

Міністерство освіти і науки України  
Чернівецький національний університет  
імені Юрія Федьковича

Навчально-науковий інститут фізико-технічних та комп'ютерних наук  
(повна назва інституту/факультету)

Кафедра інформаційних технологій та комп'ютерної фізики  
(повна назва кафедри)

**Інформаційна система для побудови тривимірних моделей  
об'єктів методом фотограметрії**

**Кваліфікаційна робота**

**Рівень вищої освіти - перший (бакалаврський)**

Виконав:

студент 4 курсу, групи 417  
спеціальності

126 Інформаційні системи та технології  
(назва спеціальності)

Васільєв Владислав Євгейнінович

(прізвище та ініціали)

Керівник д.т.н., доц., Баловсяк С. В.

(науковий ступінь, вчене звання, прізвище та ініціали)

До захисту допущено:

Протокол засідання кафедри № 20

від „15” червня 2023 р.

зав. кафедри Борча доц. Борча М. Д.

Чернівці – 2023

## РЕЦЕНЗІЯ НА БАКАЛАВРСЬКУ РОБОТУ

Студента Васільєва Владислава Євгенійовича  
Навчально-науковий інститут фізико-технічних та комп'ютерних наук  
Кафедра інформаційних технологій та комп'ютерної фізики

Чернівецького національного університету імені Юрія Федьковича

**на тему:** Інформаційна система для побудови тривимірних моделей об'єктів методом фотограмметрії

**Актуальність теми** пояснюється практичною потребою у створенні систем, які б давали рекомендації при побудові тривимірних моделей об'єктів методом фотограмметрії

**Практичне значення** отриманих у роботі результатів полягає в тому, що завдяки розробленій інформаційній системі користувачі можуть отримувати рекомендації по побудові тривимірних моделей методом фотограмметрії, що дозволяє підвищити якість отриманих моделей

**Структура та зміст роботи** робота складається з 3-х розділів. У першому розділі проведено системний аналіз предметної області – систем для побудови тривимірних моделей об'єктів, виконано постановку задачі. У другому розділі розглянуто побудову тривимірних моделей об'єктів у програмі 3DF Zephyr. подано інформацію про вхідні та вихідні дані програми. У третьому розділі описано програмну реалізацію інформаційної системи, розроблено базу фактів та правил експертної системи, проведено тестування розробленої системи

**Зауваження** до недоліків роботи слід віднести окремі граматичні помилки


**Оцінка** у результаті виконання бакалаврської роботи студент Васільєв В.Є. засвоїв принципи фотограмметрії, створив працездатну інформаційну систему для побудови тривимірних моделей об'єктів. Вважаю, що завдання бакалаврської роботи Васільєв В.Є. виконав і заслуговує оцінки „відмінно” та присвоєння кваліфікації бакалавра зі спеціальності 126 Інформаційні системи та технології.

**Рецензент**

ІТ компанія Infologic Inc.

Технічний директор

Рогов Роман

  
“ 15 “ 06 2023 р.  
(підпис)

Ім'я користувача:  
Кафедра інформаційних технологій та комп фізики

ID перевірки:  
1015661413

Дата перевірки:  
20.06.2023 22:16:09 EEST

Тип перевірки:  
Doc vs Internet + Library + DB

Дата звіту:  
20.06.2023 22:16:34 EEST

ID користувача:  
36471

Назва документа: Васильєв

Кількість сторінок: 17 Кількість слів: 6605 Кількість символів: 55243 Розмір файлу: 85.74 KB ID файлу: 1015306312

## 5.36% Схожість

Найбільша схожість: 5.36% з Інтернет-джерелом (<http://arr.chnu.edu.ua/jspui/bitstream/123456789/859/1/%d0%9a%d0%>)

5.36% Джерела з Інтернету

1

Сторінка 19

Не знайдено джерел з Бібліотеки

## 2.97% Цитат

Цитати

10

Сторінка 20

Вилучення списку бібліографічних посилань вимкнене

## 2.83% Вилучень

Деякі джерела вилучено автоматично (фільтри вилучення: кількість знайдених слів є меншою за 12 слів та 2%)

0.36% Вилучення з Інтернету

10

Сторінка 21

2.47% Вилученого тексту з Бібліотеки

13

Сторінка 21

**Міністерство освіти і науки України**  
**Чернівецький національний університет**  
**імені Юрія Федьковича**

Навчально-науковий інститут фізико-технічних та комп'ютерних наук  
(повна назва інституту/факультету)

Кафедра інформаційних технологій та комп'ютерної фізики  
(повна назва кафедри)

**Інформаційна система для побудови тривимірних моделей  
об'єктів методом фотограмметрії**

**Кваліфікаційна робота**

**Рівень вищої освіти - перший (бакалаврський)**

Виконав:

студент 4 курсу, групи 417  
спеціальності

126 Інформаційні системи та технології  
(назва спеціальності)

Васільєв Владислав Євгейнінович

(прізвище та ініціали)

Керівник д.т.н., доц., Баловсяк С. В.

(науковий ступінь, вчене звання, прізвище та ініціали)

До захисту допущено:

Протокол засідання кафедри № \_\_\_\_\_

від „\_\_\_\_\_” \_\_\_\_\_ 2023 р.

зав. кафедри \_\_\_\_\_ доц. Борча М. Д.

Чернівці – 2023

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЧЕРНІВЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ЮРІЯ ФЕДЬКОВИЧА

Навчально-науковий інститут фізико-технічних та комп'ютерних наук  
Кафедра інформаційних технологій та комп'ютерної фізики

**ЗАТВЕРДЖУЮ**

Завідувач кафедри

док. фіз.-мат. наук, доц.

\_\_\_\_\_ М. Д. Борча

“ \_\_\_\_ ” \_\_\_\_\_ 2023 р.

**Інформаційна система для побудови тривимірних моделей  
об'єктів методом фотограмметрії**

ЛИСТ ЗАТВЕРДЖЕННЯ

**УЗГОДЖЕНО**

Керівник роботи

докт. техн. наук, доцент

\_\_\_\_\_ С.В. Баловсяк

“ \_\_\_\_ ” \_\_\_\_\_ 2023 р.

Виконавець

студент 4-го курсу

\_\_\_\_\_ В. Є. Васільєв

“ \_\_\_\_ ” \_\_\_\_\_ 2023 р.

## ЗАВДАННЯ НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ

Васильєву Владиславу Євгенійовичу

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Інформаційна система для побудови тривимірних моделей об'єктів методом фотограмметрії

керівник роботи Баловсяк Сергій Васильович, докт. техн. наук, доцент,  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджена наказом вищого навчального закладу від “\_\_” \_\_\_\_\_ 202\_ року № \_\_\_\_\_

2. Строк подання студентом проекту (роботи) 2023 р.

3. Вихідні дані до проекту (роботи) Мета роботи – створити інформаційну систему правил для створення 3D об'єктів методом фотограмметрії. Поетапно вивчити створення 3D моделей, зібрати досвід у вигляді фактів, об'єднати їх логікою правил. Створити рекомендації щодо покращення моделі та полегшення її створення. Створити програму, яка видаватиме рекомендації у відповідь на уведені факти.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1) порівняти існуючі методи роботи з тривимірними моделями

2) дослідити метод фотограмметрії шляхом створення власних моделей

3) описати всі позитивні та негативні фактори, що вплинули на дослід

4) створити систему правил на базі отриманих фактів

5) створити програму та дослідити ефективність її роботи

5. Перелік графічного матеріалу

1) Скріншоти роботи із досліджуваними тривимірними об'єктами

2) Скріншоти роботи експертної системи у CLIP

Студент \_\_\_\_\_ Васильєв В.Є.  
(підпис) (прізвище та ініціали)

Керівник проекту (роботи) \_\_\_\_\_ Баловсяк С.В.  
(підпис) (прізвище та ініціали)

## АНОТАЦІЯ

Кваліфікаційна робота виконана студентом групи 417 Васільєвим Владиславом Євгенійовичем. Тема «Інформаційна система для побудови тривимірних моделей методом фотограмметрії». Робота направлена на здобуття ступеня бакалавр за спеціальністю 126 «Інформаційні системи та технології».

Метою кваліфікаційної роботи є створення та розробка інформаційної системи, яка надаватиме рекомендації для запобігання широкого спектру проблем при створенні 3D моделей. Об'єктом дослідження є процес роботи із тривимірними об'єктами методом фотограмметрії та всі фактори, які на нього впливають. У результаті виконання кваліфікаційної роботи розроблено інформаційну систему, яка видає рекомендації щодо роботи методом фотограмметрії. Програмна реалізація системи виконана у середовищі CLIPSIDE 6.3. Об'єкти досліджень отримані за допомогою програми 3D Zephyr.

Бакалаврська робота містить: кількість сторінок – 85, рисунків – 63, додатків – 1, використаних джерел – 12.

**Ключові слова:** інформаційна система, тривимірний об'єкт, експертна правил, CLIPS, фотограмметрія.

Робота містить результати власних досліджень. Використання чужих ідей, результатів і текстів мають посилання на відповідне джерело.

## ABSTRACT

The qualifying work was completed by a student of group 417 Vasiliev Vladyslav Evgeniyovych. Topic "Information system for building three-dimensional models by photogrammetry". The work is aimed at obtaining a bachelor's degree in specialty 126 "Information systems and technologies".

The purpose of the qualification work is to create and develop an information system that will provide recommendations for preventing a wide range of problems when creating 3D models. The object of research is the process of working with three-dimensional objects by the photogrammetry method and all the factors that affect it. As a result of the qualification work, an information system was developed, which issues recommendations for work using the photogrammetry method. The software implementation of the system is made in the CLIPSIDE 6.3 environment. Research objects are obtained using the 3D Zephyr program.

The bachelor thesis contains: number of pages – 84, figures – 63, appendices – 1, used sources – 12.

**Keywords:** information system, three-dimensional object, expert system, CLIPS, photogrammetry.

The work contains the results of own research. The use of other people's ideas, results and texts are linked to the appropriate source.



## ЗМІСТ

ВСТУП.....	9
РОЗДІЛ 1. ЗАГАЛЬНІ ПОЛОЖЕННЯ.....	11
1.1 Аналіз мети функціонування системи .....	11
1.2 Існуючі методи створення 3D моделей.....	11
1.2.1 Лазерне сканування.....	11
1.2.2 Структуроване світло.....	12
1.2.3 Фотограмметрія.....	13
1.2.4 Ручне моделювання.....	13
1.3 Огляд програмного забезпечення для 3D моделювання.....	14
1.3.1 Autodesk Maya.....	14
1.3.2 Blender.....	15
1.3.3 3Ds Max.....	15
1.3.4 Огляд системи 3D Zephyr.....	16
РОЗДІЛ 2. ДОСЛІДЖЕННЯ ТРИВИМІРНИХ ОБ’ЄКТІВ.....	19
2.1 Зчитування фотографій.....	19
2.1.1 Вхідні дані.....	19
2.2 Поетапне створення 3D моделі.....	22
2.2.1 Розріджена хмара точок.....	22
2.2.2 Щільна хмара точок.....	24
2.2.3 Створення сітки полігонів.....	24
2.2.4 Накладання текстури.....	25
2.3 Редагування та виміри.....	26
2.3.1 Зміна вертикальної орієнтації.....	26
2.3.2 Видалення зайвих елементів.....	27
2.3.3 Зміна за осями координат.....	27
2.3.4 Отримання вимірів та властивостей.....	28
2.3.5 Формати файлів.....	31

2.4 Представлення високоякісної моделі.....	31
РОЗДІЛ 3. ДОДАТКОВІ ДОСЛІДЖЕННЯ ТА ЕКСПЕРТНА СИСТЕМА ПРАВИЛ.....	36
3.1 Представлення еталонної моделі.....	36
3.1.1 Побудова за стандартними параметрами.....	36
3.1.2 Побудова за високими параметрами.....	43
3.2 Представлення дефективної моделі.....	52
3.3 Створення системи правил.....	56
3.3.1 Виділення фактів-умов.....	56
3.3.2 Виділення фактів-наслідків.....	58
3.3.3 Формування фактів-рекомендацій.....	58
3.3.4 Логічне об'єднання у правила.....	59
3.3.5 Опис CLIPSIDE 6.3 .....	61
3.3.6 Створення програми.....	64
ВИСНОВКИ.....	68
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	69
ДОДАТКИ .....	71
Додаток А. Лістинги програми .....	71

## **Перелік умовних позначень**

СПР – система правил та рекомендацій.

## ВСТУП

Мета даної бакалаврської роботи полягає у створенні інформаційної системи для побудови тривимірних моделей об'єктів методом фотограмметрії. Для цього використані власні 3D моделі, отримані на основі зображень та створені за допомогою програмного забезпечення 3D Zephyr. Для забезпечення ефективної роботи з отриманими моделями застосовано систему рекомендацій, засновану на CLIPS 6.3 - мові програмування та середовищі для розробки експертних систем.

Актуальність створення тривимірних моделей через фотограмметрію виявляється у багатьох галузях. Наприклад, у сфері архітектури та будівництва тривимірне моделювання може допомогти архітекторам та дизайнерам у візуалізації проектів, плануванні простору та аналізі конструкцій. В інженерії тривимірне моделювання може бути використане для розробки та оптимізації складних механізмів та систем. У географії тривимірні моделі можуть бути використані для створення детальних карт та віртуальних туристичних маршрутів. В медицині тривимірне моделювання може бути використане для планування хірургічних втручань та створення протезів.

Постановка задачі для бакалаврської роботи передбачає наступні кроки:

- Збір даних. Зібрати набір фотографій об'єктів, для яких необхідно створити тривимірні моделі.
- Обробка фотографій. Використовуючи програмне забезпечення 3D Zephyr, обробити фотографії та отримати початкову тривимірну модель.
- Оптимізація моделі. Виконати оптимізацію отриманої моделі для поліпшення її точності.
- Розробка інформаційної системи. Створити інформаційну систему, яка дозволяє користувачам взаємодіяти з тривимірними моделями та отримувати потрібну інформацію про об'єкти.

- Розробка системи рекомендацій. Застосувати CLIPS 6.3 для створення системи рекомендацій, яка допомагає користувачам вибирати оптимальні параметри обробки та аналізу тривимірних моделей.
- Тестування та оцінка. Провести тестування розробленої системи з використанням різних об'єктів та порівняти результати з відомими методами фотограмметрії.
- Аналіз результатів. Проаналізувати отримані результати та зробити висновки щодо ефективності та придатності розробленої системи.

За допомогою програмного забезпечення 3D Zephyr отримані тривимірні моделі об'єктів шляхом обробки фотографій. 3D Zephyr є потужним інструментом для реконструкції тривимірних моделей та має ряд функцій для поліпшення якості моделей. Використовуючи це програмне забезпечення, можна ефективно створювати тривимірні моделі для інформаційної системи.

Для реалізації системи рекомендацій використано CLIPS 6.3 - мову програмування та середовище для розробки експертних систем. CLIPS надає потужні можливості для розробки логічних правил та інтерпретації знань, що дозволяє створити систему рекомендацій, яка забезпечує оптимальний вибір параметрів обробки та аналізу тривимірних моделей.

Об'єктом дослідження є інформаційна система для побудови тривимірних моделей об'єктів.

Предметом дослідження є експертна система для побудови тривимірних моделей об'єктів методом фотограмметрії.

## **РОЗДІЛ 1**

### **ЗАГАЛЬНІ ПОЛОЖЕННЯ**

#### **1.1 Аналіз мети функціонування системи**

Метою інформаційної системи є надання допомоги для більш точного та якісного створення 3D-моделей. Предметною областю інформаційної системи є такі галузі, як архітектура, створення інтер'єрів екстер'єрів (візуальне оформлення), будівництво, інженерне проектування, 3D-друк та ряд інших. Тривимірне моделювання має дуже широке застосування, тому СПР, правила у якій створені безпосередньо на досвіді розробки 3D-моделей, може стати непоганим додатком, що прискорить роботу і поліпшить якість продуктів в цій галузі.

Існує багато методів та програм для проектування тривимірних об'єктів. Необхідно розглянути те, що вже використовується на сьогоднішній день.

#### **1.2 Існуючі методи створення 3D моделей**

##### **1.2.1 Лазерне сканування (Laser Scanning)**

Лазерне сканування є методом, що використовує принцип розсіяння лазерного проміню для отримання точних тривимірних даних об'єкта. У процесі сканування лазерний промінь посилається на поверхню об'єкта, а відбите світло реєструється детектором. За допомогою прецизійних вимірювань кутів та відстаней лазерна система здатна створювати точкові хмари даних, що представляють геометричну інформацію про об'єкт.

Лазерне сканування знаходить широке застосування у різних галузях, таких як архітектура, будівництво, промисловість, медицина та археологія. В архітектурі і будівництві лазерне сканування використовується для створення точних моделей будівель та ландшафтів, контролю якості та вимірювання розмірів. У промисловості цей метод застосовується для інспекції складних деталей, контролю якості та вирішення інженерних

завдань. В медицині та археології, лазерне сканування допомагає відтворювати анатомічні структури тіла або артефакти з високою точністю.

Однією з переваг лазерного сканування є висока точність та деталізація отриманих даних. Цей метод дозволяє отримувати точні тривимірні моделі з високою роздільною здатністю. Крім того, лазерне сканування є швидким і надійним методом збору даних. Однак, він має свої обмеження, зокрема, висока вартість обладнання та складність обробки великого обсягу отриманих даних.

### **1.2.2 Структуроване світло (Structured Light)**

Структуроване світло є методом, що використовує проекцію структурованого світлового малюнка на об'єкт для отримання тривимірної інформації про його форму та геометрію. Принцип полягає у тому, що структурований світловий шаблон проектується на об'єкт, а камера фіксує його спотворення через деформацію на поверхні об'єкта. За допомогою аналізу змін у структурі проекції можна відновити тривимірну модель об'єкта.

Структуроване світло широко використовується у сфері 3D моделювання, зокрема в робототехніці, віртуальній реальності, комп'ютерній графіці та медицині. У робототехніці цей метод допомагає роботам отримувати інформацію про навколишнє середовище та об'єкти для планування руху та уникнення перешкод. Він також використовується для створення інтерактивних віртуальних середовищ, моделювання поверхонь у комп'ютерній графіці та візуалізації медичних зображень.

Перевагами структурованого світла є висока швидкість збору даних, можливість роботи з різними матеріалами та достатньо висока точність вимірювання. Крім того, він є відносно простим у використанні та не вимагає дорогого обладнання. Однак, метод має свої обмеження, такі як вплив зовнішнього освітлення на точність результатів та обмеження використання у великих просторах.

### **1.2.3 Фотограмметрія (Photogrammetry)**

Фотограмметрія - це метод отримання тривимірних моделей за допомогою фотографій. Принцип фотограмметрії полягає в аналізі зображень з різних ракурсів для визначення геометричних параметрів об'єкта і його поверхні.

Процес отримання 3D моделей за допомогою фотограмметрії включає кілька кроків. Спочатку знімаються фотографії об'єкта з різних ракурсів. Потім застосовуються алгоритми обробки зображень, які аналізують спільні точки на фотографіях та відновлюють геометричні параметри об'єкта. На основі цих даних створюється точкова хмара, яка потім конвертується у поверхневу мережу та текстури для отримання тривимірної моделі.

Фотограмметрія застосовується у різних галузях. В архітектурі та будівництві вона використовується для створення точних моделей будівель, пам'яток архітектури та ландшафтів. У географії та картографії фотограмметрія використовується для створення цифрових моделей місцевості та карт. В медицині фотограмметрія застосовується для створення тривимірних моделей анатомічних структур для діагностики та планування хірургічних втручань.

Переваги фотограмметрії полягають у високій точності отриманих моделей, широких можливостях застосування та відносно низьких вартостях порівняно з іншими методами отримання 3D моделей. Однак, фотограмметрія також має свої обмеження. Вона вимагає високоякісних фотографій з різних ракурсів та оптимальних умов освітлення. Також обробка великої кількості фотографій може бути часо- та ресурсоємною задачею.

### **1.2.4 Ручне моделювання (Manual Modeling)**

Ручне моделювання є процесом створення тривимірних моделей об'єктів шляхом прямого втручання художника або дизайнера. У цьому



методі модель створюється шляхом детального розроблення форми, деталей та текстур з використанням різноманітних інструментів 3D-моделювання.

Ручне моделювання знаходить застосування в різних практичних ситуаціях, зокрема в архітектурі, машинобудуванні, ігровій індустрії та візуальних ефектах. У архітектурі ручне моделювання використовується для створення детальних моделей будівель та ландшафтів. У машинобудуванні цей метод застосовується для проектування автомобілів, літаків та інших транспортних засобів. У галузі відеоігор та візуальних ефектів ручне моделювання використовується для створення персонажів, об'єктів та деталей оточуючого середовища.

Ручне моделювання має свої переваги, зокрема повний креативний контроль над процесом створення моделей та можливість детального пророблення кожної частини об'єкта. Крім того, воно дозволяє художникам виразити свою унікальну творчість та стиль у роботі. Однак, ручне моделювання може бути часо- та ресурсоемним процесом, особливо для складних та деталізованих моделей.

### **1.3 Огляд програмного забезпечення для 3D моделювання**

#### **1.3.1 Autodesk Maya**

Autodesk Maya є одним з провідних програмних засобів для 3D моделювання, анімації та візуальних ефектів. Вона надає широкий набір функцій і можливостей для створення складних тривимірних моделей, анімації персонажів, симуляції фізичних ефектів та багато іншого. Autodesk Maya широко використовується у галузі кіноіндустрії, відеоігор, телебачення та архітектурного дизайну.

Перевагами Autodesk Maya є його потужність, гнучкість та розширюваність. Програма підтримує широкий спектр форматів файлів, що дозволяє зручно обмінюватися даними з іншими програмами. Однак, через велику кількість функцій та складність інтерфейсу, вивчення та володіння Autodesk Maya можуть вимагати певного часу та досвіду.

### **1.3.2 Blender**

Blender є відкритою програмою для 3D моделювання, анімації та візуалізації, що надає безкоштовні та потужні інструменти для створення тривимірних моделей. Вона включає різноманітні функції, такі як моделювання, текстурування, анімація, рендеринг та композитинг. Blender широко використовується в галузі відеоігор, візуальних ефектів, архітектури та анімації.

Blender має багато переваг, зокрема його безкоштовність та відкритий код, що робить його доступним для широкого кола користувачів. Програма також має активну спільноту користувачів, яка постійно розвиває нові функції та розширення. Незважаючи на це, Blender може мати деяку складність навчання для новачків та вимагати більше часу для досягнення професійного рівня.

### **1.3.3 3ds Max**

3ds Max є популярним програмним забезпеченням для 3D моделювання, анімації та візуальних ефектів. Програма надає широкий спектр інструментів для створення складних тривимірних моделей, анімації персонажів, симуляції фізичних ефектів та багато іншого. 3ds Max широко використовується у галузі графічного дизайну, архітектури, відеоігор та візуальних ефектів.

3ds Max має потужні функції моделювання, анімації та рендерингу, що дозволяє створювати реалістичні тривимірні сцени та візуальні ефекти. Програма також підтримує широкий спектр форматів файлів, що полегшує співпрацю з іншими програмами. Однак, 3ds Max може бути дорогим програмним забезпеченням, а також вимагати великих обчислювальних ресурсів для роботи з великими та складними сценами.

### 1.3.4 Огляд системи 3D Zephyr

3D Zephyr є програмним забезпеченням для фотограмметрії, яке дозволяє створювати тривимірні моделі з фотографій. Ця програма має широкий набір функцій та інструментів, що полегшують процес отримання 3D моделей та їх подальшу обробку.

3D Zephyr надає можливість імпортувати фотографії з різних джерел та використовувати їх для створення точкових хмар та тривимірних моделей. Програма використовує потужні алгоритми фотограмметрії, які дозволяють точно визначати геометричні параметри об'єкта.

Однією з ключових переваг 3D Zephyr є його швидкість та ефективність. Вона може обробляти великий обсяг фотографій і створювати якісні 3D моделі у короткі терміни. Крім того, програма надає можливості для редагування, реконструкції та оптимізації отриманих моделей.

Щодо обмежень, 3D Zephyr вимагає достатньо потужного комп'ютера для ефективної роботи з великими обсягами даних. Також варто враховувати, що програма може мати певну складність в навчанні та оволодінні усіма функціями.

В порівнянні з іншими програмами для фотограмметрії, 3D Zephyr пропонує зручний та потужний інтерфейс, широкий набір функцій та високу якість результуючих моделей. Використання 3D Zephyr у поєднанні з фотограмметрією дозволяє отримати детальні, точні та реалістичні тривимірні моделі об'єктів з фотографій.

