Generowanie modelu następuje dopiero po kontroli zdjęć, ponieważ przetwarzanie chmurowe wymaga czasu i zależy od jakości materiału. Na tym etapie student uczy się także krytycznie oceniać wynik, a nie tylko akceptować pierwszy wygenerowany model.

Krok 1. Akwizycja w aplikacji RealityScan

Nazewnictwo i monitorowanie pokrycia w aplikacji porządkują pracę oraz ułatwiają identyfikację skanu po eksporcie. Dobra nazwa pliku zapobiega pomyleniu modeli, szczególnie gdy kilka zespołów pracuje równolegle. Otwieramy RealityScan → New Scan. Nadajemy skanowi opisaną nazwę (np. Lab05_NazwaObiektu_DataInicjały).

1. Włączamy tryb Capture i powoli poruszamy się według schematu trójpasmowego.
2. Obserwujemy mapę ciepła pokrycia w aplikacji. Przerywamy zdjęcia dopiero, gdy cały obiekt jest zielony.
3. Klikamy Done Capturing i potwierdzamy przesłanie zdjęć na serwer.

Krok 2. Ocena i akceptacja modelu

Ocena modelu przed eksportem pozwala podjąć racjonalną decyzję: „doskanować” obiekt, poprawiać model ręcznie albo użyć modelu zastępczego. To odpowiada realnemu procesowi produkcyjnemu, w którym jakość zasobu jest weryfikowana przed integracją z projektem.

1. Po przetworzeniu RealityScan wyświetla interaktywny podgląd 3D modelu. Oceniamy:
   - Kompletność geometrii, czy nie ma widocznych dziur lub brakujących powierzchni?
   - Jakość tekstury, czy kolorystyka jest spójna, bez widocznych światłocieniowych artefaktów?
   - Dokładność kształtu, czy geometria odpowiada rzeczywistemu obiektowi?
2. Jeśli jakość jest niewystarczająca (liczne dziury, zniekształcenia), możemy:
   - dodać dodatkowe zdjęcia i wysłać do ponownego przetworzenia,
   - zaakceptować model i uzupełnić go ręcznie w Blenderze,
   - skorzystać z modelu udostępnionego przez prowadzącego.

Krok 3. Eksport modelu

Format FBX i wysoka rozdzielczość tekstury są kompromisem między zgodnością z Unity a jakością wizualną. Eksport w niewłaściwym formacie lub bez tekstur może wymusić ręczne odtwarzanie materiałów i utrudnić analizę końcową

1. Klikamy Export → wybieramy format FBX (zalecany dla Unity).
2. Ustawiamy rozdzielczość tekstury: 4096 × 4096 px (dobra jakość przy akceptowalnej wadze pliku).
3. Wybieramy docelową aplikację w menu eksportu: Unity.
4. Zapisujemy plik na smartfonie, następnie przesyłamy go na komputer (kabel USB, AirDrop, cloud).

> **Struktura eksportowanego archiwum**
> RealityScan eksportuje model jako archiwum .zip zawierające: plik .fbx (siatkę z informacjami o materiałach), plik tekstury .jpg lub .png (Albedo/Diffuse) oraz opcjonalnie mapę normalnych. Rozpakowujemy archiwum przed importem do Unity.

4.4. Import i integracja modelu z Unity

Import do Unity wymaga osobnego rozdziału, ponieważ model fotogrametryczny nie jest typowym, ręcznie przygotowanym assetem. Trzeba zweryfikować skalę, orientację, materiał, kolizję i wydajność, inaczej poprawny skan może źle działać w scenie czasu rzeczywistego.

Modele fotogrametryczne różnią się od ręcznie tworzonych modeli 3D w kilku kluczowych aspektach: zazwyczaj mają dużą liczbę polygonów, jedną dużą teksturę (zamiast wielu małych) i bywają przeskalowane nieoczekiwanie. Każdy z tych aspektów wymaga uwagi podczas importu.