Таким чином, фотограмметрія є потужним методом отримання тривимірних моделей, а 3D Zephyr - ефективним програмним рішенням для їх створення та обробки. Використання цих технологій у поєднанні відкриває безліч можливостей для різних галузей, включаючи архітектуру, будівництво, географію, картографію та медицину.

Обговорюючи різні методи побудови тривимірних моделей об'єктів, було прийняте рішення використовувати фотограмметрію разом з програмою 3D Zephyr. Цей вибір був зумовлений кількома факторами:

1. Точність та реалістичність: Фотограмметрія використовує високоякісні фотографії для створення тривимірних моделей, що дозволяє отримувати деталізовані та реалістичні результати. Це особливо важливо у випадку детальної візуалізації об'єктів та їх аналізу.
2. Ефективність: Фотограмметрія у поєднанні з 3D Zephyr пропонує швидкий та ефективний процес отримання тривимірних моделей. Широкий набір функцій та алгоритмів програми сприяють оптимальному використанню фотографій та швидкому створенню якісних моделей.
3. Доступність програмного забезпечення: 3D Zephyr є потужною програмою, доступною для використання студентами та професіоналами. Вона має зручний інтерфейс та широкий спектр інструментів, що дозволяє зручно працювати з фотограмметрією та отримувати задоволення від процесу моделювання.

Порівнюючи з іншими методами, фотограмметрія та 3D Zephyr мають свої переваги:

- Лазерне сканування (Laser Scanning): Фотограмметрія не потребує спеціального обладнання, такого як лазерний сканер, що робить її більш доступною та зручною для використання. Крім того, фотограмметрія може забезпечити більш деталізовані результати в складних геометричних сценах.
- Структуроване світло (Structured Light): Фотограмметрія не потребує додаткового обладнання для проектування структурованого світла на об'єкт. Вона використовує лише фотографії, що робить процес більш простим та доступним.
- Ручне моделювання (Manual Modeling): Ручне моделювання вимагає значно більшого часового затрат та навичок. Фотограмметрія разом з 3D Zephyr дозволяють швидко створювати моделі за допомогою автоматичної обробки фотографій та алгоритмів відтворення форми.

## **Висновки до розділу 1**

У підсумку, фотограмметрія та програма 3D Zephyr є потужним інструментарієм для побудови тривимірних моделей об'єктів. Їх поєднання забезпечує точність, ефективність та доступність процесу моделювання, що відкриває широкі можливості для застосування у різних галузях.

## РОЗДІЛ 2

### ДОСЛІДЖЕННЯ ТРИВИМІРНИХ ОБ'ЄКТІВ

#### 2.1 Зчитування фотографій

3DF Zephyr – це комерційне програмне забезпечення для фотограмметрії (вимірювання розмірів об'єктів за їх фотографіями) та побудови 3D-моделей об'єктів на основі зображень. У роботі потрібно використати безкоштовну версію програми 3DF Zephyr Free [1, 2].

##### 2.1.1 Вхідні дані

Для побудови 3D-моделі потрібно спочатку отримати серію початкових фотографій об'єкта (рис. 2.1). Чим більше знімків – тим краща якість моделі. Є деякі обмеження щодо завантажуваних знімків:

- їх має бути мінімум 25 (обмеження програми);
- їх має бути не більше 50 (обмеження безкоштовної версії).

Звісно, чим менше фотографій, тим гірша якість моделі, це відмічається навіть самою програмою – менше завантажувати заборонено.

У безкоштовній версії, яка використовувалась у роботі, «верхнє» обмеження та кількість фото – 50. При придбанні платної версії можна завантажувати більше і відповідно якість моделі буде вищою.

Проте, не дивлячись на обмеженість, 50 фотографій досить наглядно демонструють можливості програми. Для створення СПР цього достатньо, тим більше, що у роботі використовувалось менше фотографій, ніж 50 (для кожної моделі по-різному – загалом по 48-49).

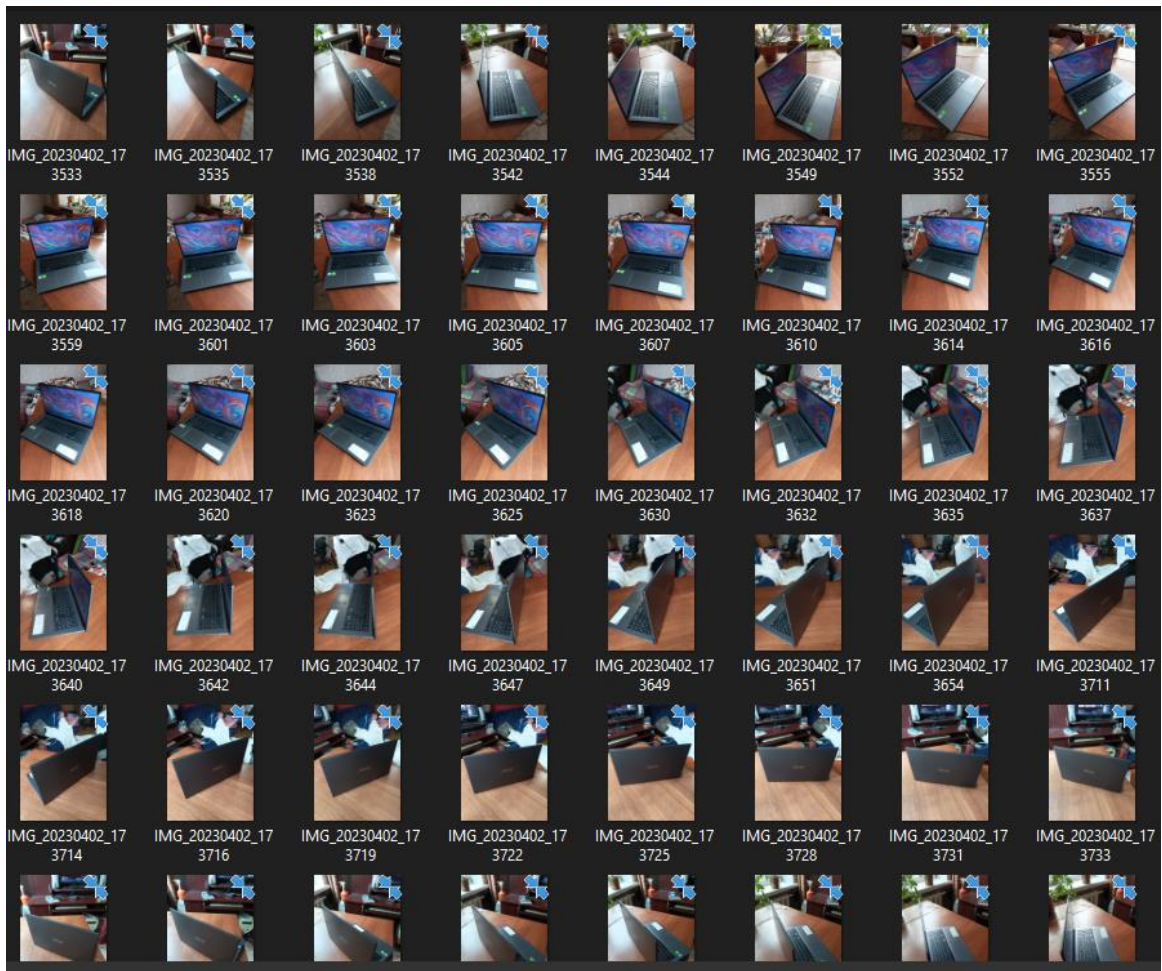


Рис. 2.1. Серія з 48 початкових зображень об'єкта  
(розміром  $1560 \times 2080$  пікселів)

Для побудови 3D-моделі об'єкту створюється новий проект (командою Workflow/ New Project) (рис. 2.2).

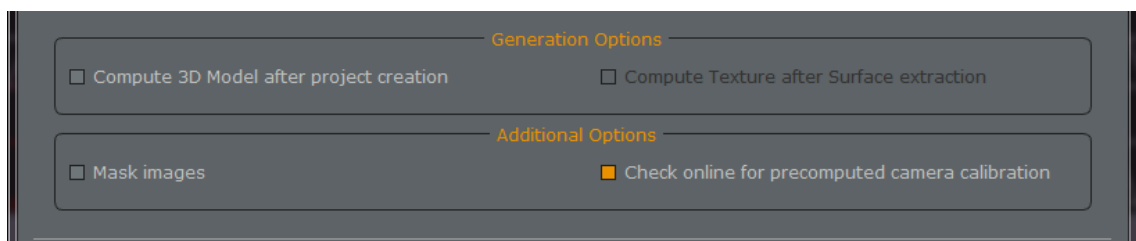


Рис. 2.2. Опції проекту побудови 3D-моделі об'єкту

Отримані фотографії об'єкту (рис. 2.1) додаються до проекту (рис. 2.3). Після цього виконується автокалібрування параметрів фотокамер (рис. 2.4).

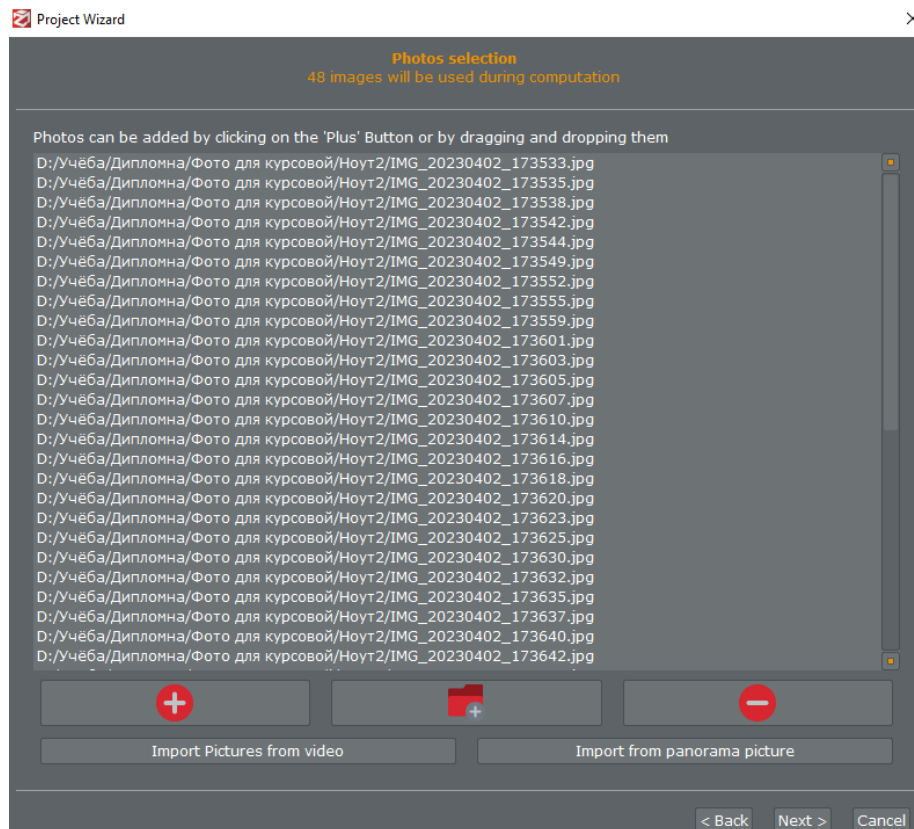


Рис. 2.3. Додавання фотографій до проекту

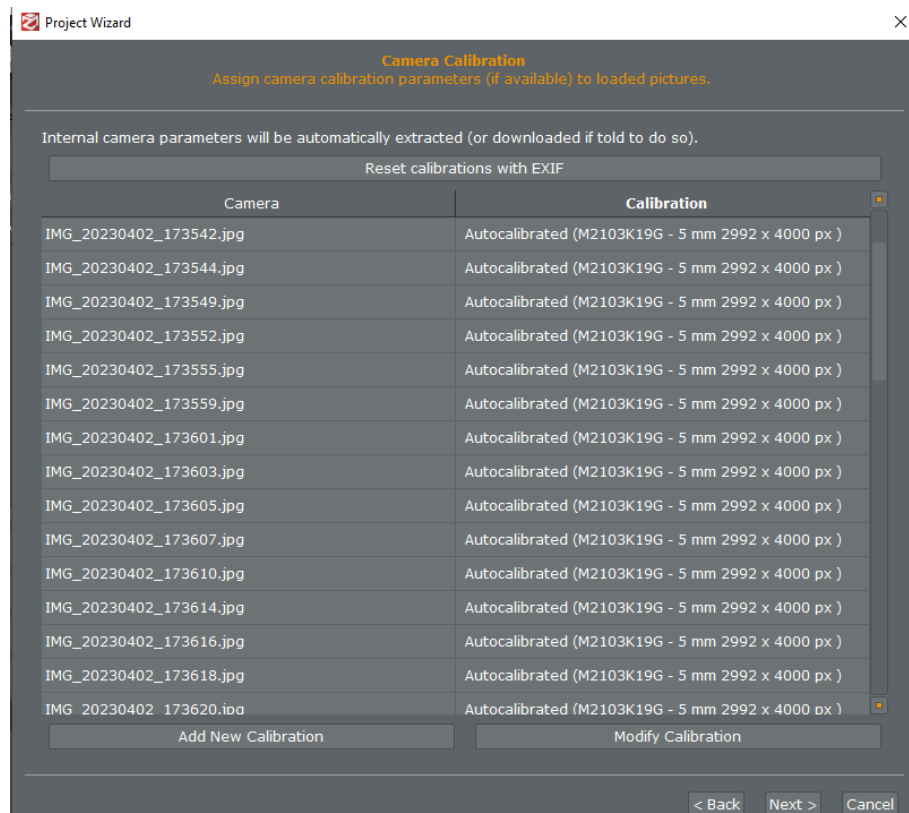


Рис. 2.4. Автокалібрування параметрів фотокамер



## 2.2 Поетапне створення 3D моделі

### 2.2.1 Розріджена хмара точок

Проведено визначення положення та орієнтації камер, а також параметрів внутрішньої геометрії камер (фокусної відстані, дисторсії та ін.) (рис. 2.5). Для цього програма визначає ключові точки (keypoints) зображень та спільні ключові точки різних зображень шляхом суміщення. Для побудови 3D-моделі не використовуються зображення, які погано перекриваються з іншими, або зображення візуально подібних фрагментів (рис. 2.6). У результаті обчислюється розріджена хмара точок (Sparse Point Cloud) в 3D просторі моделі, а також координати та орієнтації фотокамер (рис. 2.7).

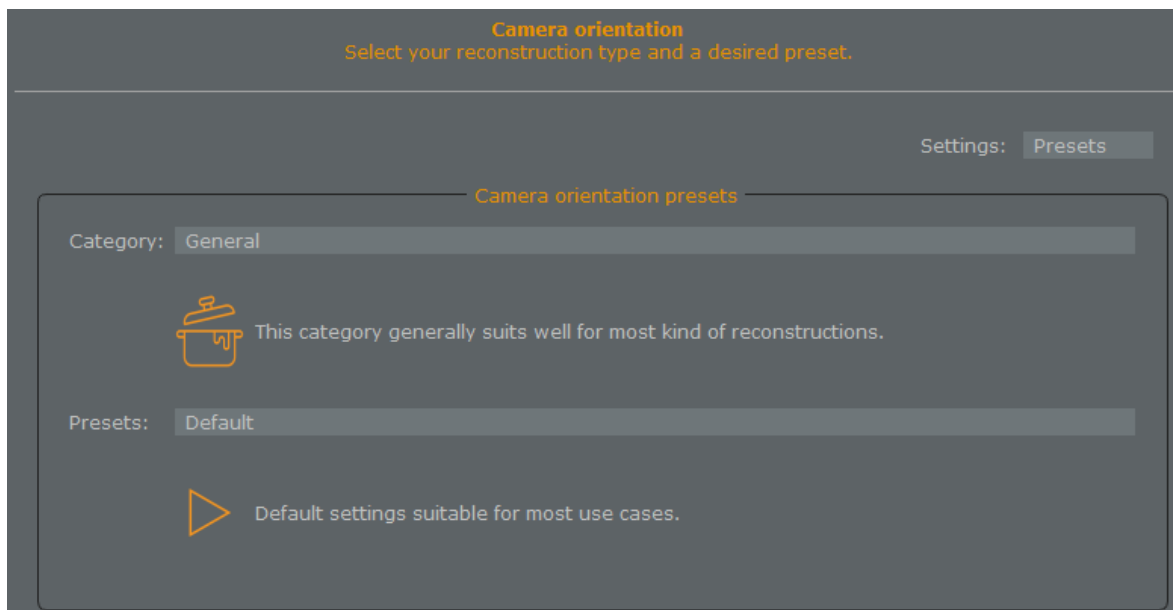


Рис. 2.5. Параметри майстра (за замовчуванням), які використовуються при визначенні положення і орієнтації камер

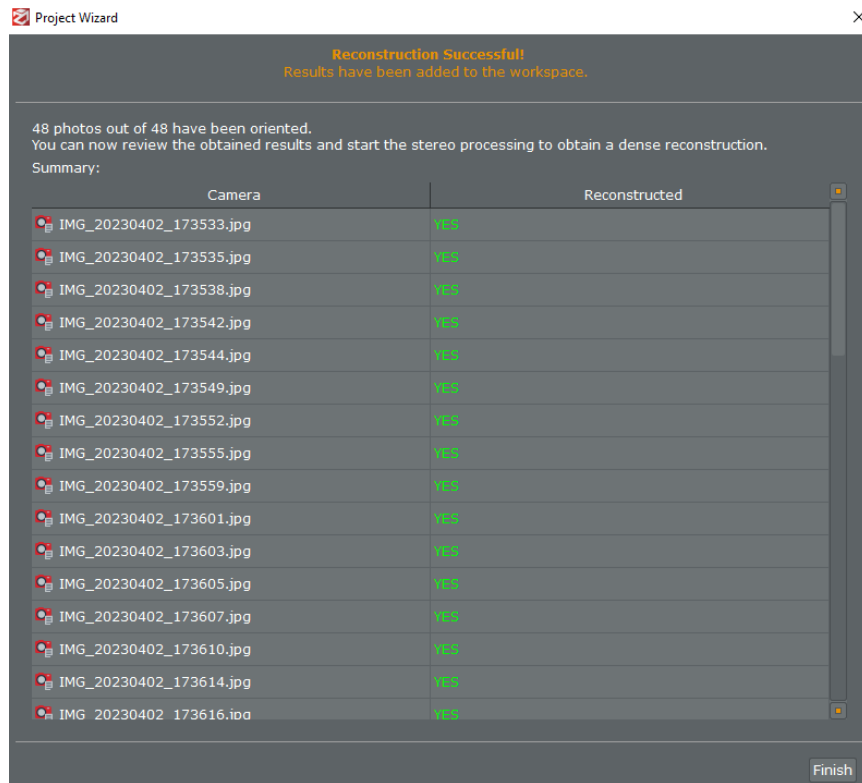


Рис. 2.6. Перелік зображень, які використано для побудови (реконструкції) 3D-моделі

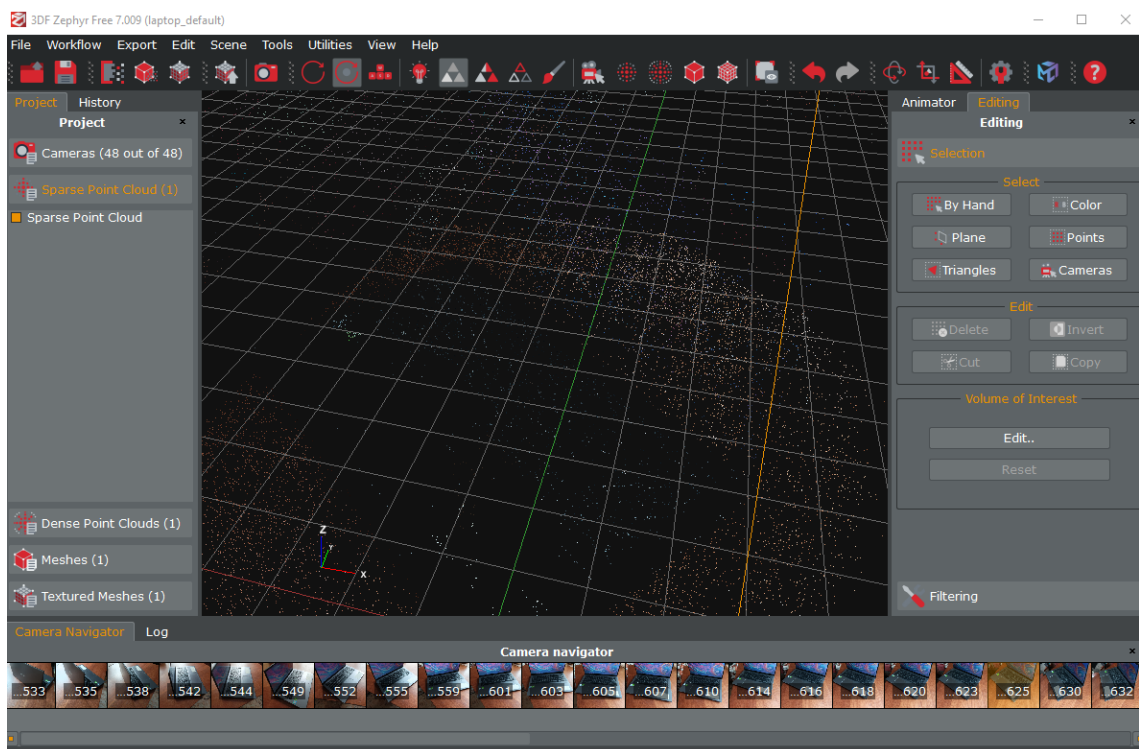


Рис. 2.7. Розріджена хмара точок (Sparse Point Cloud) у системі координат XYZ; синіми маркерами показано положення фотокамер (для кожного зображення з рис. 2.1)

### 2.2.2 Щільна хмара точок

Побудова щільної хмари точок (**Dense Point Cloud**) на основі попередньо визначених параметрів камер (рис. 2.7) та початкових фотографій (рис. 2.1). Побудова щільної хмари точок виконується командою «Workflow/Advanced/ Dense Point Cloud Generation» (рис. 2.8).

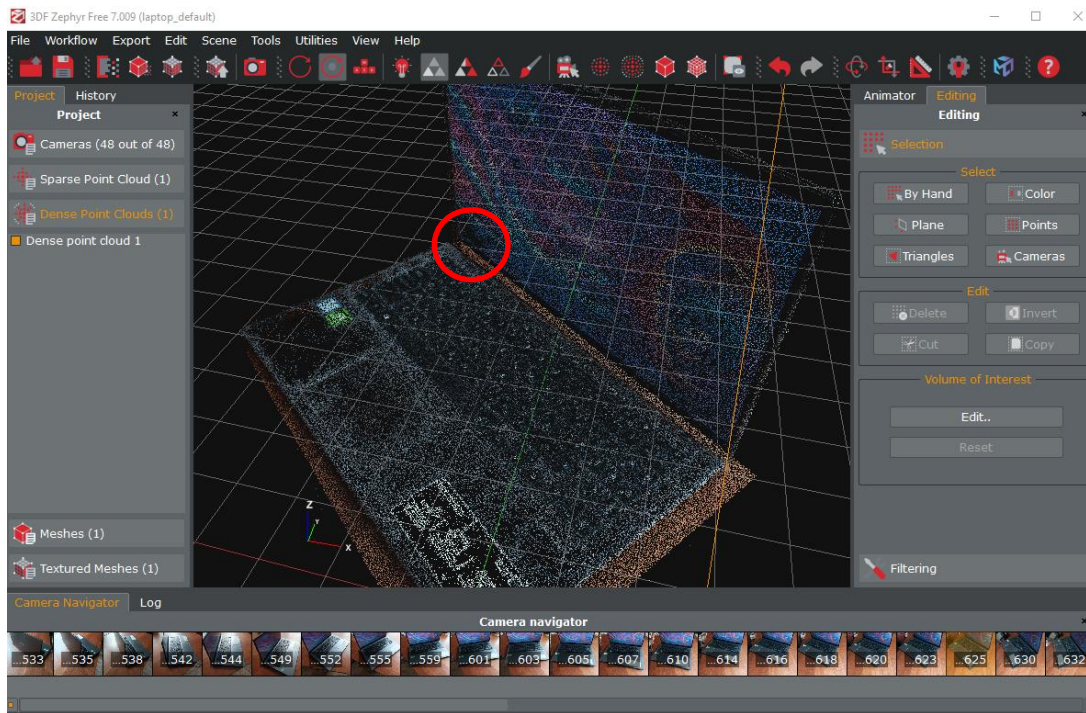


Рис. 2.8. Щільна хмара точок (Dense Point Cloud); параметри побудови – за замовчуванням

З даного ракурсу це не помітно, але в роботі вже є «артефакт» - чорний пробіл у лівій нижній частині монітору (виділено червоним кругом). Далі слідуватиме добування моделі, на якій він не зникне.