Krok 1. Import do Unity

Porządek folderów i ustawienia importu decydują o powtarzalności projektu. Umieszczanie plików w osobnym katalogu ułatwia ocenę, archiwizację i ewentualne usunięcie zasobu bez naruszenia pozostałej sceny.

1. Tworzymy folder Assets/Models/FotogrametriaLab05 i kopiujemy tam plik .fbx oraz pliki tekstur z rozpakowanego archiwum.
2. Unity automatycznie importuje model. Zaznaczamy plik .fbx w panelu Project.
3. W Inspector → Import Settings konfigurujemy:
   - **Scale Factor:** 1.0, sprawdzimy skalę po umieszczeniu w scenie i korygujemy w razie potrzeby.
   - **Read/Write Enabled:** zaznaczamy, jeśli planujemy dynamicznie modyfikować siatkę w kodzie.
   - **Generate Lightmap UVs:** zaznaczamy, potrzebne do poprawnego działania oświetlenia statycznego.
   - **Materials**: Import via MaterialDescription.
4. Klikamy Apply

Krok 2. Konfiguracja materiału fotogrametrycznego

Materiał musi być skonfigurowany ręcznie, ponieważ tekstura fotogrametryczna zwykle zawiera informację o kolorze i częściowo o oświetleniu, ale nie pełny zestaw map PBR. Zbyt wysoka gładkość lub metaliczność może spowodować nienaturalny wygląd obiektu.

1. W folderze Materials tworzymy nowy materiał Mat_FotogrametriaLab05.
2. Shader: Universal Render Pipeline/Lit (lub Standard w projektach bez URP).
3. W pole Albedo (Base Map) przeciągamy teksturę diffuse z folderu Textures.
4. Smoothness: ustawiamy na 0.0–0.2, modele fotogrametryczne mają pieczoną teksturę bez informacji o metaliczności/chropowatości, więc przyjmujemy wartości złożone.
5. Jeśli eksport zawierał mapę normalnych, to przypisujemy ją do pola Normal Map (pamiętamy o zmianie Texture Type na Normal Map).
6. Przeciągamy materiał na model w scenie.

Krok 3. Skalowanie i orientacja

Skala i orientacja są krytyczne dla integracji z fizyką, kamerą i interakcją. Model wizualnie poprawny, ale ma niepoprawną skalę, zniszczy proporcje sceny, działanie kolizji i wyniki analizy porównawczej.

1. Umieszczamy model w scenie i sprawdzamy skalę względem referencyjnych obiektów (np. sześcianu 1 m × 1 m × 1 m).
2. Jeśli model jest źle zorientowany (leży na boku, odwrócony), możemy:
   - zmienić Rotation w Import Settings (pole Bake Axis Conversion) [9]
   - lub dodać model jako dziecko pustego obiektu-rodzica i korygować obroty przez rodzica.
3. Dodajemy do modelu Mesh Collider (convex: false dla statycznego), aby był elementem sceny fizycznej.

> **Ostrożnie ze skalą modeli fotogrametrycznych**
> RealityScan nie zawsze poprawnie interpretuje skalę bezwzględną obiektu, gdy brak jest znaczników skali w scenie. Model może być 200 razy za duży lub 200 razy za mały. Zawsze porównujemy z referencyjną kostką 1×1×1 m zaraz po imporcie.

Krok 4. Optymalizacja LOD (Level of Detail)

Modele fotogrametryczne mogą mieć setki tysięcy polygonów, to zbyt dużo jak na mały rekwizyt w scenie. Stosujemy komponent LOD Group, który automatycznie zamienia model na prostszy w miarę oddalania kamery.