### 2.2.3 Створення сітки полігонів

Побудова тривимірної поверхні або сітки полігонів (**Meshes**) на основі щільної хмари точок (Dense Point Cloud) шляхом триангуляції. Сітка полігонів будується командою «Workflow/Advanced/ Mesh Extraction» (рис. 2.9).

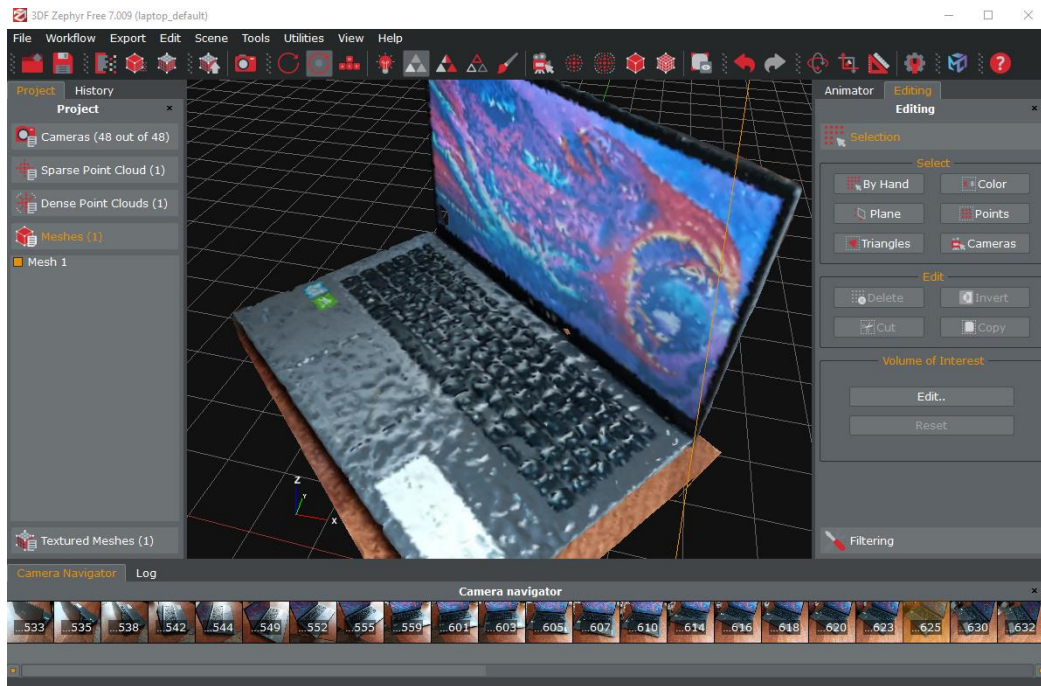


Рис. 2.9. Сітка полігонів (Meshes); параметри побудови – за замовчуванням

## 2.2.4 Накладання текстури

Побудова текстури /текстурної сітки/ (**Textured Meshes**) для полігональної моделі (рис. 2.9) виконується командою WorkFlow/Textured Mesh Generation) (рис. 2.10).

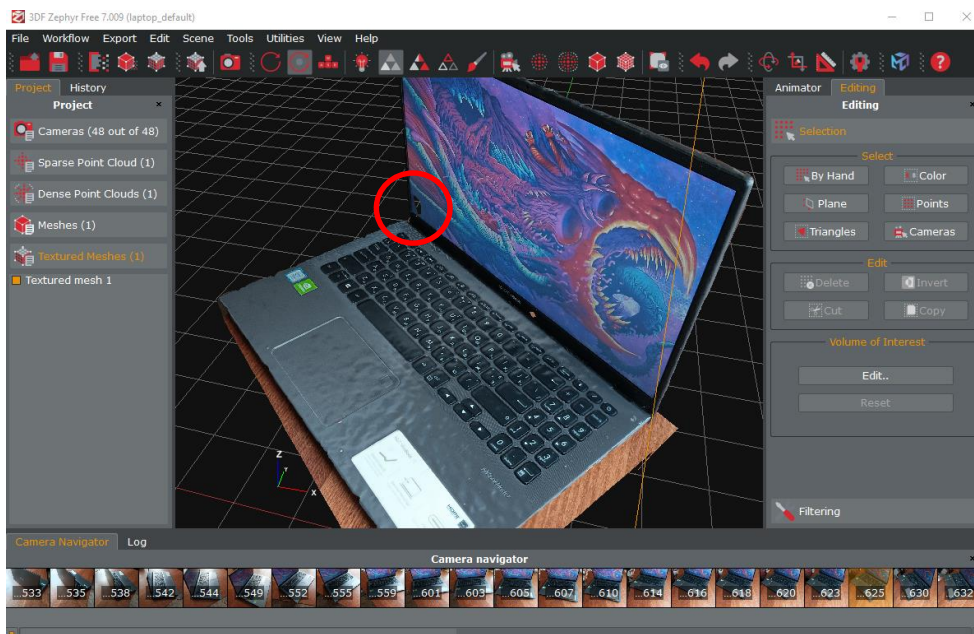


Рис. 2.10. Текстурна сітка (Textured Meshes) для полігональної моделі; параметри побудови – за замовчуванням

Таким чином, побудова 3D моделі (3D Reconstruction) полягає у створенні:

1. Розрідженої хмари точок (Sparse Point Cloud), етап Samantha (рис. 2.7).
2. Щільної хмари точок (Dense Point Cloud), етап Stasia (рис. 2.8).
3. Сітки полігонів (Meshes), етап Sasha (рис. 2.9).
4. Текстурної сітки (Textured Meshes) (рис. 2.10).

На отриманій моделі кількість точок є відносно низькою. Хоча поверхня яскрава та добре освітлена, але присутні дефекти.

## 2.3 Редагування та виміри

### 2.3.1 Зміна вертикальної орієнтації

Вертикальна орієнтація 3D-моделі (в декартовій системі координат  $xuz$ ) виконується командою Tools/ Workspace/ Define Up Vector, наприклад, вказанням напрямку вертикальної осі  $z$  /за двома точками/ (рис. 2.11).

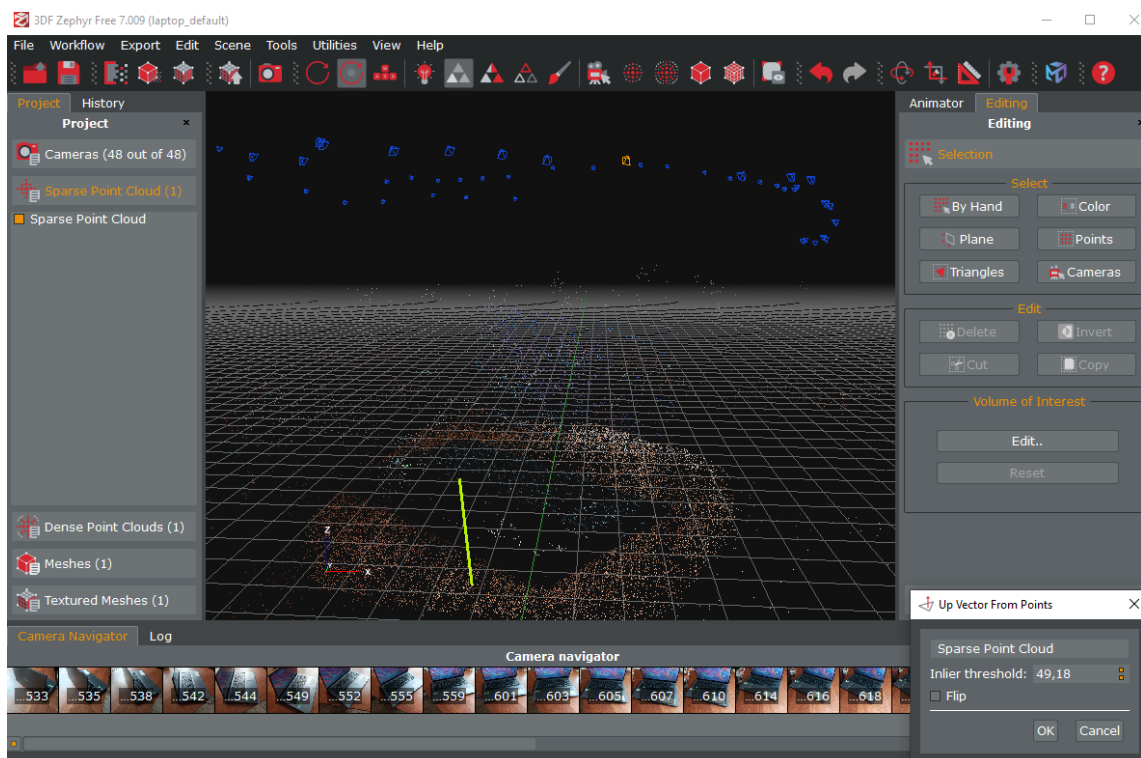


Рис. 2.11. Вказання орієнтації 3D-моделі за допомогою вертикального відрізка в декартовій системі координат  $xuz$ .

### 2.3.2 Видалення зайвих елементів

Виділення зайвих фрагментів моделі виконується командою «Tools/ Selection/ Select by Plane» (рис. 2.12). Після цього виконується видалення виділених фрагментів командою «Tools/ Selection/ Delete Selected Items».

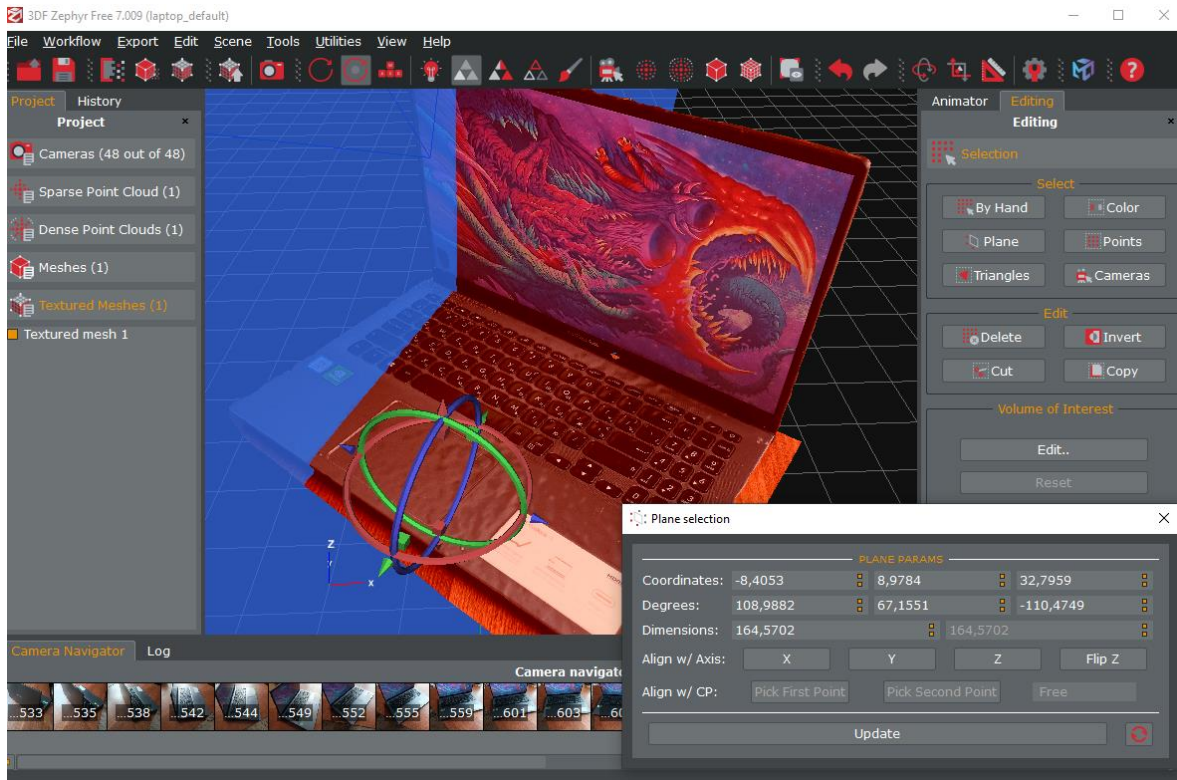


Рис. 2.12. Виділення зайвих фрагментів 3D-моделі командою «Tools/ Selection/ Select by Plane»

### 2.3.3 Зміна за осями координат

Зміна положення, напрямку та масштабу осей координат хуз виконується командою Tools/ Workspace / Scale/ Rotate/Translate Object (рис. 2.13). Поворот моделі навколо осей координат виконується Пр.кл.м. і командою Bounding Box. Команда Smart Reset будує паралелепіпед мінімального об'єму для моделі.

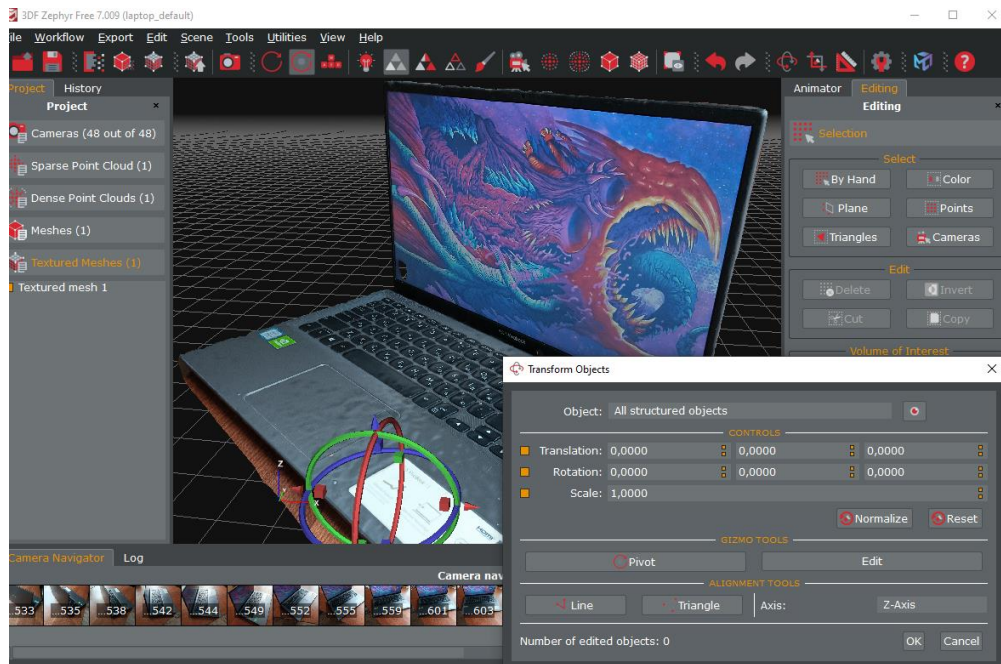


Рис. 2.13. Зміна положення, напрямку та масштабу осей координат хуз

### 2.3.4 Отримання вимірів та властивостей

Виміри розмірів моделі (у метрах) виконуються командою Tools/Distance and Measurements (рис. 2.14).

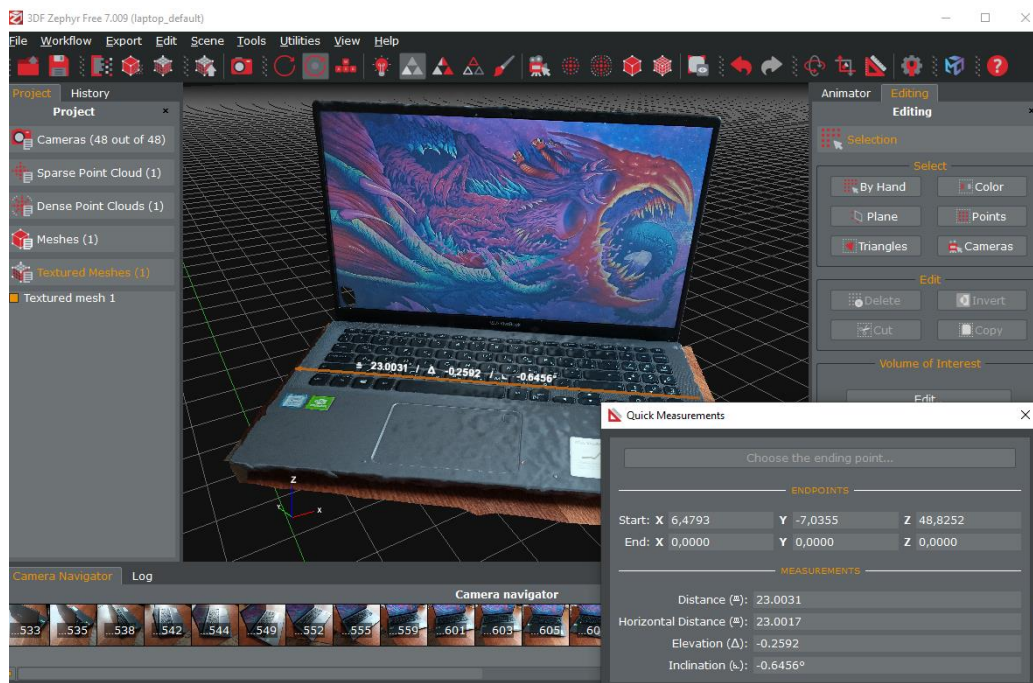


Рис. 2.14. Вимірювання розмірів на 3D-моделі: **D** (Distance) – відстань, м; **E** (Elevation) – підняття, м (різниця висот для початкової і кінцевої точок), **I** (Inclination) – нахил (відрізка відносно горизонтальної площини, у °)

На основі відстані  $D$  (рис.2.14) встановлюється правильний масштаб моделі. Якщо початкова довжина відрізка  $D_0$ , а правильна довжина  $D$ , то для моделі потрібно встановити масштаб  $Scale = D/D_0$  командою «Tools/Workspace/Scale/Rotate/Translate Object» (рис. 2.15).

Наприклад:

Початкова довжина відрізка  $D_0 = 2.1167$  м; дійсне значення  $D = 2.02$  м.

Тоді  $Scale = D/ D_0 = 2.02/ 2.1167 = 0.95431568$  (рис. 2.15).

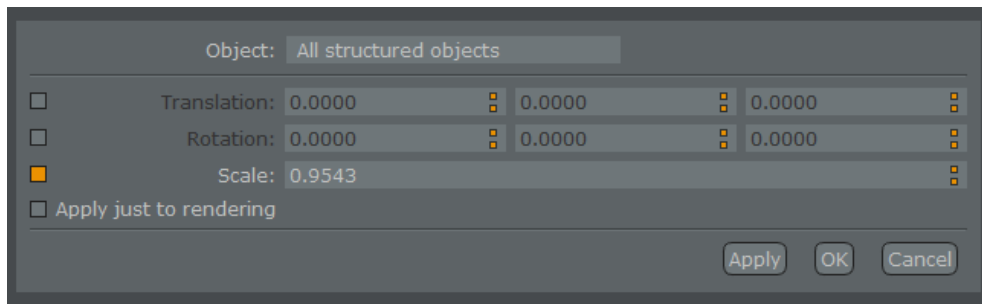


Рис. 2.15. Встановлення масштабу 3D-моделі

Властивості 3D-моделі отримуються через контекстне меню об'єкту Textured mesh (рис. 2.16).

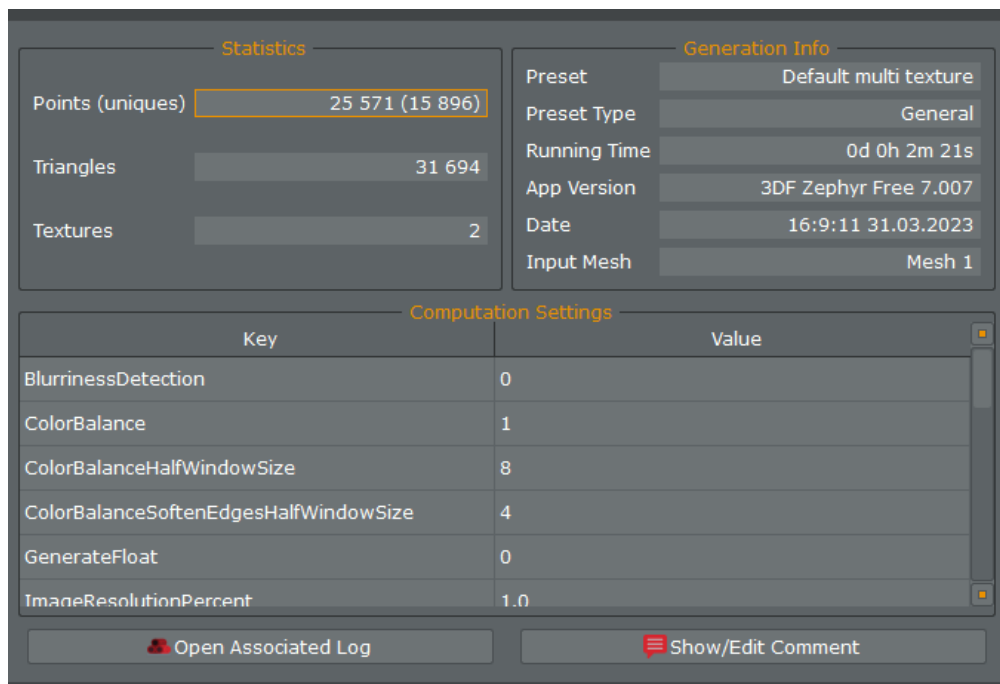


Рис. 2.16. Властивості 3D-моделі



За фотографією можна вибрати відповідне їй положення камери (рис. 2.17).

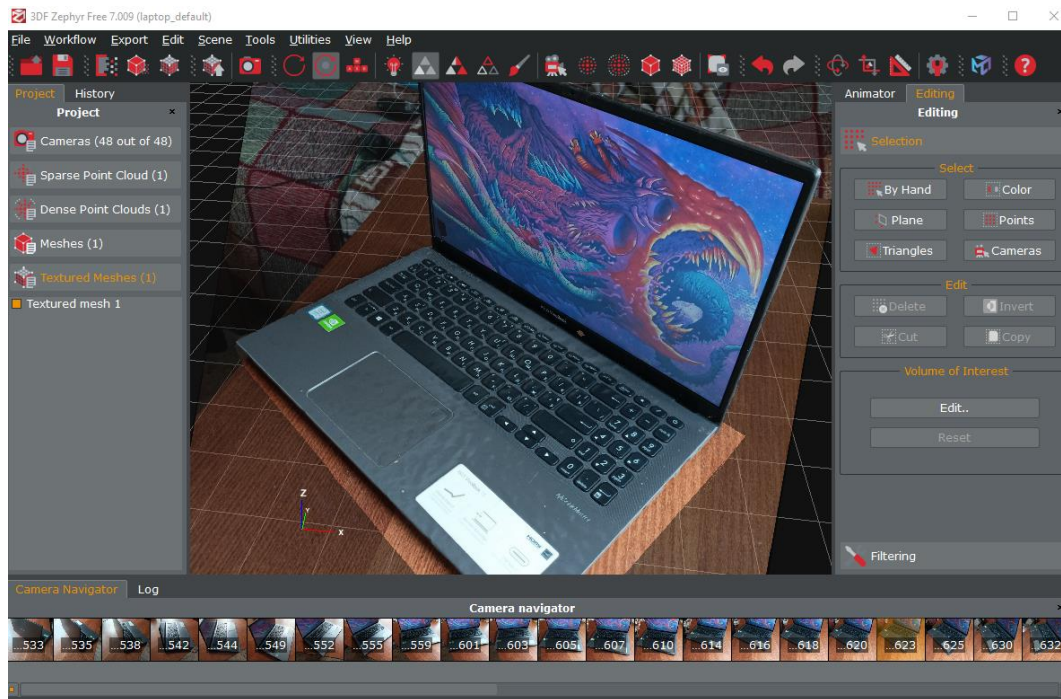


Рис. 2.17. Фотографія та відповідне їй положення фотокамери

Для 3D-моделі триангуляційна сітка відображається командою Rendering/Enable Wireframe з контекстного меню (рис. 2.18).