1. Zaznaczamy model w hierarchii → Add Component → LOD Group.
2. Unity domyślnie tworzy 3 poziomy LOD. Na potrzeby tego ćwiczenia wystarczy jeden uproszczony poziom, naciskamy + i tworzymy LOD1.
3. Dla LOD0 (pełny model) ustawiamy zakres: 0–30%.
4. Dla LOD1 tworzymy uproszczoną wersję siatki. Na tych zajęciach możemy użyć prostego prymitywu (np. Capsule lub Cube) dopasowanego do bryły obiektu jako placeholder.
5. Zaznaczamy Culled dla dystansu > 5% i sprawdzamy w Scene View, jak przełącza się LOD (fioletowy / niebieski kontur wokół obiektu).

LOD jest wymagany, ponieważ modele fotogrametryczne często mają nadmiarową geometrię. W grach i symulacjach liczy się nie tylko wierność obiektu z bliska, ale też koszt renderowania wielu obiektów w scenie.

4.5. Analiza porównawcza modelów

Następnym etapem ćwiczenia jest rzetelna analiza porównawcza modelu fotogrametrycznego i klasycznego modelu 3D pobranego z biblioteki zasobu (np. z Lab 2). Analiza powinna być udokumentowana piśmiennie w folderze „Documentation" projektu.

Analiza porównawcza zamienia ćwiczenie z prostego importu modelu w ocenę inżynierską. Student ma wykazać, kiedy fotogrametria jest opłacalna, a kiedy klasyczny model 3D będzie lepszy pod względem wydajności, kontroli i kosztu przygotowania.

Szablon analizy porównawczej

Kryterium	Model fotogrametryczny	Klasyczny model 3D
Liczba wierzchołków (Verts)	(odczytać ze Stats Unity)	(odczytać ze Stats Unity)
Liczba trójkątów (Tris)	(odczytać ze Stats Unity)	(odczytać ze Stats Unity)
Rozmiar pliku (MB)	(plik .fbx + tekstury)	(plik .fbx + tekstury)
Jakość tekstury (subiet.)	Opis spostrzegań studenta	Opis spostrzegań studenta
Zgodność z obiektem	Wysoka / Średnij / Niska + uzasadnienie	n.d. (model nie odwzorowuje konk. obiektu)
Czas pozyskania	(ile minut trwał cały pipeline?)	(ile minut: znalezienie + import?)
Konfiguracja materiałów	Automatyczna / ręczna + opis trudności	Automatyczna / ręczna + opis trudności
Zastosowania	Kiedy warto użyć fotogrametrii?	Kiedy warto użyć klasycznego modelu?

□ **MIEJSCE NA GRAFIKĘ / DIAGRAM** □

Rysunek 5. Zestawienie wizualne: model fotogrametryczny vs klasyczny model 3D

Dwa zrzuty ekranu z Unity obok siebie: lewy, model fotogrametryczny w Scene View (włączony tryb Wireframe), prawy, klasyczny model 3D w tym samym trybie. Widoczna różnica gęstości siatki.

Sugerowana wysokość: 8 cm

Dlaczego wymagane są konkretne metryki, a nie tylko opis wizualny

Subiektywna ocena wyglądu nie wystarcza, ponieważ model może wyglądać dobrze z jednej kamery, a jednocześnie mieć zbyt dużo trójkątów, ciężkie tekstury albo błędną skalę. Metryki ze Stats Unity pozwalają porównać jakość wizualną z kosztem renderowania i rozmiarem plików.

4.6. Zastosowania fotogrametrii i cyfrowych bliźniaków

Na podstawie zdobytych podczas laboratorium doświadczeń oraz materiałów z wykładu przygotowujemy krótką analizę (~1 stroną A4) umieszczoną w folderze Documentation. Analiza powinna odpowiadać na pytania:

1. Jakie są główne ograniczenia fotogrametrii jako metody pozyskiwania modeli 3D? Które klasy obiektów są z definicji trudne lub niemożliwe do zeskanowania?
2. W jakich zastosowaniach informatycznych i inżynierskich fotogrametria jest już stosowana produkcyjnie? (Odwołaj się do co najmniej dwóch przykładów z literatury [7])
3. Co sprawia, że cyfrowy bliźniak to coś więcej niż tylko model 3D? Jakie elementy infrastruktury technicznej są potrzebne do stworzenia pełnoprawnego bliźniaka?
4. Ocen potencjał użycia RealityScan lub podobnych narzędzi w Twoim obszarze zainteresowań zawodowych.