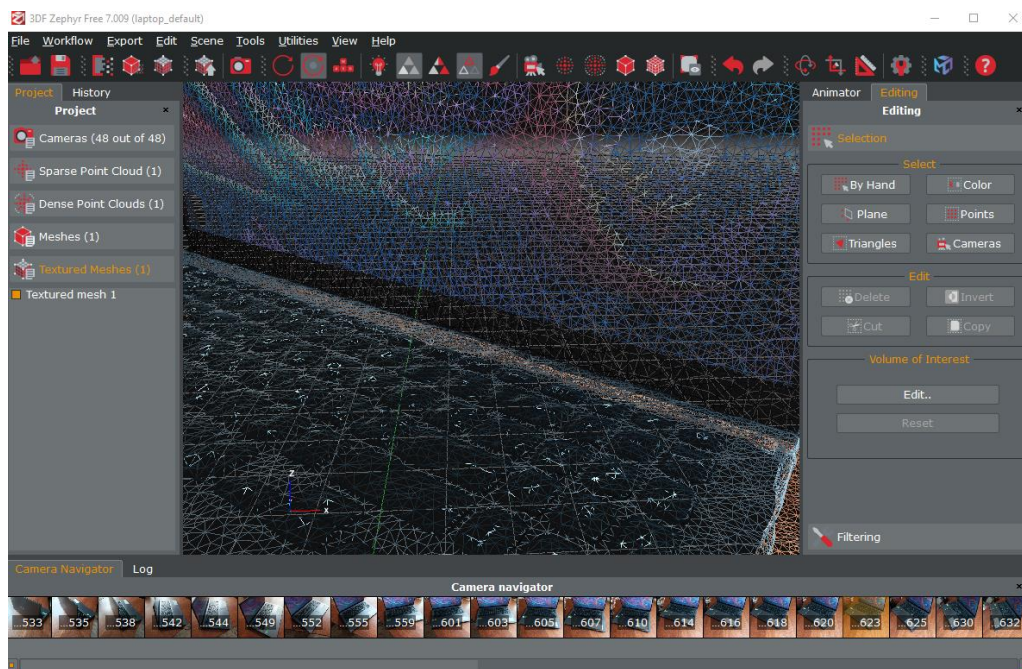


Рис. 2.18. 3D-модель з відображенням триангуляційної сітки (Wireframe)

### 2.3.5 Формати файлів

Шляхом повороту вибирається потрібний ракурс моделі. 3D модель зберігається у файлі формату .zep. Сітка зберігається командою Export / Export Mesh (формати .ply, Obj, Glb, Upload to Sketchlab).

Побудова 3D моделі займає значний час. Наприклад, для комп'ютера з параметрами (процесор AMD A4-6300, 3.70 ГГц, 2 ядра; жорсткий диск Seagate Barracuda 7200, 1 TB; SSD 120 ГБ; ОЗП 4 ГБ; відеокарта AMD Radeon HD8370D /інтегрована/) час побудови 3D моделі – 12 хв.

На всіх етапах був присутній «артефакт». У наступному підрозділі будуть встановлені високі параметри майстра. Вони збільшать кількість та щільність точок і можна буде побачити – чи виправить це даний пробіл.

### 2.4 Представлення високоякісної моделі

У програмі є можливість обробити зображення за високими ('high') параметрами. Параметри при кожній дії встановлюються окремо.

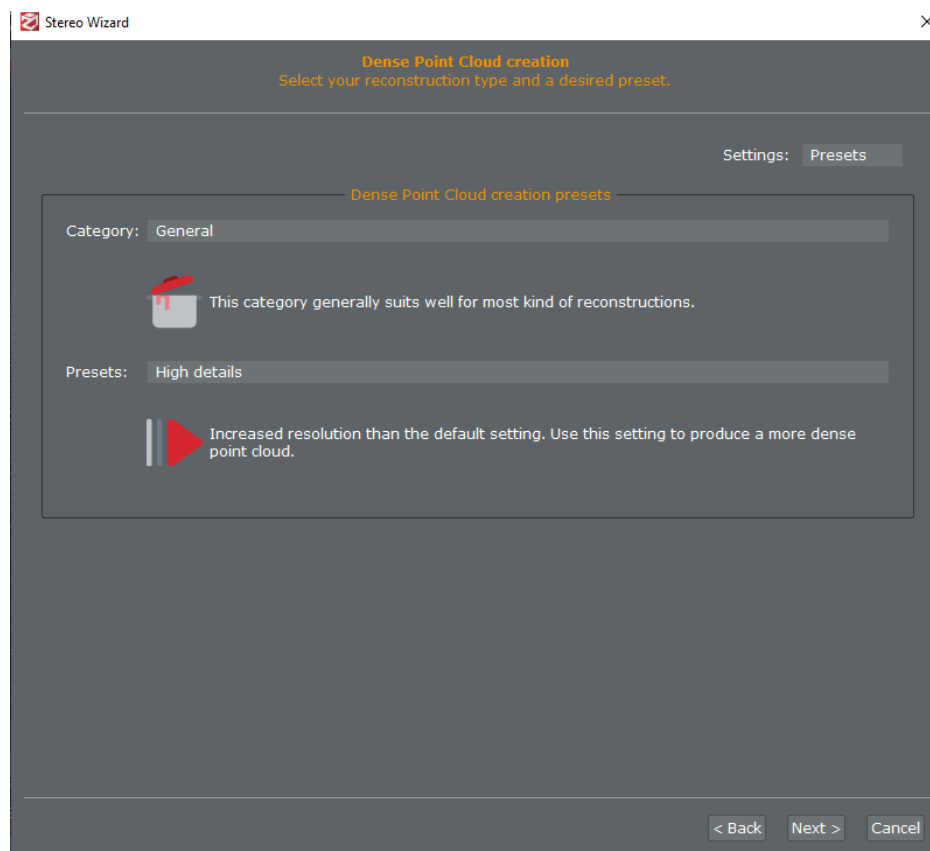


Рис. 2.19. Встановлення високих параметрів

Це налаштування виконується однаково для всіх процесів.

Таким чином програма обробить більше точок, які у хмарі будуть розміщатись щільніше, а полігональна та текстурна сітка будуть відповідно більш деталізовані. Зображення 2.20 – 2.25 показують модель, отриману при високих параметрах майстра, яку можна порівняти із отриманною раніше.

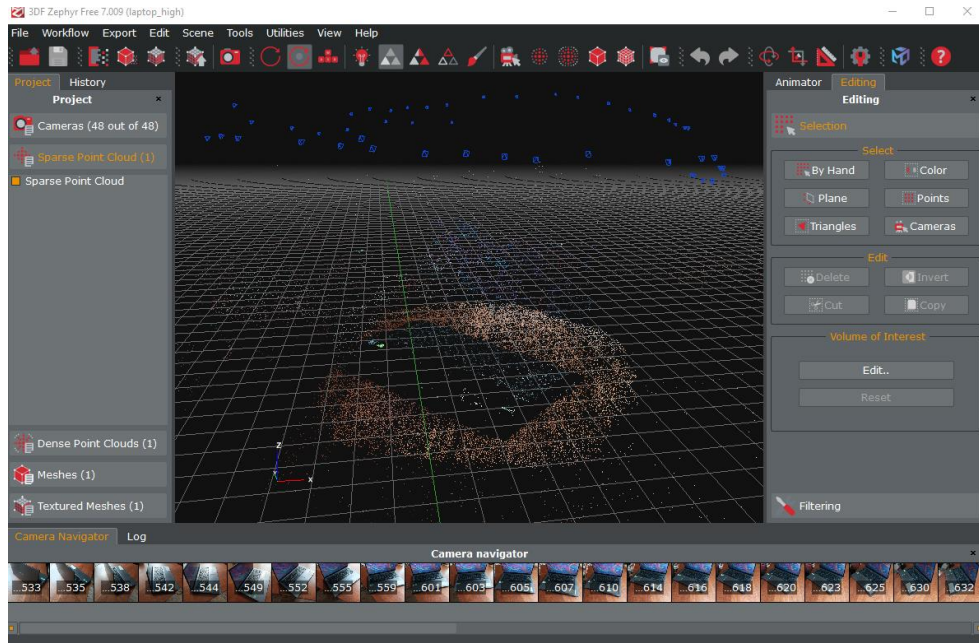


Рис. 2.20. Розріджена хмара точок

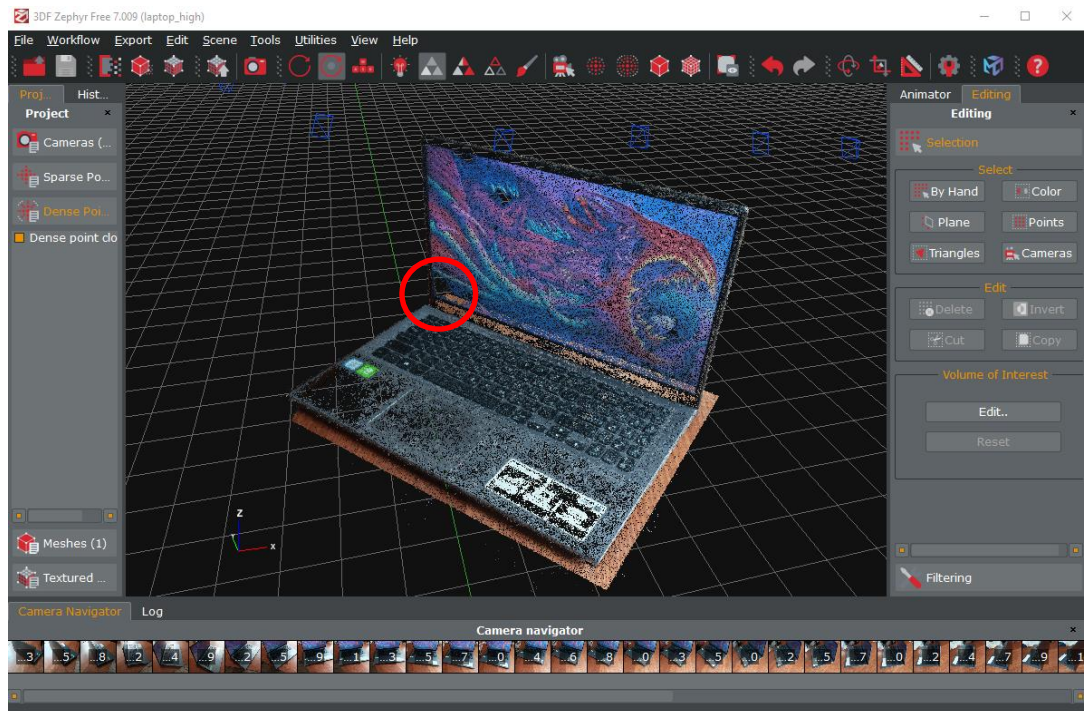


Рис. 2.21. Щільна хмара точок

Вже на даному етапі знову видно, що пробіл у моніторі не зник.

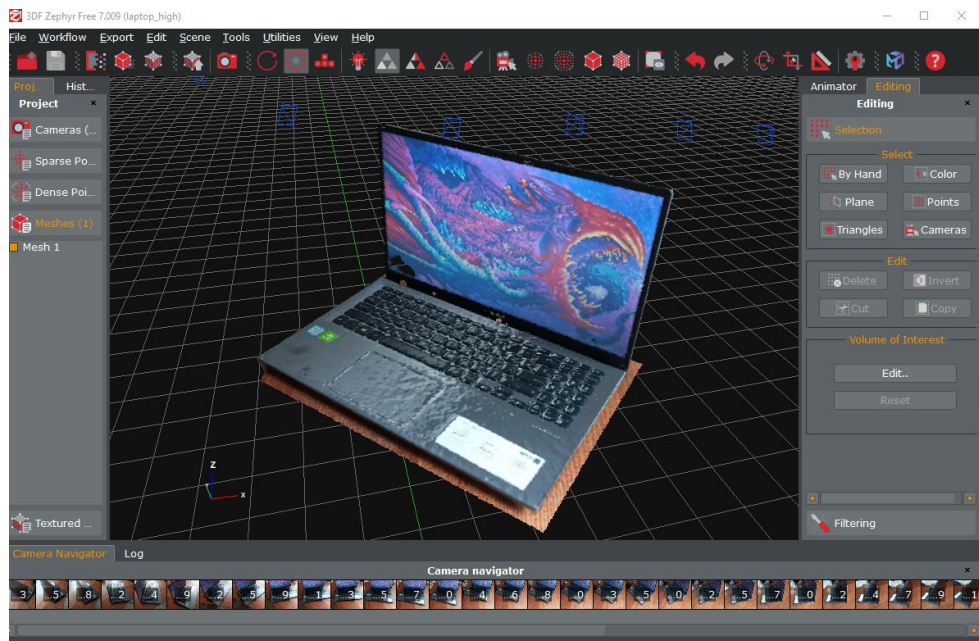


Рис. 2.22. Сітка полігонів

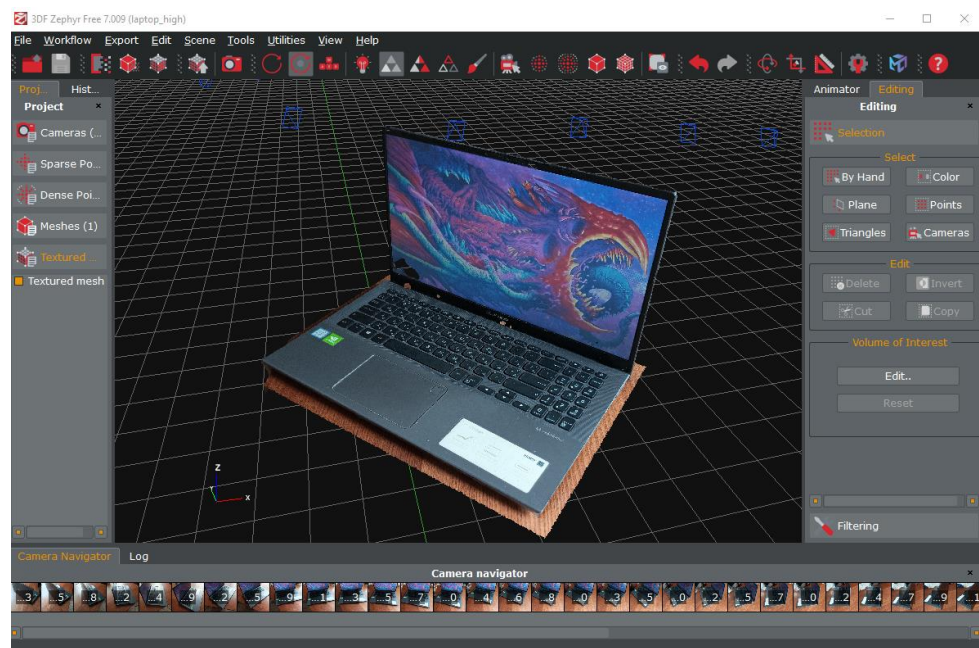


Рис. 2.23. Текстурна сітка

Однак є різниця у властивостях моделі та у вигляді триангуляційної сітки. Точок стало набагато більше і текстура більш деталізована.

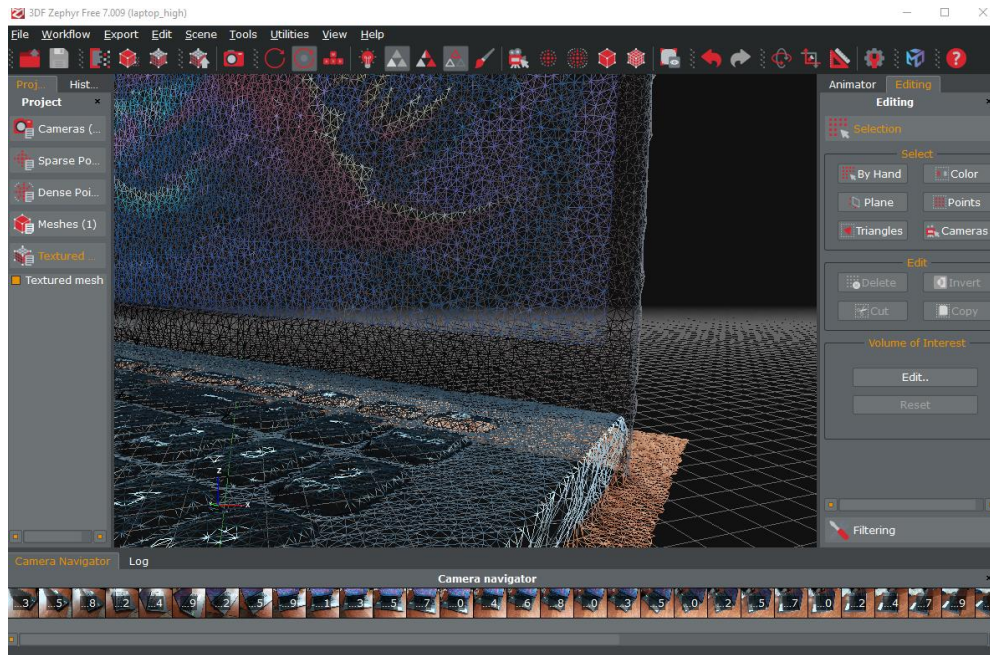


Рис. 2.24. Триангуляційна сітка на високих налаштуваннях

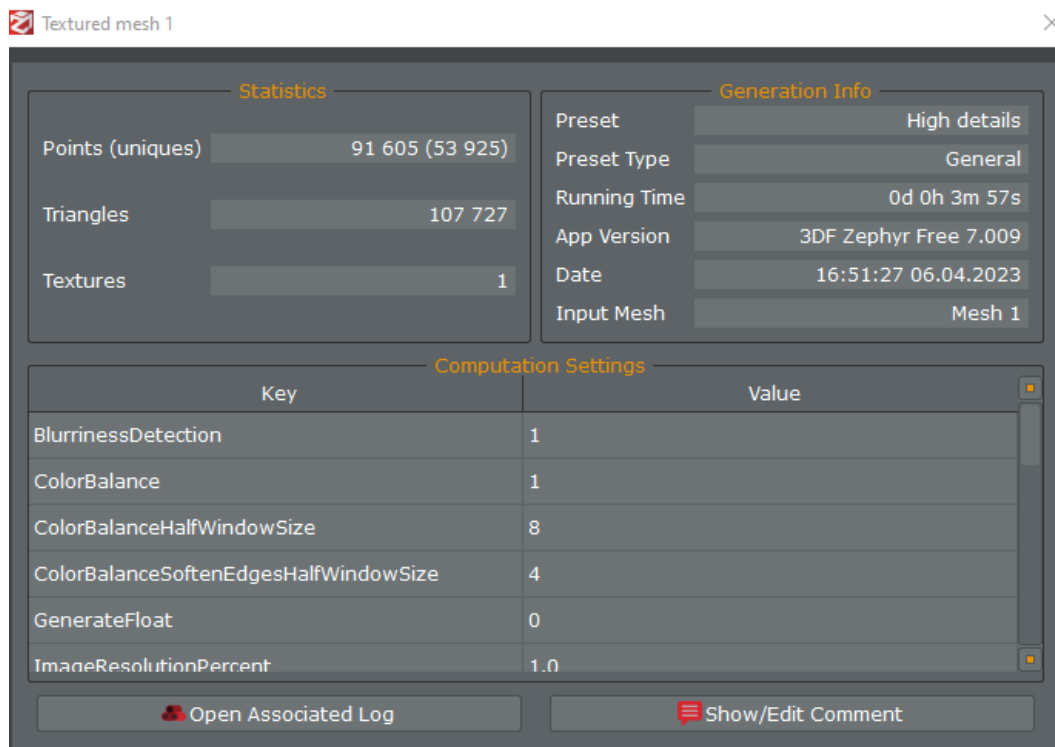


Рис. 2.25. Властивості моделі

Отже, за одним й тим самим набором фотографій було отримано дві різні моделі. Розглянуто по-етапне проектування «шар за шаром».

## **Висновки до розділу 2**

За допомогою програми 3DF Zephyr побудовано тривимірні моделі об'єктів методом фотограмметрії. Аналіз побудованих тривимірних моделей показав, що в них присутні певні артефакти. Тому існує потреба у створенні СПР, яка б надавала рекомендації при побудові моделі.

Як недолік можна відмітити «хвилясту» поверхню корпусу ноутбука, фотографії якого були отримані при прямому сонячному світлі. Пряме освітлення може приводити до появи біків, особливо на гладких поверхнях.

Наступний розділ доповнює отриману інформацію аналізом ще кількох моделей та описує систему рекомендацій на основі всіх отриманих даних.

## РОЗДІЛ 3

### ДОДАТКОВІ ДОСЛІДЖЕННЯ ТА ЕКСПЕРТНА СИСТЕМА ПРАВИЛ

#### 3.1 Представлення еталонної моделі

##### 3.1.1 Побудова за стандартними параметрами

Для більш повного досвіду та можливості порівняння було проаналізовано ще два набори фотографій: для ноутбука із вимкненим монітором та для шкатулки із зернистою поверхнею.

Програмою 3DF Zephyr було створено ще по дві моделі для кожного набору – одна з стандартними налаштуваннями, а інша з високими параметрами.

Спочатку описується найбільш успішна модель – шкатулка.

Побудова моделі починається з отримання серії знімків (рис. 3.1).

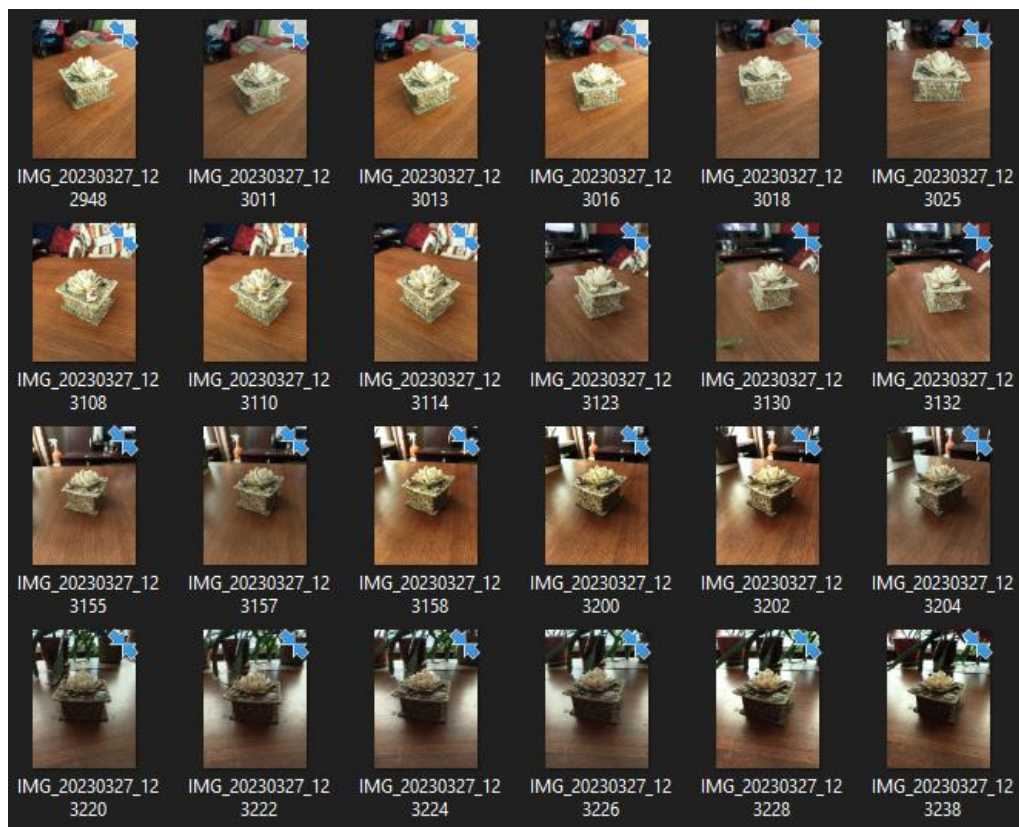


Рис. 3.1. Серія з 49 початкових зображень об'єкта  
(розміром  $1560 \times 2080$  пікселів)

Далі йдуть етапи, ідентичні до описаних у попередньому розділі. Опис в них ідентичний і замість великих підрозділів все поділено на етапи.

Для побудови 3D-моделі об'єкту створюється новий проект (командою WorkFlow/ New Project) (рис. 3.2).

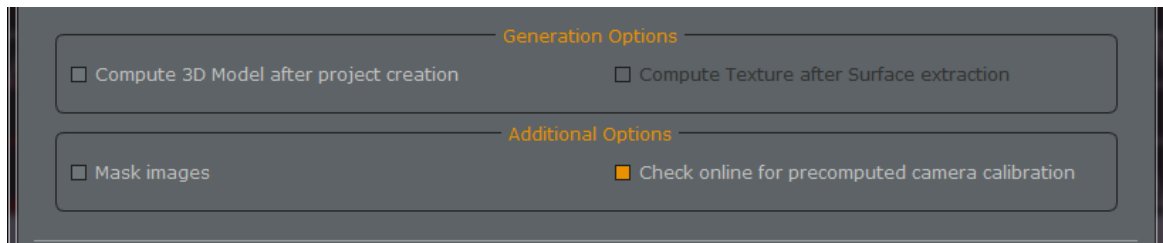


Рис. 3.2. Опції проекту побудови 3D-моделі об'єкту

Отримані фотографії об'єкту (рис. 3.1) додаються до проекту (рис. 3.3). Після цього виконується автокалібрування параметрів фотокамер (рис. 3.4).