Wymusza to przejście od wykonania zadania do interpretacji technologii. Student powinien umieć nie tylko odtworzyć procedurę, ale także ocenić, w jakich problemach zawodowych fotogrametria i cyfrowe bliźniaki mają uzasadnienie.

Oczekiwany poziom odpowiedzi

Odpowiedzi powinny wynikać z doświadczenia z własnym modelem i z literatury. Nie wystarczy napisać, że „fotogrametria jest szybka”; należy wskazać, dla jakiego typu obiektu była szybka, jakie ograniczenia wystąpiły i czy uzyskany model nadaje się do zastosowania w czasie rzeczywistym.

4.6. Prezentacja jakości zasobu w Unity

Zadanie to, domyka nasz pipeline produkcyjny, ponieważ w praktyce asset 3D nie jest przekazywany dalej bez krótkiego raportu jakości. Student uczy się dokumentować decyzje techniczne: co zostało zeskanowane, jakie są koszty modelu, jakie artefakty pozostały i czy model jest gotowy do użycia.

Rozbudowanie instrukcji o kolejne zadanie: student przygotowuje krótką mini-prezentację jakości zasobu w formie pliku PDF lub dokumentu w folderze Documentation/QualityReport_Lab05. Raport ma mieć ~1 stronę i zawierać materiał z Unity oraz wnioski techniczne.

Krok 1. Scena prezentacyjna

Jednolita scena prezentacyjna pozwala porównać modele między studentami w podobnych warunkach oświetlenia i kamery. Dzięki temu ocenie podlega jakość assetu, a nie przypadkowe ustawienia sceny.

1. Utwórz w Unity pusty obiekt Lab05_ShowcaseRoot i umieść pod nim model fotogrametryczny, referencyjny sześcian 1×1×1 m oraz jedno źródło światła Area Light lub Directional Light.
2. Ustaw kamerę tak, aby model i sześcian były widoczne jednocześnie.
3. Wykonaj trzy zrzuty ekranu: widok materiału, widok wireframe lub shaded wireframe oraz widok z włączonym LOD Group.

Krok 2. Karta jakości modelu

Karta jakości wymusza zebranie mierzalnych danych. Takie dane są potrzebne, aby zdecydować, czy asset można włączyć do gry, czy wymaga retopologii, kompresji tekstur albo ponownego skanowania.

Pole raportu	Wymagana treść
Nazwa obiektu	Krótka identyfikacja skanowanego przedmiotu.
Liczba zdjęć	Łączna liczba zdjęć użytych w RealityScan oraz informacja o pasach fotografowania.

Parametry eksportu	Format, rozdzielczość tekstury, rozmiar archiwum i rozmiar po rozpakowaniu.
Metryki Unity	Verts, Tris, liczba materiałów, liczba tekstur, docelowa skala w metrach.
Artefakty	Lista zauważonych problemów: dziury, rozmazania tekstury, błędne odbicia, brak spodu, zbyt gęsta siatka.
Decyzja produkcyjna	Jednoznaczna rekomendacja: użyć bez zmian / użyć po optymalizacji / zeskanować ponownie / zastąpić modelem klasycznym.

Krok 3. Krótka obrona techniczna

Krótka obrona techniczna sprawdza, czy student rozumie przyczynę uzyskanej jakości. Celem nie jest wyłącznie pokazanie ładnego modelu, lecz uzasadnienie, dlaczego zastosowano konkretne parametry i jakie kompromisy zostały zaakceptowane.

1. Przygotuj 3-5 zdań wyjaśniających, który etap najbardziej wpłynął na jakość modelu.
2. Wskaż jedną rzecz, którą zmieniłbyś/zmieniłabyś przy ponownym skanowaniu.
3. Określ, czy model nadaje się do użycia jako rekwizyt pierwszoplanowy, tło sceny, element VR/AR czy wyłącznie prototyp.

W katalogu Documentation powinny znaleźć się: QualityReport_Lab05.pdf lub QualityReport_Lab05.docx, trzy zrzuty ekranu z Unity oraz krótka decyzja produkcyjna dotycząca gotowości modelu.

5. Lista kontrolna wykonanych zadań

Przed zakończeniem laboratorium sprawdź, czy wszystkie poniższe elementy zostały zrealizowane:

☐	Obiekt wybrany i oceniony pod kątem skanowalności. Trudne powierzchnie matowo spryskane lub zastosowano inne środki zapobiegawcze.
☐	Stanowisko fotograficzne przygotowane z równomiernym oświetleniem i neutralnym tłem.
☐	Co najmniej 60 zdjęć wykonanych w trzech pasach (równikowy, górny, zenit).
☐	Model 3D pomyślnie wygenerowany w RealityScan (lub zastosowano model zastępczy)
☐	Model wyeksportowany w formacie FBX z teksturami w rozdzielczości min. 2048 × 2048 px.
☐	Model zaimportowany do Unity z poprawną skalą i orientacją w scenie.
☐	Materiał fotogrametryczny skonfigurowany: Albedo + Normal Map (jeśli dostępna).
☐	Komponent Mesh Collider dodany do modelu fotogrametrycznego.
☐	Komponent LOD Group skonfigurowany (co najmniej LOD0 i LOD1).
☐	Tabela analizy porównawczej modeli wypełniona na podstawie danych ze Stats Unity.
☐	Analiza pisemna zastosowań fotogrametrii i cyfrowych bliźniaków zapisana w Documentation.
☐	Raport jakości QualityReport_Lab05 przygotowany i zapisany w Documentation.
☐	Do raportu dołączono trzy zrzuty ekranu z Unity: materiał, wireframe oraz konfiguracja LOD.
☐	W raporcie sformułowano decyzję produkcyjną dotyczącą dalszego użycia modelu.

Lista kontrolna redukuje ryzyko oddania niekompletnego projektu. Jest ułożona zgodnie z kolejnością pipeline'u, dlatego pozwala szybko wykryć, czy brakuje danych wejściowych, importu, konfiguracji materiału, optymalizacji, analizy albo nowego raportu jakości.

6. Zadanie dodatkowe na 5: 3D Gaussian Splatting (3DGS)

W oparciu o workflow UnityGaussianSplatting, zasób tworzy się przez `Tools -> Gaussian Splats -> Create GaussianSplatAsset` dla pliku `.ply/.spz`, oraz na alternatywnym pakiecie `gsplat-unity`, który obsługuje import plików `.ply` i komponent `Gsplat Renderer`.

6.1. Import obiektu 3D Gaussian Splatting / 3DGS

3D Gaussian Splatting jest inną reprezentacją sceny niż siatka polygonowa. Student musi zobaczyć, że fotogrametria w klasycznym wariancie daje mesh z materiałem, colliderem i LOD, natomiast 3DGS daje zbiór półprzezroczystych, anizotropowych „splatów" renderowanych specjalnym shaderem lub rendererem GPU.

Import wykonujemy jako osobne zadanie, ponieważ kopiowanie procedury FBX prowadziłoby do błędnych oczekiwań: plik 3DGS zwykle nie ma standardowej siatki, nie obsługuje typowego Mesh Collidera i wymaga osobnego komponentu renderującego.