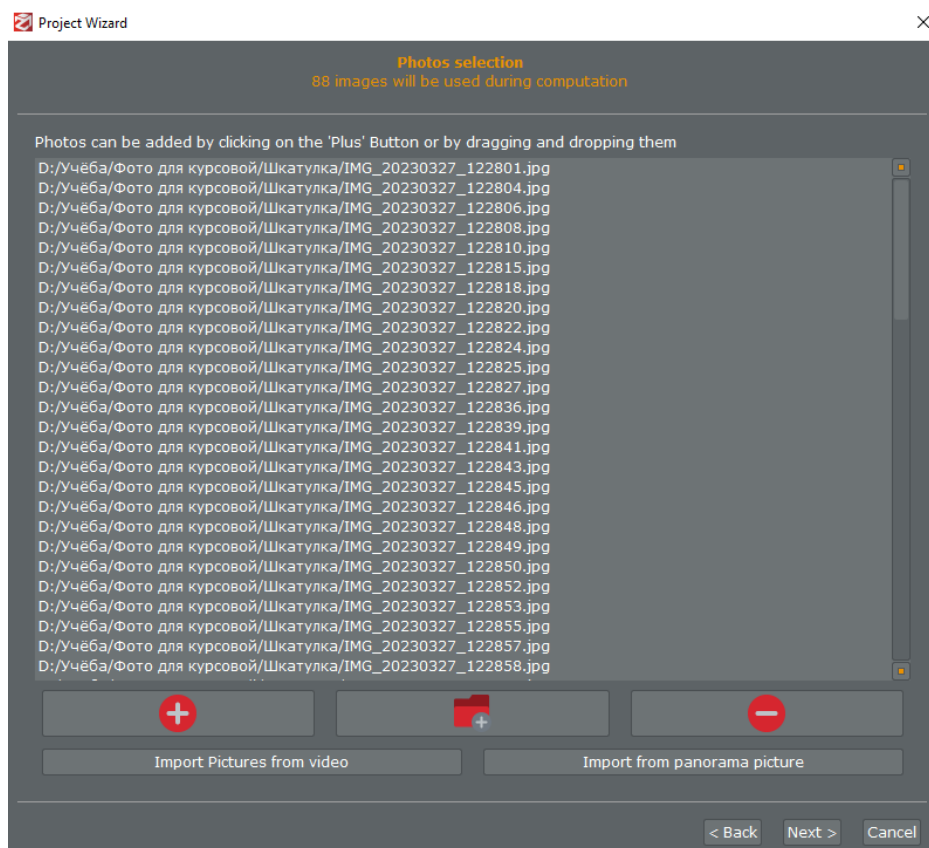


Рис. 3.3. Додавання фотографій до проекту



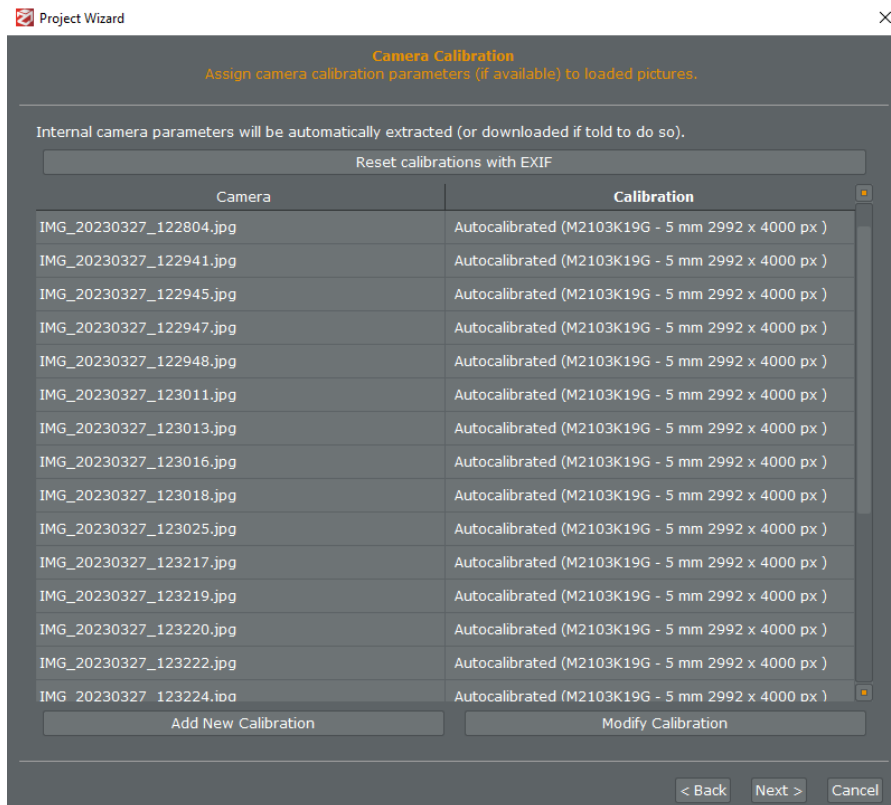


Рис. 3.4. Автокалібрування параметрів фотокамер

Проект побудови 3D-моделі об'єкту містить 4 етапи:

**Етап 1.** Визначення положення та орієнтації камер, а також параметрів внутрішньої геометрії камер (фокусної відстані, дисторсії та ін.) (рис. 3.5). Для цього програма визначає ключові точки (keypoints) зображень та спільні ключові точки різних зображень шляхом суміщення. Для побудови 3D-моделі не використовуються зображення, які погано перекриваються з іншими, або зображення візуально подібних фрагментів (рис. 3.6). У результаті обчислюється розріджена хмара точок (**Sparse Point Cloud**) в 3D просторі моделі, а також координати та орієнтації фотокамер (рис. 3.7).

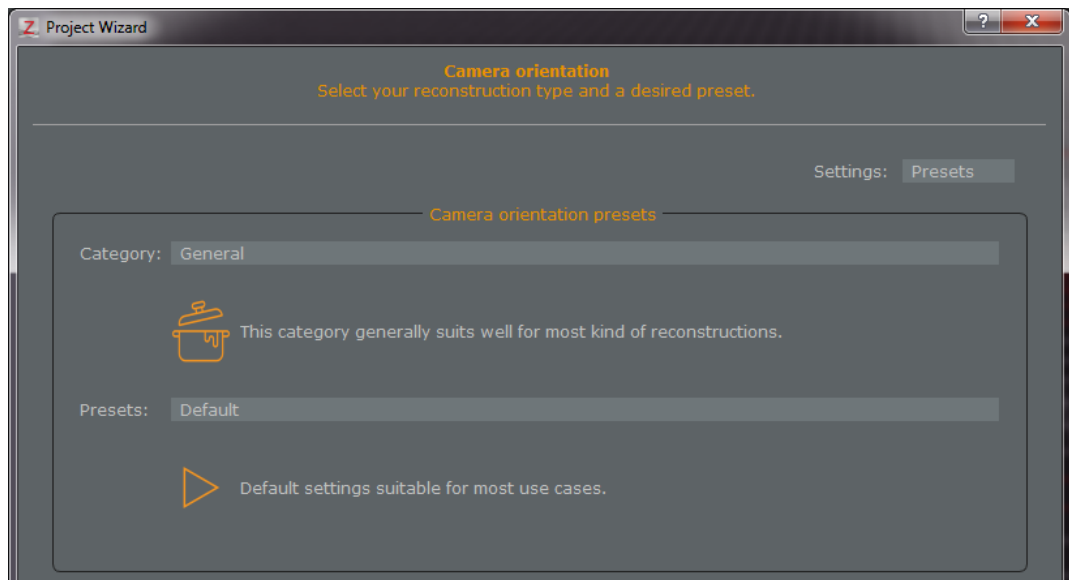


Рис. 3.5. Параметри майстра (за замовчуванням), які використовуються при визначенні положення і орієнтації камер

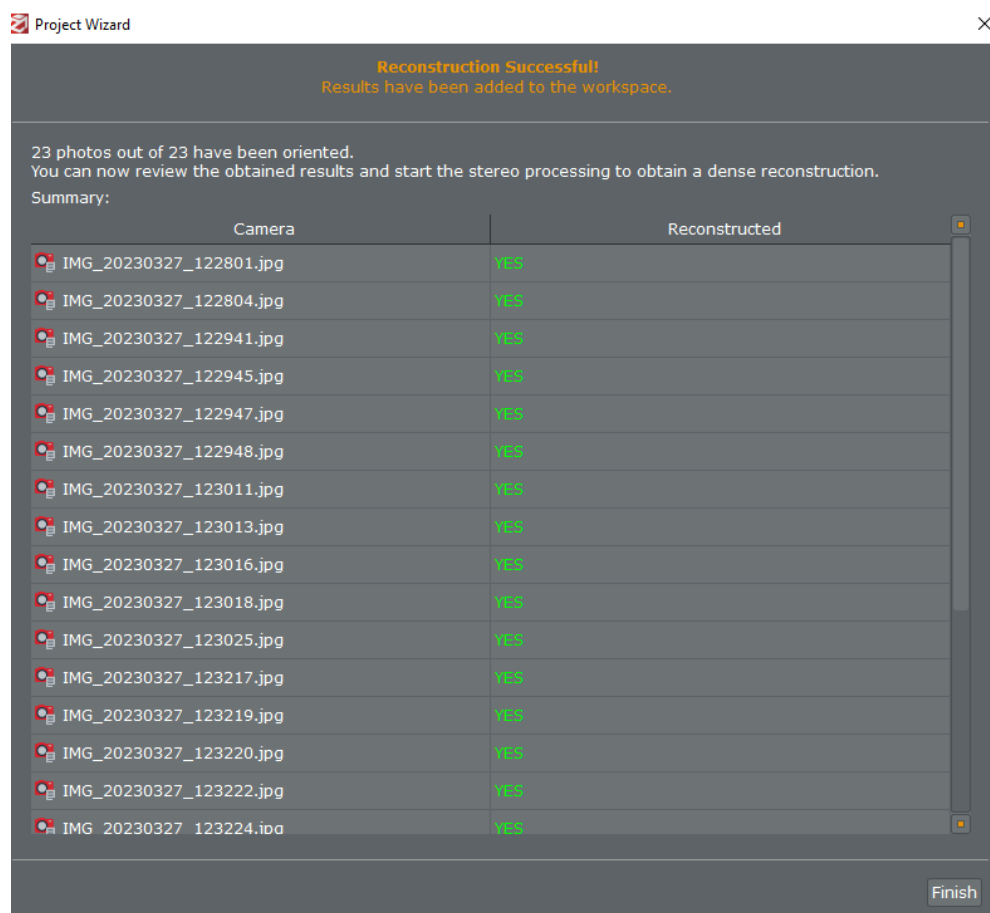


Рис. 3.6. Перелік зображень, які використано для побудові (реконструкції) 3D-моделі

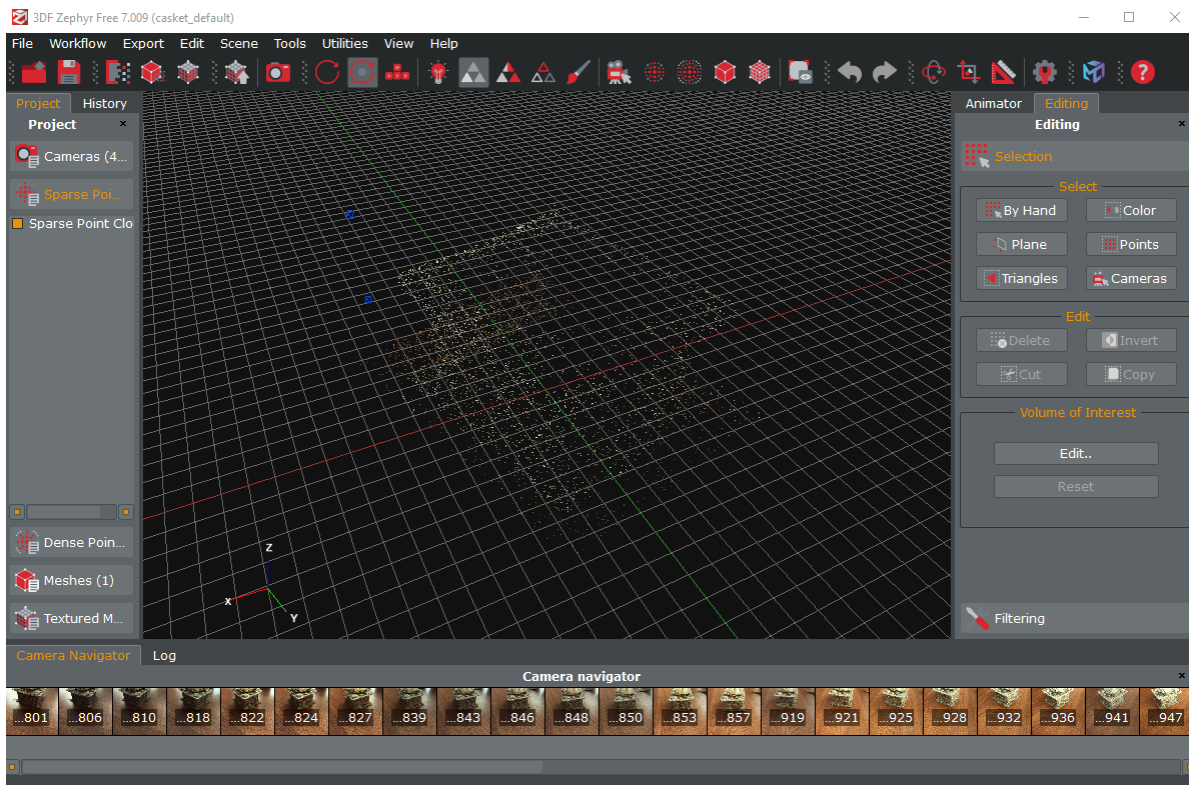


Рис. 3.7. Розріджена хмара точок (Sparse Point Cloud) у системі координат XYZ; синіми маркерами показано положення фотокамер (для кожного зображення з рис. 3.1)

**Етап 2.** Побудова щільної хмари точок (**Dense Point Cloud**) на основі попередньо визначених параметрів камер (рис. 3.7) та початкових фотографій (рис. 3.1). Побудова щільної хмари точок виконується командою «Workflow/Advanced/ Dense Point Cloud Generation» (рис. 3.8).

**Етап 3.** Побудова тривимірної поверхні або сітки полігонів (**Meshe**) на основі щільної хмари точок (Dense Point Cloud) шляхом триангуляції. Сітка полігонів будується командою «Workflow/Advanced/ Mesh Extraction» (рис. 3.9).

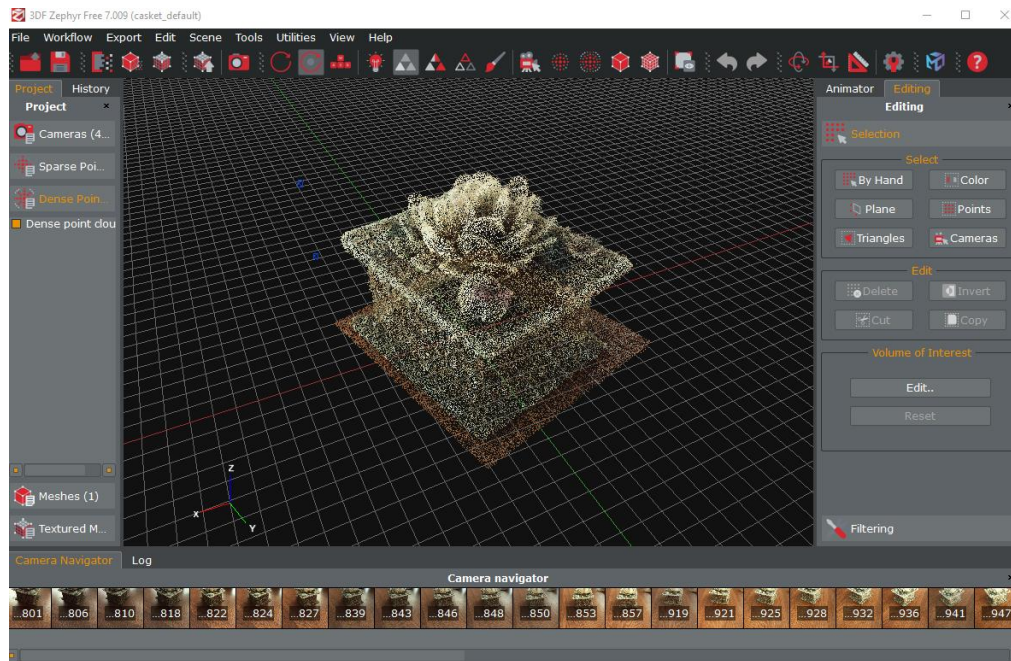


Рис. 3.8. Щільна хмара точок (Dense Point Cloud); параметри побудови – за замовчуванням

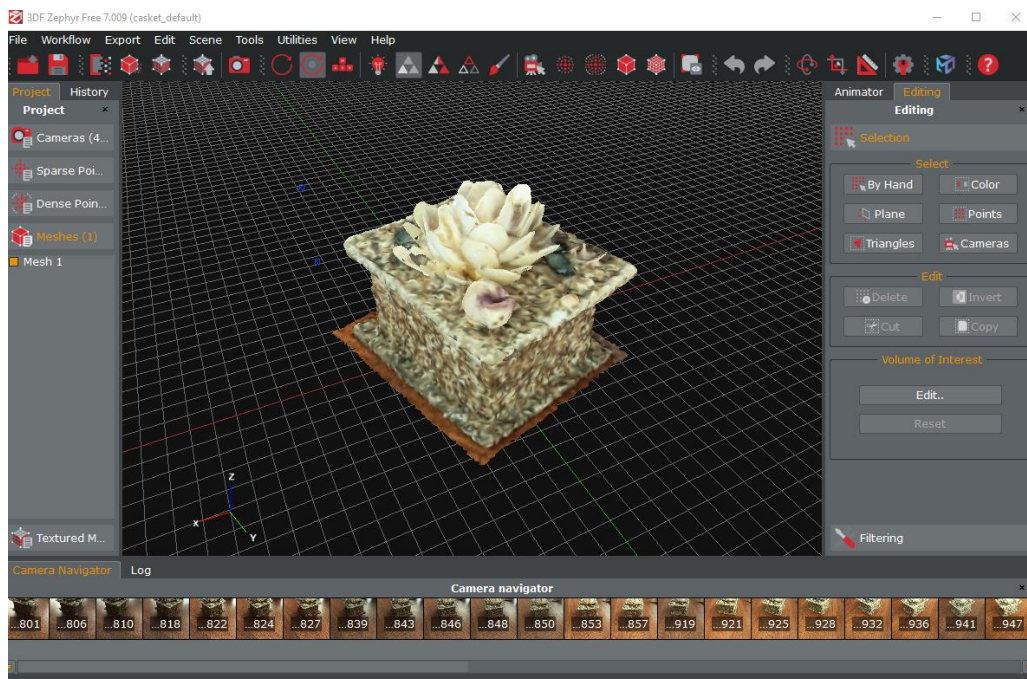


Рис. 3.9. Сітка полігонів (Meshes); параметри побудови – за замовчуванням

**Етап 4.** Побудова текстури /текстурної сітки/ (**Textured Meshes**) для полігональної моделі (рис. 3.9) виконується командою **WorkFlow/Textured Mesh Generation**) (рис. 3.10).

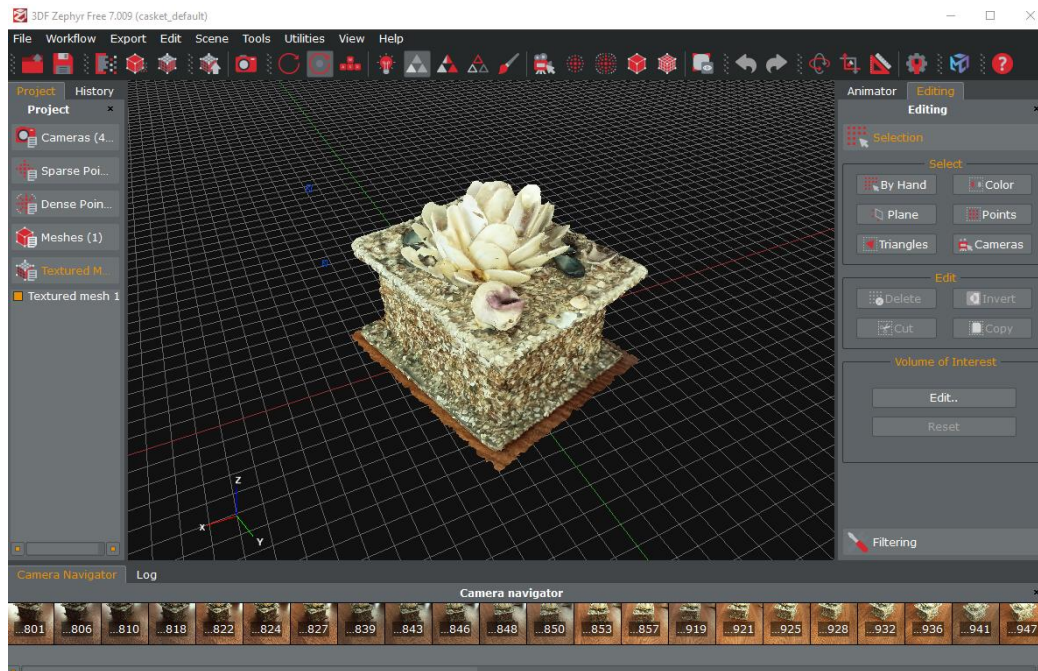


Рис. 3.10. Текстурна сітка (Textured Meshes) для полігональної моделі;  
параметри побудови – за замовчуванням

Як можна бачити на скріншотах – ніяких артефактів та знижень якості на жодному з етапів не було. При огляді моделі з усіх боків їх також не було виявлено. Освітлення, час, поверхня, на якій був об'єкт – все ідентичне.

Свою роль зіграв розмір та форма. Шкатулка – це невеликий прямокутний об'єкт, простий для обробки і який легко розміщується у середині кадру. Ноутбук, порівняно зі шкатулкою, більш габаритний, має гладку, частково відбиваючу поверхню та у відкритому стані відкидує тінь, що є негативним фактором при зчитуванні з фотографій. Необхідно було обрати правильний кут падіння світла, щоб все було коректно. Ще одним негативним, але менш значним фактором був розмір – ноутбук не більше стола, на якому він стояв, але чим більше сторонніх об'єктів потрапляє у камеру та чим вони світліші чи темніші – тим більше змінних доводиться враховувати при постановці кадру та відбиранні фотографій.

Отриманий досвід побудови моделей у подальшому перетворено на факти-умови, наслідки та рекомендації для СПР.

### 3.1.2 Побудова за високими параметрами

Проведено подальшу обробку тривимірних моделей.

Вертикальна орієнтація 3D-моделі (в декартовій системі координат  $xuz$ ) виконується командою **Tools/ Workspace/ Define Up Vector**, наприклад, вказанням напрямку вертикальної осі  $z$  /за двома точками/ (рис. 3.11).