Cel zadania

Student importuje do Unity gotowy obiekt 3DGS w formacie `.ply` lub `.spz`, umieszcza go w osobnej scenie testowej, konfiguruje renderer 3DGS, a następnie porównuje tę reprezentację z modelem fotogrametrycznym FBX przygotowanym wcześniej. Zadanie jest dodatkowe: nie zastępuje importu modelu RealityScan, tylko rozszerza laboratorium o aktualną technikę reprezentacji obiektów i scen 3D.

Wymagane dane wejściowe

1. Gotowy plik 3DGS w formacie `.ply` lub `.spz`. Plik może pochodzić z materiałów prowadzącego, z narzędzia eksportującego splaty albo z publicznego zestawu demonstracyjnego.
2. Projekt Unity używany w laboratorium albo jego kopia. Zalecane jest wykonanie zadania w kopii projektu, ponieważ pakiety 3DGS dodają własne shadery, importery i zasoby renderujące.
3. Komputer z obsługą renderowania GPU zgodną z wymaganiami użytego pakietu. Jeżeli render nie działa na stanowisku laboratoryjnym, student dokumentuje próbę importu oraz korzysta ze zrzutów referencyjnych udostępnionych przez prowadzącego.

Używamy gotowego pliku 3DGS, ponieważ trenowanie splatów z własnych zdjęć wymaga osobnego pipeline'u obliczeniowego, zwykle poza zakresem dwugodzinnego laboratorium. Kopia projektu chroni wcześniejsze wyniki pracy przed konfliktami pakietów, a weryfikacja GPU zapobiega sytuacji, w której student interpretuje problem sprzętowy jako błąd merytoryczny importu.

Krok 1. Przygotowanie sceny i pakietu 3DGS

1. Utwórz nową scenę `ScenaLab05_3DGS` w folderze `Scenes`.
2. Zainstaluj wybrany pakiet obsługujący 3DGS zgodnie z instrukcją prowadzącego. Dopuszczalne są np. `UnityGaussianSplatting` albo `gsplat-unity`, jeżeli są zgodne z wersją Unity używaną na zajęciach.
3. Po instalacji uruchom ponownie Unity, jeżeli pakiet dodał własne importery, menu narzędziowe lub shadery.
4. Utwórz folder `Assets/Models/GS_Lab05` i skopiuj do niego plik `.ply` lub `.spz`.

Dlaczego tak?

Osobna scena oddziela eksperyment 3DGS od sceny z modelem FBX. Instalacja pakietu przed importem pliku jest konieczna, ponieważ standardowy Unity importer nie rozumie atrybutów splatów, takich jak pozycja, skala, rotacja, kolor, przezroczystość i współczynniki funkcji sferycznych. Restart edytora bywa potrzebny, aby Unity zarejestrowało nowe importery i menu edytora.

Krok 2. Konwersja lub import zasobu 3DGS

1. Jeżeli używasz `UnityGaussianSplatting`, otwórz `Tools -> Gaussian Splats -> Create GaussianSplatAsset`, wskaż plik wejściowy `.ply` lub `.spz` i zapisz wygenerowany asset w folderze `GS_Lab05`.
2. Jeżeli używasz pakietu z automatycznym importerem `.ply`, przeciągnij plik do folderu `Assets` i poczekaj na utworzenie zasobu 3DGS widocznego w panelu `Project`.
3. Nazwij zasób w sposób jednoznaczny, np. `GS_NazwaObiektu_Lab05`.
4. Sprawdź w Inspectorze liczbę splatów, rozmiar zasobu oraz ewentualne ustawienia kompresji lub jakości.

Większość plików 3DGS jest przechowywana jako PLY, ale semantyka takiego pliku różni się od klasycznej chmury punktów PLY i od siatki polygonowej. Dlatego część pakietów wymaga konwersji do własnego formatu assetu, a część importuje PLY bezpośrednio. Jednoznaczna nazwa zasobu jest wymagana, aby później porównać metryki FBX i 3DGS bez pomylenia plików.

Krok 3. Umieszczenie 3DGS w scenie

1. Utwórz pusty obiekt `GameObject` o nazwie `GO_3DGS_NazwaObiektu`.
2. Dodaj komponent renderujący splaty zgodny z użytym pakietem, np. `Gaussian Splat Renderer` albo `Gsplat Renderer`.
3. Przypisz zaimportowany asset 3DGS do pola zasobu w komponencie renderera.
4. Ustaw pozycję obiektu w centrum sceny i dopasuj skalę względem kostki referencyjnej `1m x 1m x 1m`.
5. Dodaj światło i kamerę tak, aby obiekt był widoczny z podobnego kąta jak model FBX z poprzedniego zadania.

3DGS jest renderowany przez komponent, a nie przez `Mesh Renderer`. Przypisanie assetu do dedykowanego renderera jest odpowiednikiem przypisania mesha i materiału w klasycznym pipeline, ale technicznie działa inaczej. Skala referencyjna jest nadal potrzebna, ponieważ pliki 3DGS również mogą nieść niejednoznaczną skalę, zwłaszcza gdy powstały z rekonstrukcji bez markerów pomiarowych.

Krok 4. Weryfikacja renderowania i ograniczeń

1. Uruchom scenę w trybie `Play` i sprawdź, czy obiekt renderuje się poprawnie podczas ruchu kamery.
2. Sprawdź obiekt z małej, średniej i dużej odległości. Zwróć uwagę na migotanie, sortowanie przezroczystości, gubienie detali oraz koszt wydajnościowy.
3. Spróbuj dodać `Mesh Collider`. Jeżeli nie jest to możliwe albo nie działa zgodnie z oczekiwaniem, zapisz w raporcie, że 3DGS nie dostarcza klasycznej geometrii kolizyjnej.

4. Wykonaj trzy zrzuty ekranu: widok sceny z 3DGS, Inspector zaimportowanego assetu oraz porównanie 3DGS z modelem FBX lub klasycznym modelem 3D.

Najważniejszą różnicą między 3DGS, a siatką mesh nie jest wygląd statyczny, lecz zachowanie w czasie rzeczywistym. Splaty mogą wyglądać bardzo fotorealistycznie z typowych kierunków kamery, ale nie muszą być dobrym zamiennikiem obiektu gry wymagającego fizyki, cięcia, animacji szkieletowej albo precyzyjnej interakcji. Test ruchu kamery ujawnia ograniczenia, których nie widać na pojedynczym zrzucie ekranu.

Krok 5. Krótka analiza porównawcza FBX vs 3DGS

W folderze Documentation utwórz plik GS_Comparison_Lab05.pdf (lub docx). Dokument powinien mieć 1 stronę i zawierać odpowiedzi na pytania:

1. Który zasób wygląda bardziej fotorealistycznie z typowej odległości kamery: model FBX z RealityScan czy 3DGS?
2. Który zasób łatwiej wykorzystać jako interaktywny obiekt gry z kolizją i logiką?
3. Jakie są główne koszty użycia 3DGS w Unity: pamięć, wydajność, kompatybilność platform, brak klasycznego mesha?
4. W jakim scenariuszu użyłbyś/użyłabyś 3DGS zamiast klasycznego modelu 3D, a kiedy pozostałbyś/pozostałabyś przy FBX?

Rezultat dodatkowego zadania 3DGS

W projekcie powinny znaleźć się: scena `ScenaLab05_3DGS`, zaimportowany zasób 3DGS, obiekt z komponentem renderującym splaty, trzy zrzuty ekranu oraz dokument `GS_Comparison_Lab05` z porównaniem 3DGS i FBX.