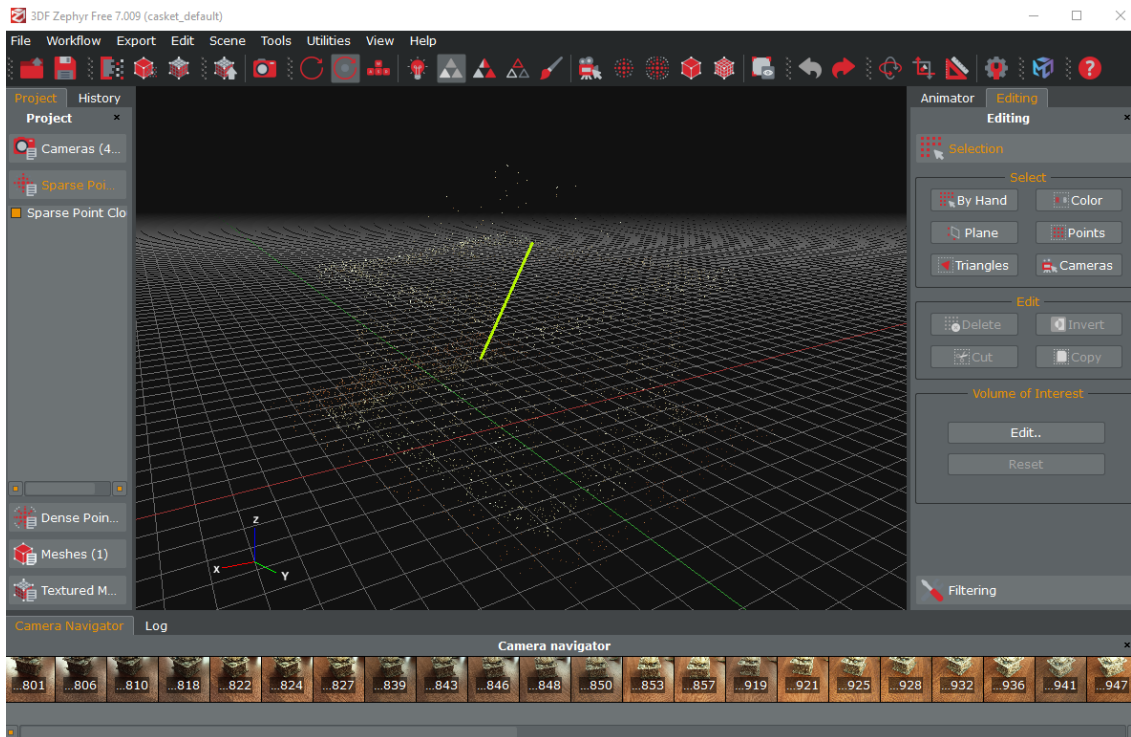


Рис. 3.11. Вказання орієнтації 3D-моделі за допомогою вертикального відрізка в декартовій системі координат  $xuz$

Виділення зайвих фрагментів моделі виконується командою «**Tools/ Selection/ Select by Plane**» (рис. 3.12). Після цього виконується видалення виділених фрагментів командою «**Tools/ Selection/ Delete Selected Items**».

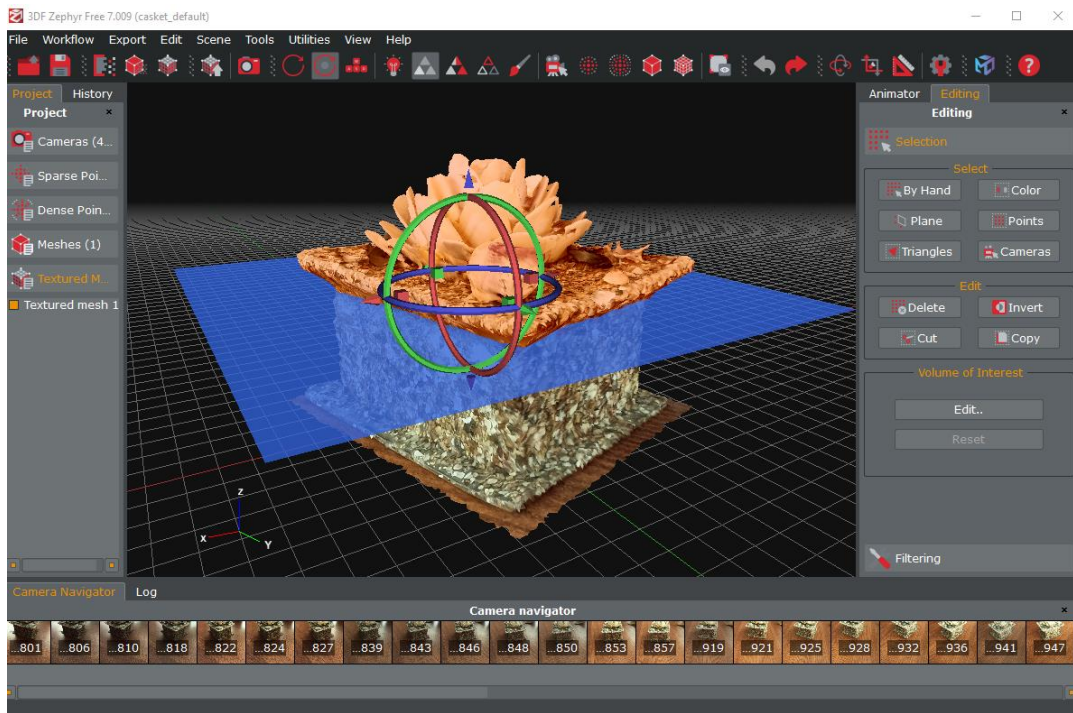


Рис. 3.12. Виділення зайвих фрагментів 3D-моделі  
командою «Tools/ Selection/ Select by Plane»

Зміна положення, напрямку та масштабу осей координат хуз виконується командою Tools/ Workspace / Scale/ Rotate/Translate Object (рис. 3.13). Поворот моделі навколо осей координат виконується Пр.кл.м. і командою Bounding Box. Команда Smart Reset будує паралелепіпед мінімального об'єму для моделі.

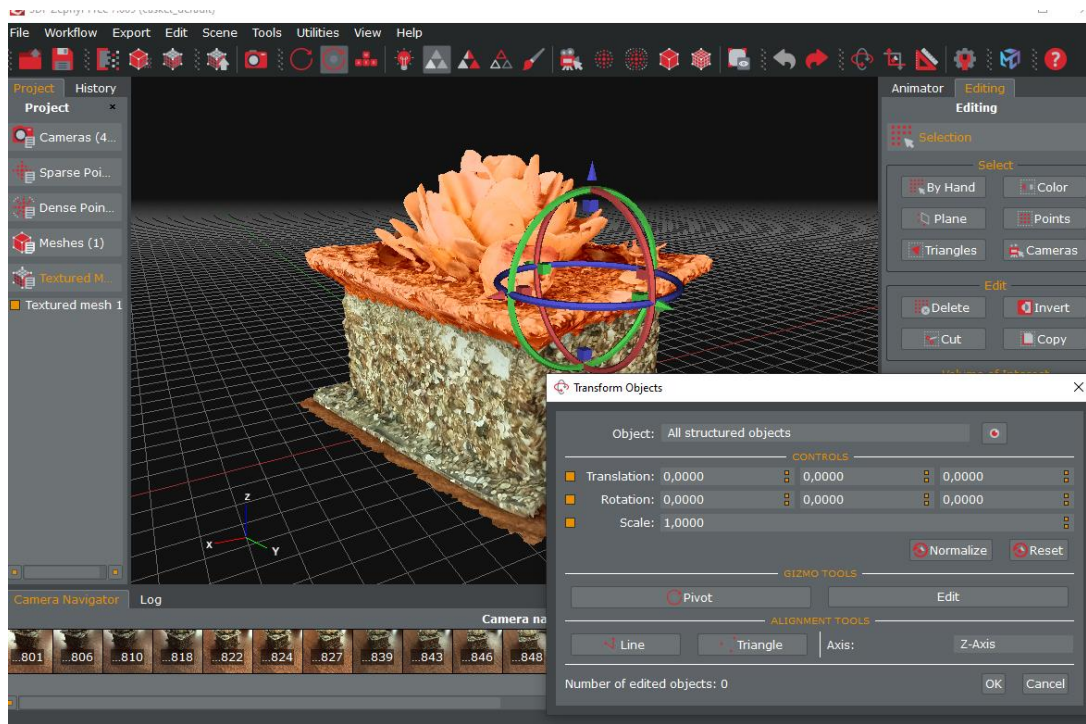


Рис. 3.13. Зміна положення, напрямку та масштабу осей координат хуз

Виміри розмірів моделі (у метрах) виконуються командою Tools/Distance and Measurements (рис. 3.14).

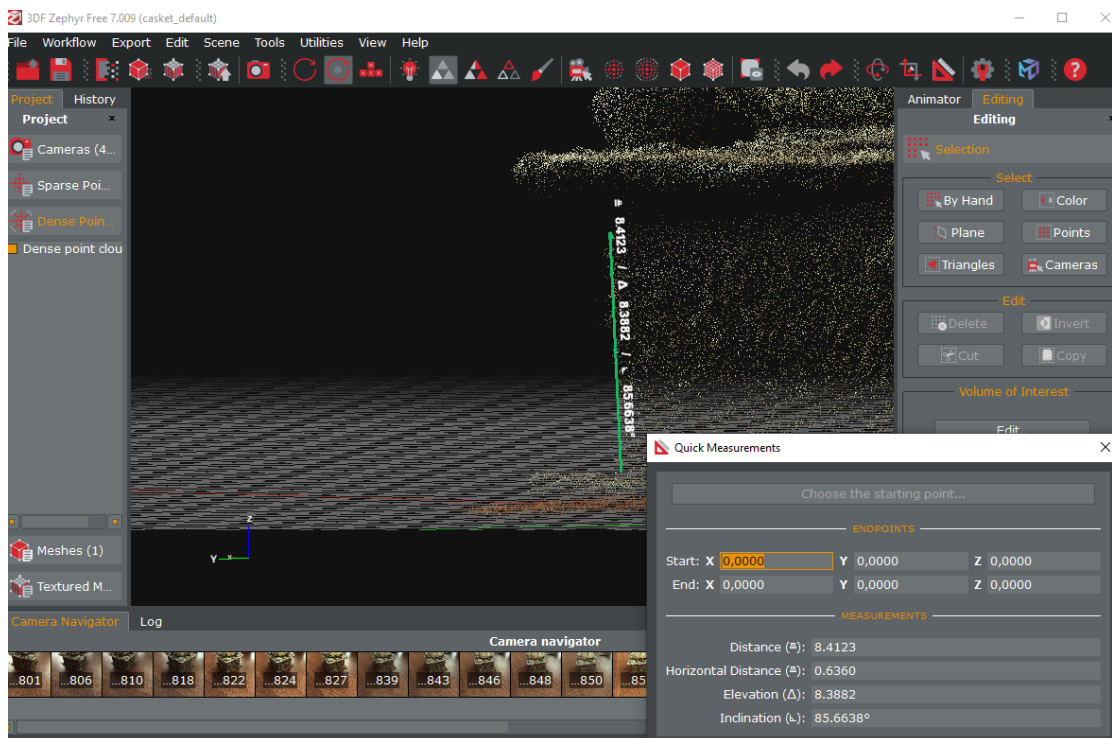


Рис. 3.14. Вимірювання розмірів на 3D-моделі: **D** (Distance) – відстань, м; **E** (Elevation) – підняття, м (різниця висот для початкової і кінцевої точок), **I** (Inclination) – нахил (відрізка відносно горизонтальної площини, у °)



На основі відстані  $D$  (рис. 3.14) встановлюється правильний масштаб моделі. Якщо початкова довжина відрізка  $D_0$ , а правильна довжина  $D$ , то для моделі потрібно встановити масштаб  $Scale = D/D_0$  командою «Tools/Workspace/Scale/Rotate/Translate Object» (рис. 3.15).

Наприклад:

Початкова довжина відрізка  $D_0 = 2.1167$  м; дійсне значення  $D = 2.02$  м.

Тоді  $Scale = D/D_0 = 2.02/2.1167 = 0.95431568$  (рис. 3.15).

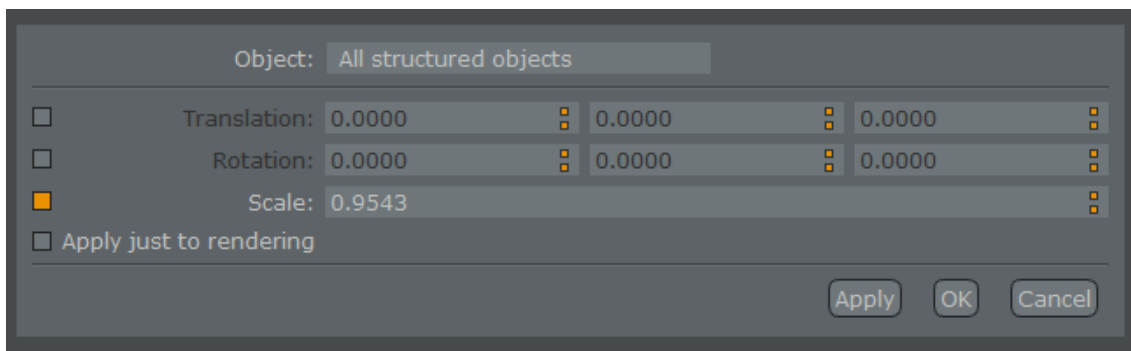


Рис. 3.15. Встановлення масштабу 3D-моделі

Властивості 3D-моделі отримуються через контекстне меню об'єкту Textured mesh (рис. 3.16).

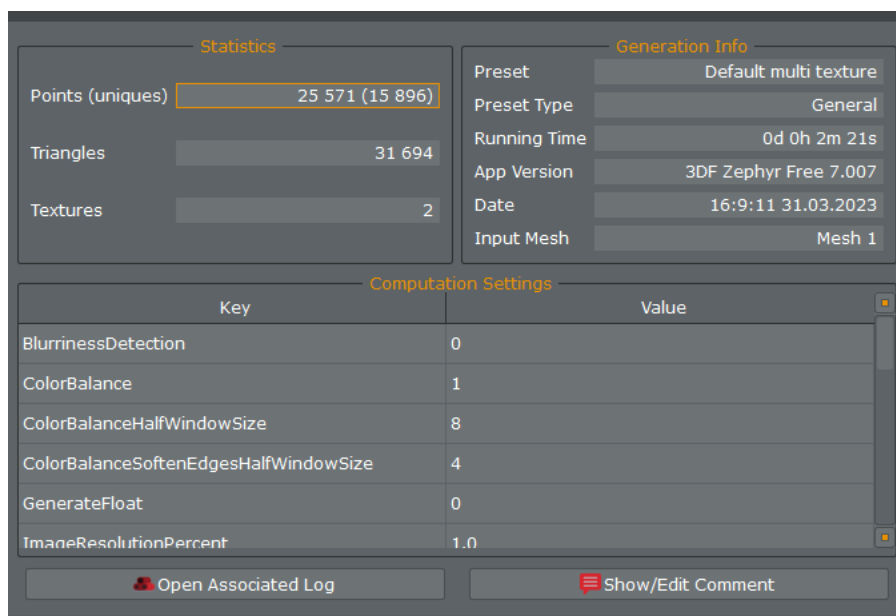


Рис. 3.16. Властивості 3D-моделі

За фотографією можна вибрати відповідне їй положення камери (рис. 3.17).

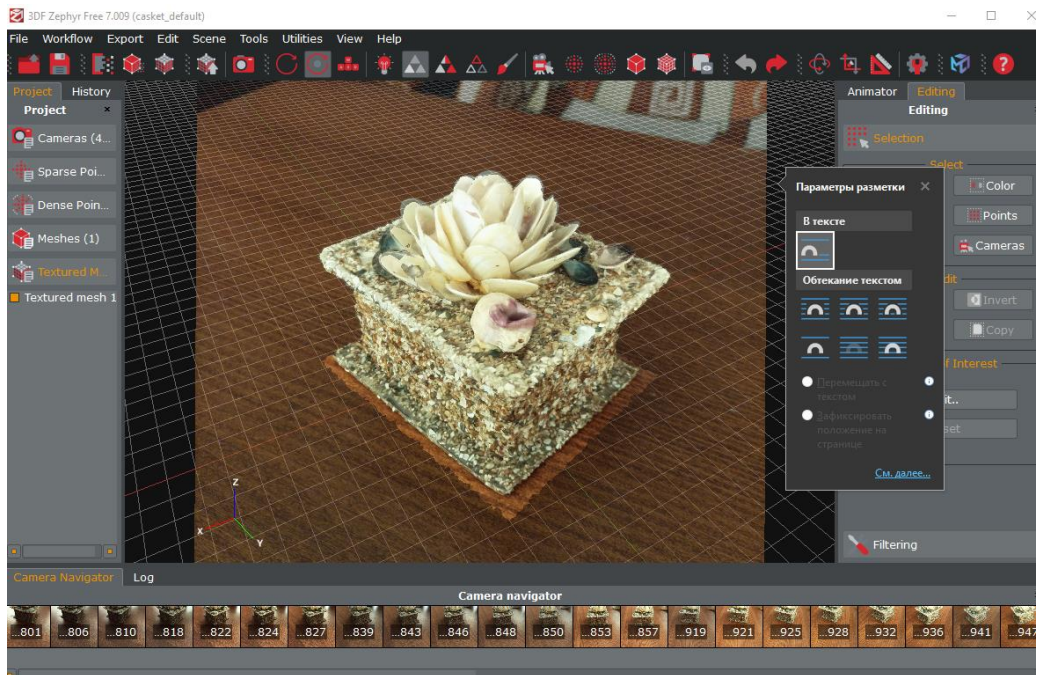


Рис. 3.17. Фотографія та відповідне їй положення фотокамери

Для 3D-моделі триангуляційна сітка відображається командою Rendering/Enable Wireframe з контекстного меню (рис. 3.18).

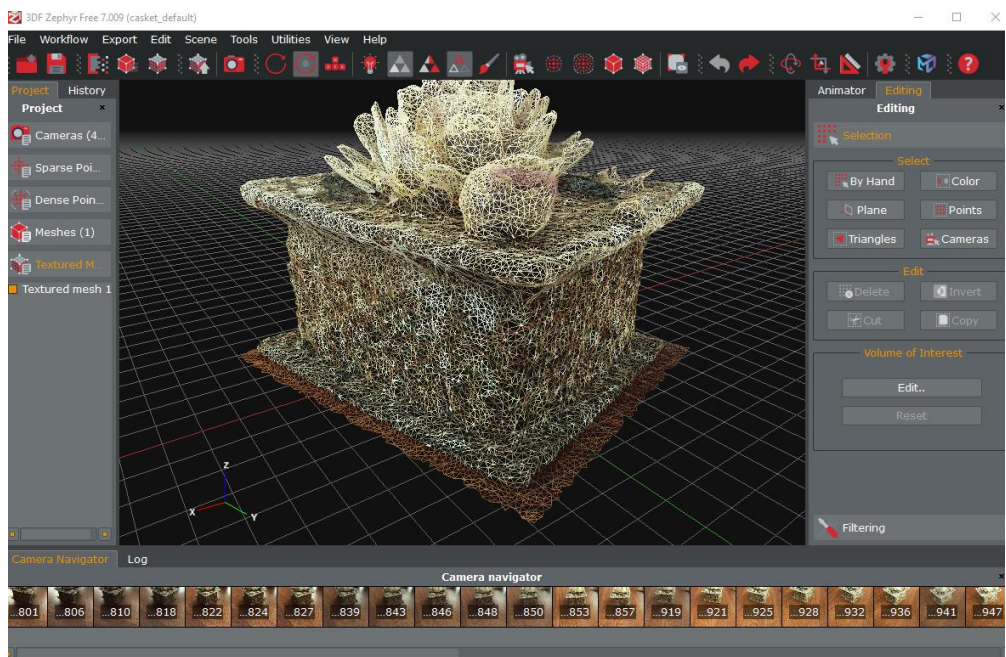


Рис. 3.18. 3D-модель з відображенням триангуляційної сітки (Wireframe)

Шляхом повороту вибирається потрібний ракурс моделі. 3D модель зберігається у файлі формату .zpr. Сітка зберігається командою Export / Export Mesh (формати .ply, Obj, Glb, Upload to Sketchlab).

Далі повторено етапи побудови моделі із високими налаштуваннями (рис. 3.19).

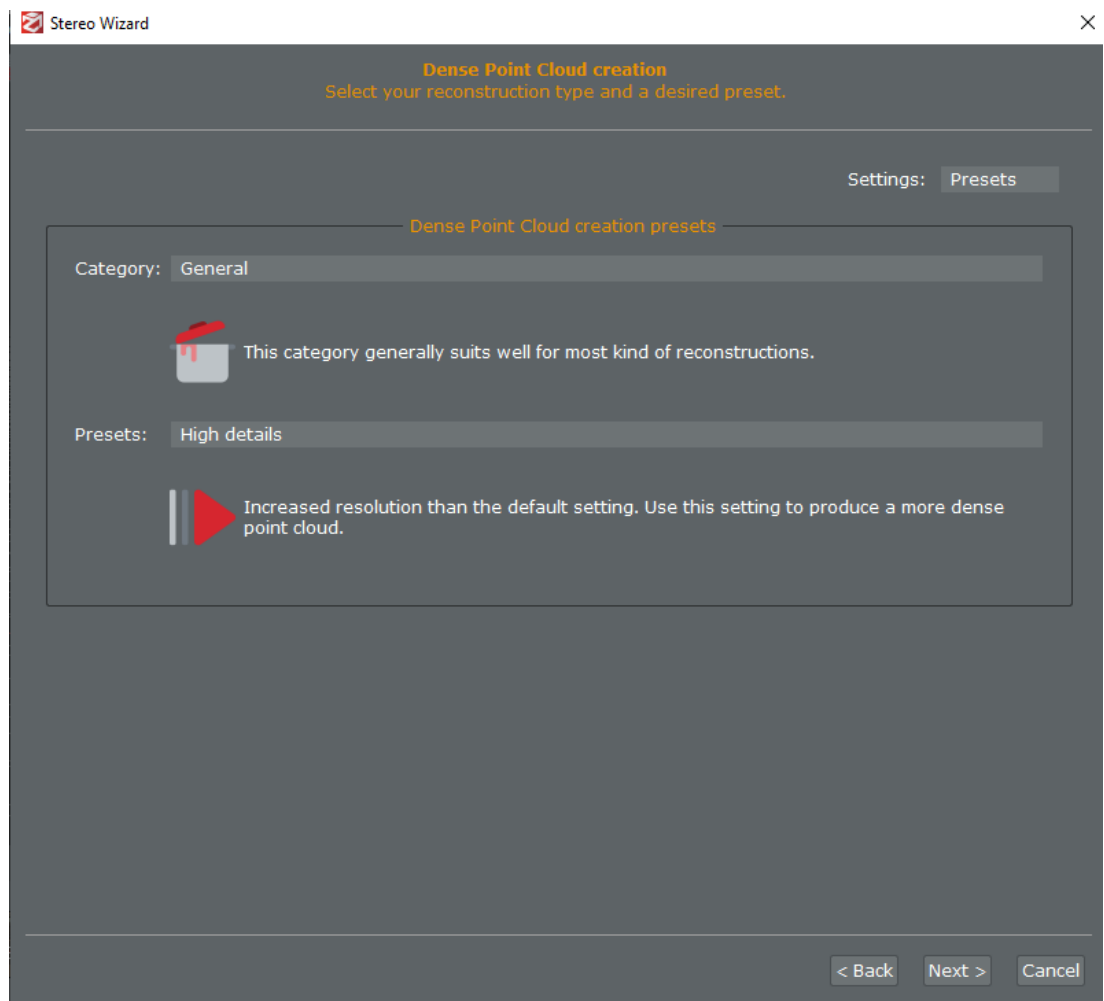


Рис. 3.19. Встановлення високих параметрів

Ідеальна поверхня – зерниста, без бліків, відбивань, добре освітлена. Додати сюди високі параметри і можна отримати найкращий варіант із можливих (рис. 3.20 – рис. 3.23).

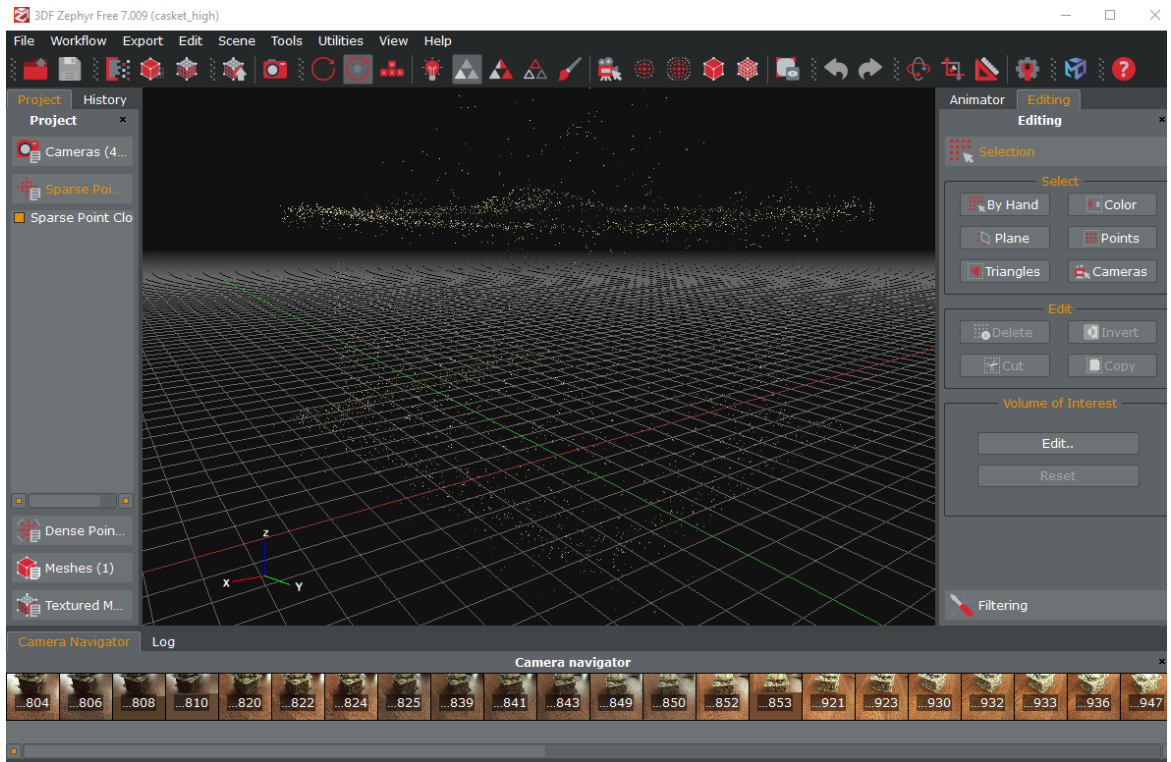


Рис. 3.20. Розріджена хмара точок

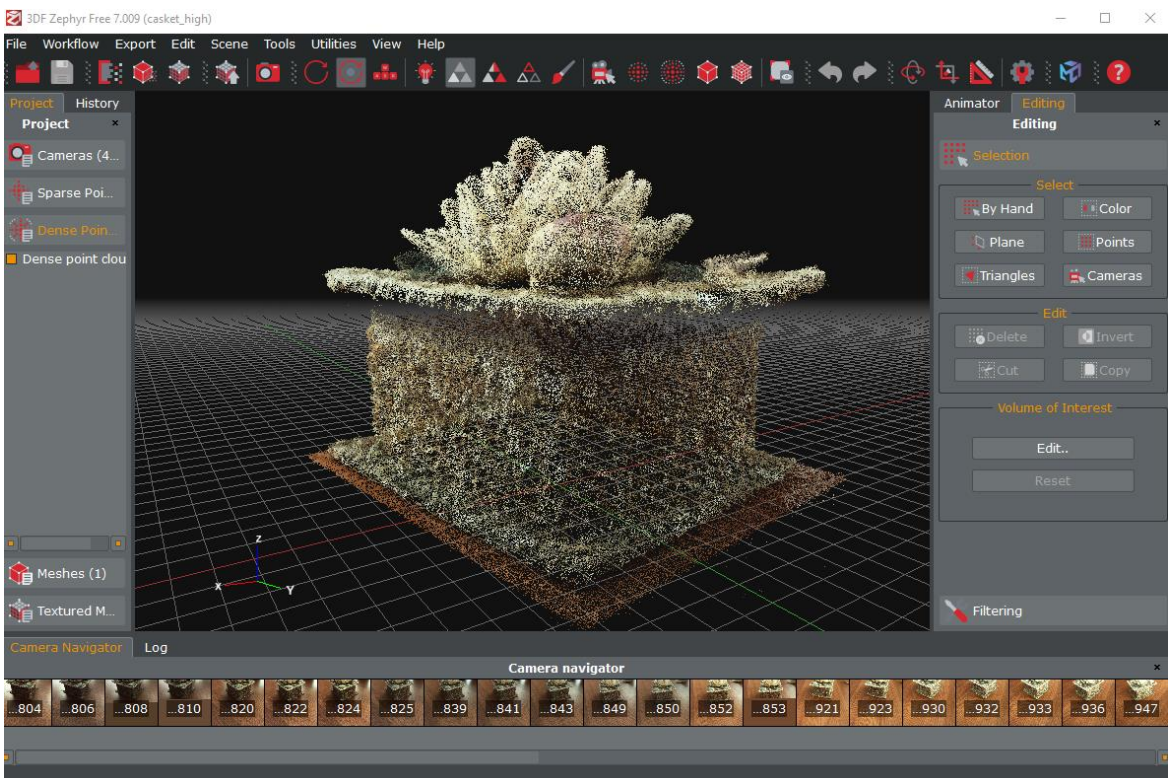


Рис. 3.21. Щільна хмара точок

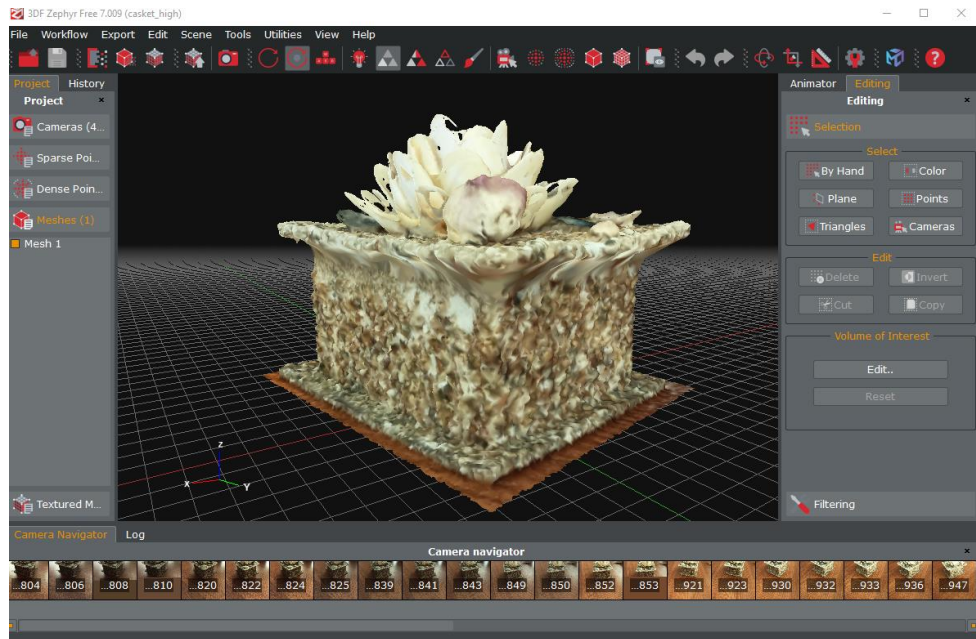


Рис. 3.22. Сітка полігонів

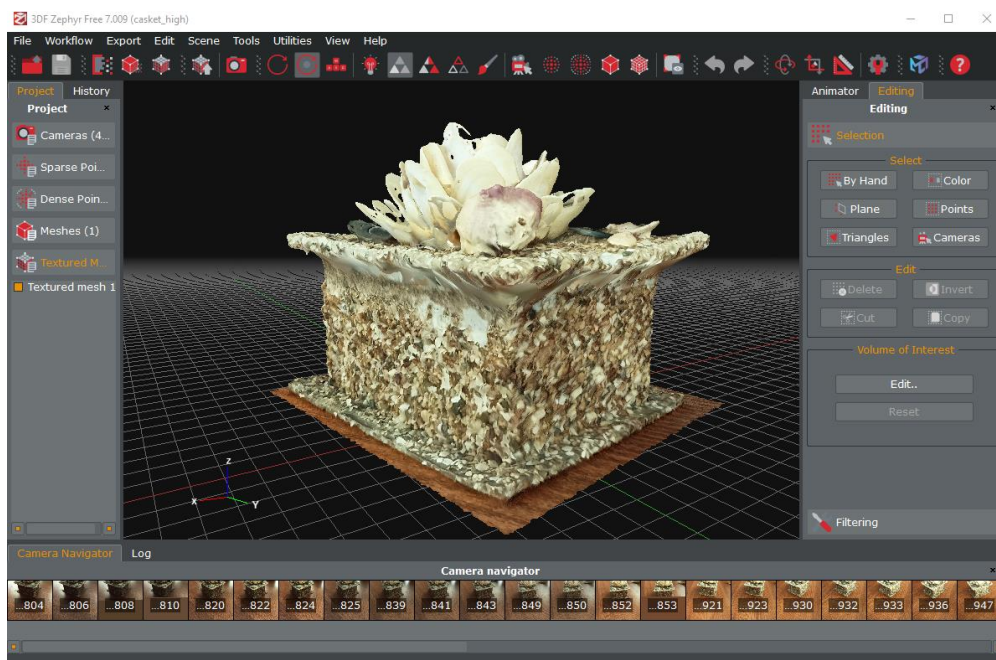


Рис. 3.23. Текстура сітка

Також є різниця у властивостях моделі і у вигляді триангуляційної сітки (рис. 3.24, рис. 3.25).

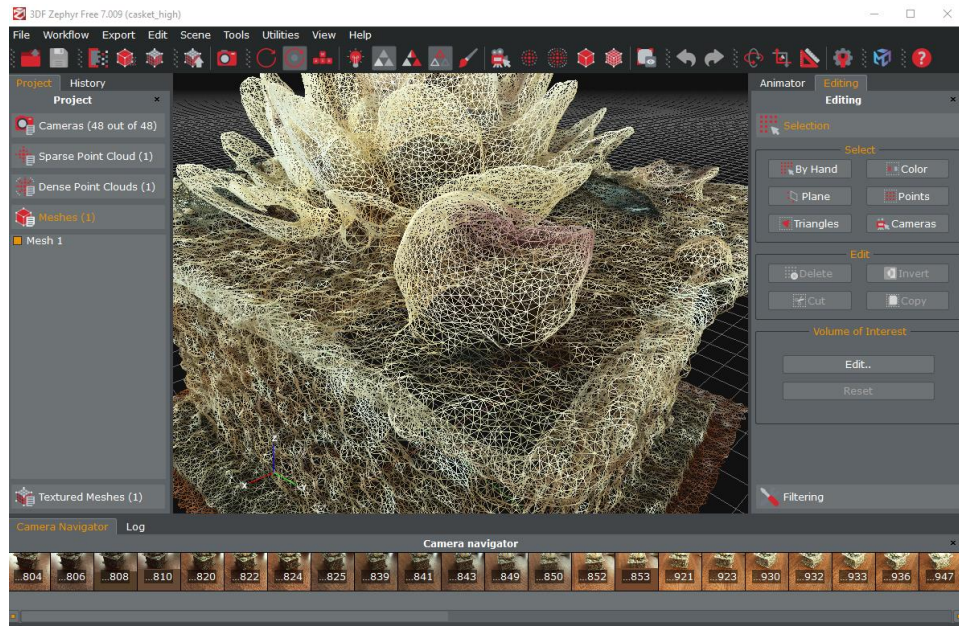


Рис. 3.24. Триангуляційна сітка на високих налаштуваннях

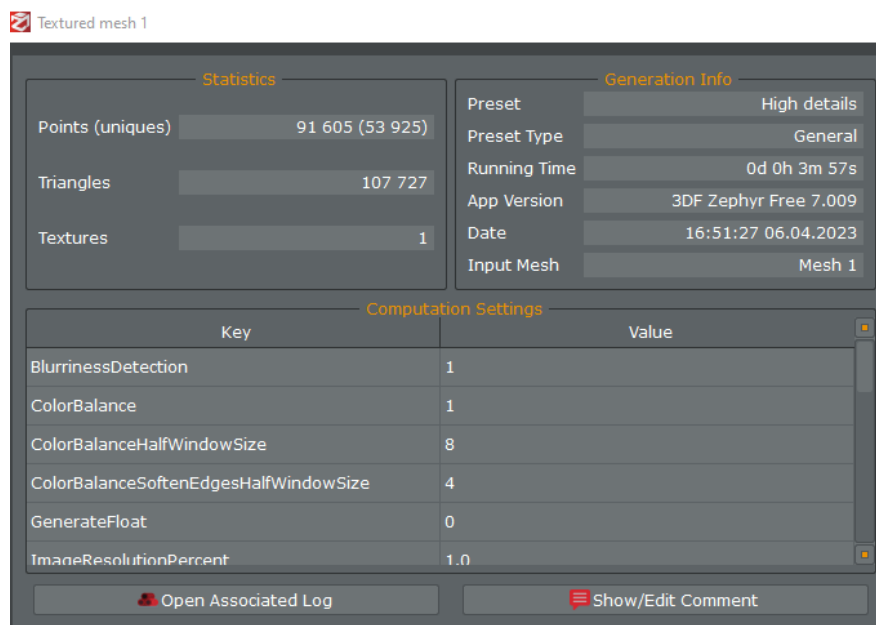


Рис. 3.25. Властивості моделі

Отже, було побудовано чотири тривимірних моделі.

Моделі ноутбука із увімкненим монітором були з артефактами, які не вдалось виправити. Це невдалі моделі, в обробці яких зіграли роль багато негативних факторів – відбиваюча поверхня, габарити, освітлення. Проте основним позитивним фактором була яскрава та кольорова поверхня екрану.

У наступному підрозділі буде розглянутий протилежний приклад та описано, наскільки важливу роль це грає.

Моделі шкатулки вдалися найкраще. Зерниста текстура, невеликий розмір, чудове освітлення – хоч воно аналогічно до ноутбуку, через невеликі габарити об'єкту та відсутність значних затіньовань дозволити отримати добрий результат (хоча найкраще використовувати непряме розсіяне світло). У СПР будуть описані і позитивні фактори, які впливають на побудову моделі.

Далі слідує дослідження найбільш невдалого досвіду: ноутбука із виключеним екраном. Для такого об'єкту описано всі фактори, які вплинули на спотворення моделі.

### **3.2 Представлення моделі з дефектами**

Розроблено рекомендації щодо того, як уникнути деяких артефактів під час створення 3D моделі.

Окрім ноутбука із включеним екраном, була спроба працювати із виключеним. Фон був світлий (використовувалась звичайне біле полотно), а фотографії отримувалися у приміщенні при денному та штучному освітленні (рис. 3.26 – рис. 3.29).



Рис. 3.26. Приклад фотографії

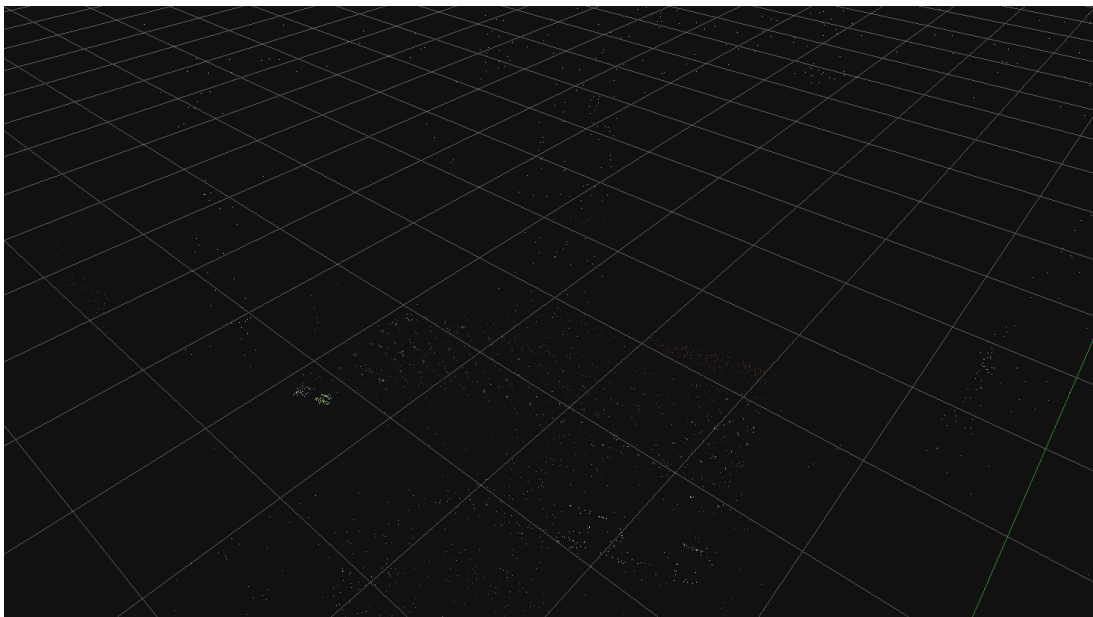


Рис. 3.27. Розріджена хмара точок

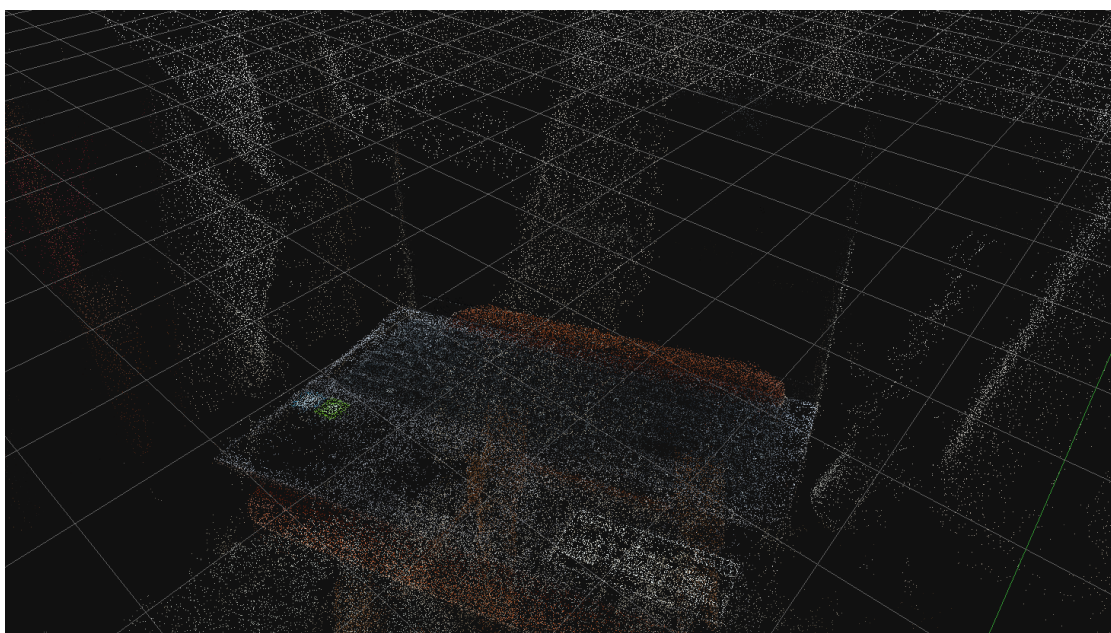


Рис. 3.28. Щільна хмара точок





Рис. 3.29. Сітка полігонів

Вже на етапі побудови сітки полігонів добре видно дефекти. Програма коректно зчитує кольорові, зернисті, добре освітлені об'єкти. В даному прикладі поверхня ноутбука гладка, утворює білі і чорна. Програма зчитала її як тінь та наклала її на білий фон з полотна (рис. 3.30).



Рис. 3.30. Інший ракурс моделі

На рис. 3.31 видно, як програма «побачила» об'єкт. Окрім рамки і кількох плям з бліків, вона не зчитала поверхню екрану, замість цього «зафарбувала» нею полотно на фоні.

Така проблема вирішується лише зміною об'єкту та покращенням якості зйомки (наприклад ще більше фотографій з різних ракурсів або зміни освітлення). Зміна налаштувань програми, підвищення якості обробки тут не спрацює, об'єкт все одно буде викривлений (рис. 3.32).

Було використано розсіяне штучне освітлення, білий фон. Ноутбук має чорну та гладку поверхню екрану, тому зміна параметрів майстра не вкливає на обробку. Ці фактори враховано у СПР, як факти-умови та факти-наслідки.



Рис. 3.31. Модель з дефектами

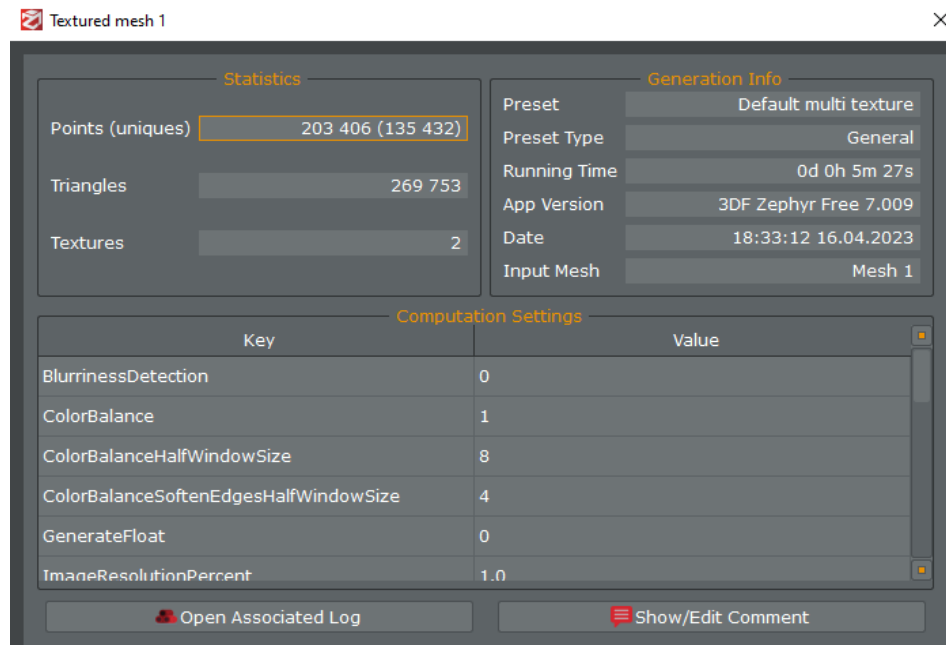


Рис. 3.32. Текстурована модель та її властивості

Отже, було отримано модель з дефектами та виділено деякі негативні фактори при її побудові. Такі фактори, як темна поверхня, гладка поверхня, розсіяне штучне освітлення і т.д. У наступному розділі приводиться повний список фактів, отриманий на основі досвіду побудови всіх моделей.

### 3.3 Створення системи правил для експертної системи

#### 3.3.1 Виділення фактів-умов

Дослідивши три об'єкти та п'ять 3D моделей (по дві на перші два і одна на останній), отриманих за допомогою фотограмметрії, були розроблені рекомендації по побудові моделей. Такі рекомендації оформлено у вигляді системи правил продукційної експертної системи. Зібрані факти згруповано та логічно об'єднано.

Перший тип фактів – це факти-умови. Це те, що є об'єктивним стосовно моделі:

- тип освітлення;
- тип поверхні;
- задані налаштування;

- кількість фотографій;
- фокус камери;
- перекривання кадрів;
- роздільна здатність камери;
- джерела світла у кадрі;
- інші об'єкти в кадрі.

Далі наводиться список всіх фактів-умов:

1. (положення камер рівномірно розподілене у просторі)
2. (положення камер не рівномірно розподілене у просторі)
3. (кількість фотографій мала, менша 25)
4. (кількість фотографій середня, більша 25)
5. (роздільна здатність фотографій низька, менше 3000 пікселів за шириною або висотою)
6. (роздільна здатність фотографій висока, більше 3000 пікселів за шириною або висотою)
7. (перекривання кадрів низьке, менше ніж на 60%)
8. (перекривання кадрів високе, більше ніж на 60%)
9. (фотографії об'єкта отримано у всіх ракурсах)
10. (фотографії об'єкта отримано тільки у деяких ракурсах)
11. (досліджуваний об'єкт у центрі кадру)
12. (досліджуваний об'єкт не в центрі кадру)
13. (досліджуваний об'єкт у фокусі)
14. (досліджуваний об'єкт не у фокусі, зображення розмите)
15. (досліджуваний об'єкт займає меншу площу кадру)
16. (досліджуваний об'єкт займає більшу площу кадру)
17. (рухомі об'єкти у кадрі)
18. (присутні нетекстуровані поверхні)
19. (присутні прозорі та напівпрозорі поверхні)
20. (присутні дзеркально відбиваючі поверхні)
21. (присутні блики /відблиски/ у кадрі)
22. (пряме штучне освітлення)
23. (джерела світла у кадрі)
24. (прямі сонячні промені)
25. (розсіяне освітлення)
26. (параметри майстра про побудові тривимірної моделі низькі)
27. (параметри майстра про побудові тривимірної моделі за замовчуванням)
28. (параметри майстра про побудові тривимірної моделі високі)
29. (при побудові розрідженої хмари точок значна частина фотографій відкидається)
30. (присутні зайві фрагменти тривимірної моделі)

Тридцять фактів-умов є достатніми для більшості ситуацій. У подальшому планується доповнювати їх набір у програмі.

### 3.3.2 Виділення фактів-наслідків

Факти-наслідки впливають із попередніх. Вони в основному характеризують якість створюваних моделей:

- «розмитість» текстури;
- наявність артефактів;
- деформованість моделі;
- точність відтворення;
- якість освітлення;
- якість триангуляційної сітки.