Do listy kontrolnej dodaj:

☐	Scena `ScenaLab05_3DGS` utworzona w projekcie Unity.
☐	Plik `.ply` lub `.spz` z obiektem 3DGS zaimportowany albo skonwertowany do assetu obsługiwanego przez wybrany pakiet Unity.
☐	Obiekt 3DGS umieszczony w scenie za pomocą dedykowanego komponentu renderującego splaty.
☐	Wykonano trzy zrzuty ekranu dokumentujące import, widok sceny i porównanie z modelem FBX.
☐	Dokument `GS_Comparison_Lab05` zapisany w `Documentation` i zawiera wnioski o różnicach między meshem a 3DGS.

7. Literatura i odwołania

Pozycje podstawowe podają fundamenty teoretyczne i podręcznikowe. Są właściwe wtedy, gdy student chce uzasadnić pojęcia SfM, fotogrametrii lub pracy z zasobami 3D w Unity w analizie końcowej.

7.1. Pozycje podstawowe

[1] Hocking, J. (2018). Unity in Action: Multiplatform Game Development in C#. Manning Publications. ISBN 978-1617294969. Rozdział 12: Tworzenie i import zasobów 3D.

[2] Grieves, M. (2014). Digital Twin: Manufacturing Excellence through Virtual Factory Replication. White Paper. Florida Institute of Technology.

[3] Luhmann, T., Robson, S., Kyle, S., Boehm, J. (2020). Close-Range Photogrammetry and 3D Imaging (3rd ed.). De Gruyter. ISBN 978-3110607253.

[4] Schönberger, J.L., Frahm, J.M. (2016). Structure-from-Motion Revisited. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). DOI: 10.1109/CVPR.2016.445.

[5] Agisoft LLC. (2023). Agisoft Metashape User Manual: Professional Edition, Version 2.0. Agisoft LLC. Rozdział 3: Przygotowanie materiału fotograficznego. Dostępne: https://www.agisoft.com/pdf/metashape-pro_2_0_en.pdf

7.2. Dokumentacja techniczna i zasoby online

Dokumentacja techniczna jest właściwym źródłem dla ustawień narzędzi i formatów eksportu. W zadaniach praktycznych należy preferować aktualną dokumentację producenta, ponieważ interfejsy RealityScan i Unity mogą zmieniać się między wersjami.

[6] Epic Games. (2024). RealityScan, Documentation. Dostępne: https://dev.epicgames.com/documentation/en-us/realityscan/realityscan-documentation (dostęp: maj 2026).

[7] Grieves, M., Vickers, J. (2017). Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems. W: Transdisciplinary Perspectives on Complex Systems. Springer. DOI: 10.1007/978-3-319-38756-7_4.

[8] Tao, F., Zhang, M., Nee, A.Y.C. (2019). Digital Twin Driven Smart Manufacturing. Academic Press, Elsevier. ISBN 978-0128176306.

[9] Unity Technologies. (2024). Unity Manual, Importing Models: Model tab. https://docs.unity3d.com/Manual/FBXImporter-Model.html (dostęp: maj 2026).

7.3. Materiały uzupełniające

Materiały uzupełniające pokazują alternatywne narzędzia i studia przypadków. Są przydatne, gdy student chce porównać RealityScan z innymi rozwiązaniami.

Zasób	Opis	URL
Meshroom (AliceVision)	Bezpłatne, open-source oprogramowanie fotogrametryczne na PC. Alternatywa dla RealityScan.	alicevision.org
Polycam	Alternatywna mobilna aplikacja fotogrametryczna (iOS/Android). Płatna subskrypcja.	poly.cam
Capturing Reality, RealityCapture	Profesjonalne oprogramowanie fotogrametryczne (standard branżowy). Bezpłatne do użytku niekomercyjnego.	capturingreality.com
Unity Learn, Working with Models	Oficjalny kurs Unity: praca z modelami 3D, import, materiały.	learn.unity.com
Virtual Singapore	Studium przypadku cyfrowego bliźniaka całego miasta, wideo i artykuły dokumentacyjne.	smartnation.gov.sg