Список фактів-наслідків:

1. (якість 3D моделі низька)
2. (якість 3D моделі середня)
3. (якість 3D моделі висока)
4. (кількість точок 3D моделі низька)
5. (кількість точок 3D моделі середня)
6. (кількість точок 3D моделі висока)
7. (освітлення моделі низьке)
8. (освітлення моделі середнє)
9. (освітлення моделі високе)
10. (низька якість темних ділянок моделі)
11. (висока якість темних ділянок моделі)
12. (низька якість світлих ділянок моделі)
13. (висока якість світлих ділянок моделі)
14. (присутні спотворення триангуляційної сітки)
15. (присутні спотворення триангуляційної сітки у деяких фрагментах моделі)
16. (присутні спотворення текстури)
17. (спотворення текстури локальні)
18. (низька швидкість побудови моделі)
19. (наявність артефактів у моделі)

### 3.3.3 Формування фактів-рекомендацій

Найголовнішим є розробити рекомендації для побудови моделей. Для того, щоб виправити певне спотворення чи артефакт, необхідно виконати відповідну дію. Тут можна зіслатись на висновки зі створення 3D моделей у попередніх розділах роботи – саме завдяки цим дослідженням можна виділити наступні «початкові» рекомендації для поліпшення якості роботи:

1. (розмістити камери рівномірно у просторі)
2. (збільшити кількість фотографій)
3. (збільшити роздільну здатність фотографій)
4. (отримати фотографії об'єкту у всіх ракурсах)
5. (розмістити досліджуваний об'єкт у центрі кадру)
6. (розмістити об'єкт у фокусі камери)
7. (видалити рухомі об'єкти з кадру)
8. (зменшити площу нетекстурованих поверхонь)
9. (зменшити площу прозорих або напівпрозорих поверхонь)
10. (зменшити площу дзеркально відбиваючих поверхонь)
11. (використати розсіяне освітлення)
12. (встановити низькі значення параметрів майстра про побудові тривимірної моделі)
13. (встановити високі значення параметрів майстра про побудові тривимірної моделі)
14. (видалити зайві фрагменти моделі в 3DF Zephyr)
15. (збільшити освітлення моделі)
16. (зменшити освітлення моделі)

### 3.3.4 Логічне об'єднання у правила

На основі отриманих трьох типів фактів створено систему правил.

Логіка правил буде зв'язувати факти у наступний ланцюжок:

*факт-умова=>факт-наслідок=>факт-рекомендація*

Користувачу буде необхідно ввести лише факт-умову (чи декілька таких), які програма зв'яже із фактами-наслідками, і, маючи набір з умов і наслідків, вона обиратиме і виводитиме необхідну рекомендацію для вирішення проблеми. У програмі також будуть враховуватись і нормальні умови. Наприклад на факт «текстура нормальна» програма видаватиме «додаткових дій не потребується». Таким чином, можна охопити найбільш широкий набір ситуацій і враховувати можливі проблеми.

Наведемо приклад простих правил:

1. Якщо (якість 3D моделі низька) (кількість фотографій мала, менша 25) То (збільшити кількість фотографій)
2. Якщо (якість 3D моделі низька) (досліджуваний об'єкт не в центрі кадру) То (розмістити досліджуваний об'єкт у центрі кадру)
3. Якщо (якість 3D моделі низька) (досліджуваний об'єкт не у фокусі, зображення розмите) То (розмістити об'єкт у фокусі камери)

Перш, ніж створювати правила, необхідно об'єднати факти. Забігаючи наперед, необхідно відмітити, що було створено декілька версій програм, з невеликою різницею у логіці. Вони будуть описані пізніше, але у деяких з них дещо інша логіка правил.

Основна логіка «умова-наслідки-рекомендації»:

1. (положення камер рівномірно розподілене у просторі)
  - Якість 3D моделі: висока
  - Кількість точок 3D моделі: висока
  - Освітлення моделі: високе
  - Швидкість побудови моделі: висока
  - Рекомендація: Немає потреби в додаткових діях.
2. (положення камер не рівномірно розподілене у просторі)
  - Якість 3D моделі: низька
  - Кількість точок 3D моделі: низька
  - Освітлення моделі: низьке
  - Швидкість побудови моделі: низька
  - Рекомендація: Перерозташуйте камери для рівномірного покриття об'єкта.
3. (кількість фотографій мала, менша 25)
  - Якість 3D моделі: низька
  - Кількість точок 3D моделі: низька
  - Освітлення моделі: низьке
  - Швидкість побудови моделі: низька
  - Рекомендація: Збільште кількість фотографій для отримання більш детальної моделі.
4. (кількість фотографій середня, більша 25)
  - Якість 3D моделі: середня
  - Кількість точок 3D моделі: середня
  - Освітлення моделі: середнє
  - Швидкість побудови моделі: середня
  - Рекомендація: Додайте додаткові фотографії для покращення якості моделі.

Це лише приклад чотирьох правил. Всього їх в усіх версіях програми буде по 30-31 – по одному на кожен факт-умову. Повний список правил наведено у додатках.

Всього правил у різних версіях програми буде більше, в залежності від надлаштувань.

### 3.3.5 Опис CLIPSIDE 6.3

Середовище CLIPS (C Language Integrated Production System – мова С, інтегрована з продукційними системами) використовується для розробки баз знань і експертних систем (ЕС) [1-3].

Дистрибутив CLIPS «clips\_631\_windows\_64\_bit\_installer» потрібно зчитати з сайту розробника:

<https://sourceforge.net/projects/clipsrules/files/CLIPS/6.31/>

Після цього потрібно виконати інсталяцію середовища CLIPS і відкрити його командою «CLIPS 6.31/CLIPSIDE».

Мова CLIPS 6.3 використовує продукційну модель представлення знань і містить три основні елементи [2]:

1. Список фактів.
2. Базу знань (БЗ).
3. Блок логічного виведення.

Складовими частинами БЗ в мові CLIPS являються: факти (fact), правила та функції. Евристичний механізм представлення знань в CLIPS реалізується за допомогою правил, які визначають набір дій, виконуваних при виникненні певної ситуації.

Правила складаються з умови та дії:

**ЯКЩО**

<умова\_1> і ... і <умова\_N> виконуються,

**ТО**

виконати <дію\_1> і ... і <дію\_N>.

Умова (антицедент) записується після ключового слова ЯКЩО, тобто це ліва частина правила або LHS правила (left-hand side of rule). Умова записується як множина фактів, присутність яких означає виконання умови.



Дія (консиквент) правила записується після ключового слова TO, тобто це права частина правила або RHS правила (right-hand side of rule).

Якщо умова в лівій частині правила виконується, то відбувається активізація правила і виконуються всі дії, записані в його правій частині. В іншому випадку правило не активізується.

Для оголошення і додавання нових правил в БЗ використовується конструктор defrule. Синтаксична схема конструктора наступна:

```
(defrule <імя_правила>
[<"необов'язковий_коментар">]
[<необов'язкове_визначення_властивостей_правила>]
<умова_1>                ; ліва частина правила
...
<умова_m>
=>                        ; спеціальний символ
<дія_1>                   ; права частина правила
...
<дія_n>).
```

Для поділу правої і лівої частини правил використовується спеціальний символ =>.

Всі умови в лівій частині правила об'єднуються за допомогою неявного логічного оператора AND. Якщо в лівій частині правила не вказано ні один умовний елемент, CLIPS автоматично підставляє умову-зразок initial-fact. Таким чином, правило активізується кожен раз при появі в БЗ факту initial-fact.

Розглянемо приклад правила «color\_diamond\_colorless»:

```
(defrule color_diamond_colorless
(declare (salience 100))
```

```
(color_colorless)
=>
(assert (color_diamond))
(printout t "color_diamond" crlf)
)
```

Властивість `salience` (у діапазоні від -10000 до 10000) дозволяє користувачеві призначати певний пріоритет для своїх правил. Чим більше число, що визначає пріоритет, тим вище пріоритет у правила.

Правило активізується, якщо в ЕС буде присутнім факт `(color_colorless)`. При активізації правила командою `assert` додається новий факт `(color_diamond)` і виводиться повідомлення «`color_diamond`».

Існує вісім типів умовних елементів, використовуваних в лівій частині правил. Розглянемо детальніше три умовні елементи:

- Умова `and` застосовується для визначення групи умов, кожна з якої повинна бути задовільною.
- Умова `or` – для визначення однієї умови з деякою групи, яка повинна бути задовільною.
- Умова `not` – для визначення умови, яка не повинна бути задовільною.

Факти можуть бути додані в список фактів (використовуючи функцію `assert`), видалені з нього (використовуючи функцію `retract`), змінені (використовуючи функцію `modify`) або скопійовані (використовуючи функцію `duplicate`).

Крім явно заданих фактів, CLIPS також автоматично додає наперед визначений користувачем або програмою факт (`initial-fact`), кожного разу при виконанні команди (`reset`).

Для спостереження за процесом додавання, видалення або зміни фактів необхідно викликати пункт меню Execution опцію Watch і встановити прапорець в пункті меню «Facts».

При запуску програми, яка містить множину фактів і правил, інтерпретатор CLIPS запускає машину логічного виведення, яка з'ясовує, які з правил можна активізувати. Це виконується циклічно, причому кожен цикл складається з трьох кроків:

- зіставлення фактів і правил;
- вибір правила, що підлягає активізації;
- виконання дій, передбачених правилом.

Мова CLIPS 6.3 підтримує сім різних стратегій вирішення конфліктів: стратегія глибини (depth strategy), стратегія ширини (breadth strategy), стратегія спрощення (simplicity strategy), стратегія ускладнення (complexity strategy), LEX (LEX strategy), MEA (MEA strategy) і випадкова стратегія (random strategy). За замовчуванням в CLIPS встановлена стратегія глибини, згідно з якою щойно активоване правило поміщається вище всіх правил з таким же пріоритетом.

Необхідна стратегія може бути обрана в меню Execution у пункті Options за допомогою списку Strategy.

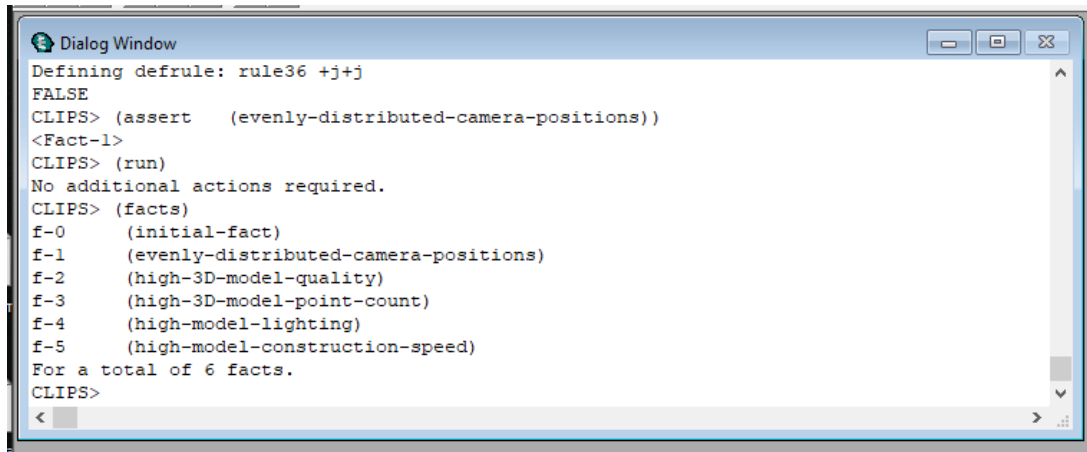
Назви фактів і правил у CLIPS потрібно вводити на англійській мові.

### **3.3.6 Створення програми**

Маючи логічно сформовані факти та правила, розроблено програму для надання рекомендацій. Лістинги приводяться у додатках.

В процесі роботи виникла необхідність відобразити декілька різних версій через те, що вони мають дещо різну стабільність. Нижче спочатку приводиться найбільш стабільна версія, яка, однак не дуже коректно слідує за визначеною логікою.

Вивід програми при запуску (run) після вводу одного факту (рис. 3.33).



```

Dialog Window
Defining defrule: rule36 +j+j
FALSE
CLIPS> (assert (evenly-distributed-camera-positions))
<Fact-1>
CLIPS> (run)
No additional actions required.
CLIPS> (facts)
f-0 (initial-fact)
f-1 (evenly-distributed-camera-positions)
f-2 (high-3D-model-quality)
f-3 (high-3D-model-point-count)
f-4 (high-model-lighting)
f-5 (high-model-construction-speed)
For a total of 6 facts.
CLIPS>

```

Рис. 3.33. Термінал CLIPS

На рисунку використовується початкова версія програми, яка діє за логікою, визначеною раніше. Після вводу файлу з правилами, виконується ввід факту через команду (assert (...)), після чого програма запускається. Оскільки факт відповідає правилу №1, програма видає «No additional actions required». Далі командою (facts) виводиться список фактів, які програма ввела, відреагувавши на факт користувача.

Отже, програма частково слідує за визначеною раніше логікою: ввести один факт-умову, отримати наслідки, вивести рекомендацію. При цьому самі наслідки ні де не використовуються, а просто запам'ятовуються, а рекомендації прив'язані скоріше до фактів-умов.

Далі йдуть інші програми з опціональними налаштуваннями (рис. 3.34).

```

Dialog Window
=>
  (retract (initial-fact
FALSE
CLIPS> (assert (quality-low))
<Fact-1>
CLIPS> (assert (point-count-low)
)
<Fact-2>
CLIPS> (assert (model-lighting-low))
<Fact-3>
CLIPS> (assert (model-construction-speed-low))
<Fact-4>
CLIPS> (run)
Initializing the expert system...
CLIPS> (facts)
f-0      (initial-fact)
f-1      (quality-low)
f-2      (point-count-low)
f-3      (model-lighting-low)
f-4      (model-construction-speed-low)
f-5      (Decision (action "Reposition the cameras for even coverage of the object.))
f-6      (Decision (action "Capture photos of the object from all views to improve model quality.))
f-7      (Decision (action "Ensure the object of interest is always in focus to improve the model quality.))
f-8      (initial-facts)
For a total of 9 facts.
CLIPS> (run)
Recommendation: Ensure the object of interest is always in focus to improve the model quality.
Recommendation: Capture photos of the object from all views to improve model quality.
Recommendation: Reposition the cameras for even coverage of the object.
Recommendations:
- Reposition the cameras for even coverage of the object.
- Capture photos of the object from all views to improve model quality.
- Ensure the object of interest is always in focus to improve the model quality.
Total recommendations: 4
CLIPS>

```

Рис. 3.34. Друга версія програми

Це остання, найбільш «технологічна» версія. Має більш ускладнену логіку: рекомендації тепер відстежуються як рішення («Decision») і групуються у список. Також частково присутній моніторинг системи. Тепер всі правила обов'язково базуються на декількох фактах-умовах, а факти наслідки запам'ятовуються програмою і вже на них видаються рекомендації.

Ця програма найбільш точно відповідає логіці правил.

Далі наводиться третя, спрощена версія (рис. 3.35).

```

Dialog Window
Defining defrule: rule2 +j+j
Defining defrule: rule3 +j+j
Defining defrule: rule4 +j+j
Defining defrule: decision-rule +j+j
Defining defrule: print-recommendations +j+j

[CSTRCPSR1] Expected the beginning of a construct.
FALSE
CLIPS> (assert (lack-of-external-model-texturing))
<Fact-1>
CLIPS> (run)
Recommendations:
- Add external texturing to the model.
- Improve the lighting of the model.
- Improve the quality of the 3D model.
Total recommendations: 4
CLIPS>

```

Рис. 3.35. Третя версія програми

По складності вона знаходиться між першою та другою. Тут правила знову відштовхуються від лише одного факту-умови, рекомендації перетворюються на Decision і також групуються. Відсутній монітор СПР, логіка задовільняється повністю.

Така версія більш стабільна, ніж друга та може бути використана для подальшої модифікації.

### **Висновок до розділу 3**

Програмою 3DF Zephyr було створено п'ять тривимірних моделей різних об'єктів та при різних параметрах побудови. Зроблено висновок про негативний вплив на якість моделі гладких і напівпрозорих поверхонь, сильного затінення. На основі отриманого досвіду розроблено набір фактів і правил для експертної системи у середовищі CLIPS. Завдяки експертній системі можливо отримувати рекомендації для покращення якості побудови тривимірних моделей.

## ВИСНОВКИ

1. За допомогою програми 3DF Zephyr побудовано тривимірні моделі об'єктів методом фотограмметрії. Аналіз побудованих тривимірних моделей показав, що в них присутні певні артефакти.
2. Програмою 3DF Zephyr було створено п'ять тривимірних моделей різних об'єктів та при різних параметрах побудови. Зроблено висновок про негативний вплив на якість моделі гладких і напівпрозорих поверхонь, сильного затінення.
3. Розроблено інформаційну систему для допомоги при створенні тривимірних моделей. На основі отриманого досвіду розроблено набір фактів і правил для СПР у середовищі CLIPS. Розроблено програму, яка автоматично видає рекомендації при вводі відповідних проблем (фактів) та відстежує рішення користувача (в деяких версіях програми).
4. Завдяки СПР можливо отримувати рекомендації для покращення якості побудови тривимірних моделей

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Купріянич І.П. Фотограмметрія та дистанційне зондування: навч. посіб. для студ. вищ. навч. закл. // І.П. Купріянич, Є.В. Бутенко. – К.: МВЦ «Медінформ», 2013. – 392 с.
2. 3DF Zephyr. The Complete Photogrammetry Solution. URL: <https://www.3dflow.net>.
3. Рибальченко М.О. Цифрова обробка сигналів. Навчальний посібник / М.О. Рибальченко, О.П. Єгоров, В.Б. Зворикін. – Дніпро: НМетАУ, 2018. – 79 с. – [https://nmetau.edu.ua/file/navchalniy\\_posibnik\\_tsos.pdf](https://nmetau.edu.ua/file/navchalniy_posibnik_tsos.pdf)
4. Вовк С.М. Методи обробки зображень та комп'ютерний зір : навч. посіб. / С.М. Вовк, В.В. Гнатушенко, М.В. Бондаренко; Дніпропетр. нац. ун-т ім. Олеся Гончара. – Дніпропетровськ : Ліра, 2016. –147 с.
5. Kerem Kargin. Computer Vision Fundamentals and OpenCV Overview. URL: <https://medium.com/mllearning-ai/computer-vision-fundamentals-and-opencv-overview-9a30fe94f0ce>
6. Gonzalez R. Digital image processing / R. Gonzalez, R. Woods. – 4th edidion, Pearson/ Prentice Hall, NY, 2018. – 1192 p.
7. Конспект лекцій з навчальної дисципліни «Комп'ютерні системи штучного інтелекту» / уклад.: С.В. Баловсяк, Х.С. Одайська. Чернівці: ЧНУ, 2022. – 175 с.
8. Комп'ютерні системи штучного інтелекту: методичні вказівки до лабораторних робіт / уклад.: С.В. Баловсяк, Х.С. Одайська. Чернівці: ЧНУ, 2022. – 96 с.
9. Савченко А.С. Методи та системи штучного інтелекту: Навч. посібник / А.С. Савченко, О. О. Синельніков. – К. : НАУ, 2017. – 190 с.
10. Шаховська Н. Б. Системи штучного інтелекту: навчальний посібник / Н.Б. Шаховська, Р. М. Камінський, О. Б. Вовк. – Львів : Видавництво Львівської політехніки, 2018. – 392 с.



- 11.A Tool for Building Expert Systems. [Электронный ресурс] – Режим доступа: [http:// clipsrules.sourceforge.net/](http://clipsrules.sourceforge.net/)
- 12.Artificial Intelligence. A Modern Approach : Fourth Edition / Ed. S. Russell, P.Norvig. – Pearson Education, 2021. – 1069 p. – <http://aima.cs.berkeley.edu>.

## ДОДАТКИ

### Додаток А. Лістинги програми

#### Лістинг 1. Перша версія

```
(defrule rule1
  (declare (saliency 100))
  (evenly-distributed-camera-positions)
  =>
  (assert (high-3D-model-quality))
  (assert (high-3D-model-point-count))
  (assert (high-model-lighting))
  (assert (high-model-construction-speed))
  (printout t "No additional actions required." crlf)
)

(defrule rule2
  (declare (saliency 100))
  (unevenly-distributed-camera-positions)
  =>
  (assert (low-3D-model-quality))
  (assert (low-3D-model-point-count))
  (assert (low-model-lighting))
  (assert (low-model-construction-speed))
  (printout t "Reposition the cameras for even coverage of the object." crlf)
)

(defrule rule3
  (declare (saliency 100))
  (low-number-of-photos)
  =>
  (assert (low-3D-model-quality))
  (assert (low-3D-model-point-count))
  (assert (low-model-lighting))
  (assert (low-model-construction-speed))
  (printout t "Increase the number of photos for a more detailed model." crlf)
)

(defrule rule4
  (declare (saliency 100))
  (medium-number-of-photos)
  =>
  (assert (medium-3D-model-quality))
  (assert (medium-3D-model-point-count))
  (assert (medium-model-lighting))
  (assert (medium-model-construction-speed))
  (printout t "Add additional photos to improve the quality of the model." crlf)
)

(defrule rule5
  (declare (saliency 100))
  (low-photo-resolution)
  =>
  (assert (low-3D-model-quality))
  (assert (low-3D-model-point-count))
  (assert (low-model-lighting))
  (assert (low-model-construction-speed))
  (printout t "Use higher resolution photos for a more detailed model." crlf)
)

(defrule rule6
  (declare (saliency 100))
  (high-photo-resolution)
  =>
```

```

(assert (high-3D-model-quality))
(assert (high-3D-model-point-count))
(assert (high-model-lighting))
(assert (high-model-construction-speed))
(printout t "Use photos with high resolution for a more detailed model." crlf)
)

(defrule rule7
(declare (saliency 100))
(low-frame-overlap)
=>
(assert (low-3D-model-quality))
(assert (low-3D-model-point-count))
(assert (low-model-lighting))
(assert (low-model-construction-speed))
(printout t "Increase the frame overlap for a more complete model." crlf)
)

(defrule rule8
(declare (saliency 100))
(high-frame-overlap)
=>
(assert (high-3D-model-quality))
(assert (high-3D-model-point-count))
(assert (high-model-lighting))
(assert (high-model-construction-speed))
(printout t "Reduce the frame overlap to improve the quality of the model." crlf)
)

(defrule rule9
(declare (saliency 100))
(photos-from-all-angles)
=>
(assert (high-3D-model-quality))
(assert (high-3D-model-point-count))
(assert (high-model-lighting))
(assert (high-model-construction-speed))
(printout t "No additional actions required." crlf)
)

(defrule rule10
(declare (saliency 100))
(photos-missing-from-some-angles)
=>
(assert (low-3D-model-quality))
(assert (low-3D-model-point-count))
(assert (low-model-lighting))
(assert (low-model-construction-speed))
(printout t "Capture photos of the object from all angles to improve the model quality." crlf)
)

(defrule rule11
(declare (saliency 100))
(object-in-center-of-frame)
=>
(assert (high-3D-model-quality))
(assert (high-3D-model-point-count))
(assert (high-model-lighting))
(printout t "Place the object in the center of the frame for a more accurate and balanced model." crlf)
)
(досліджуваний об'єкт не в центрі кадру)
clips
Copy code
(defrule rule12
(declare (saliency 100))
(object-not-in-center-of-frame)

```

```

=>
(assert (low-3D-model-quality))
(assert (low-3D-model-point-count))
(assert (low-model-lighting))
(printout t "Position the object in the center of the frame to improve the model quality." crlf)
)

(defrule rule13
(declare (saliency 100))
(object-in-focus)
=>
(assert (high-3D-model-quality))
(assert (high-3D-model-point-count))
(assert (high-model-lighting))
(printout t "Ensure that the object is always in focus for a more detailed model." crlf)
)

(defrule rule14
(declare (saliency 100))
(object-out-of-focus)
=>
(assert (low-3D-model-quality))
(assert (low-3D-model-point-count))
(assert (low-model-lighting))
(printout t "Ensure that the object is always in focus to improve the model quality." crlf)
)

(defrule rule15
(declare (saliency 100))
(object-takes-less-space)
=>
(assert (low-3D-model-quality))
(assert (low-3D-model-point-count))
(assert (low-model-lighting))
(printout t "Fill the frame with the object to get a more detailed model." crlf)
)

(defrule rule16
(declare (saliency 100))
(object-takes-more-space)
=>
(assert (high-3D-model-quality))
(assert (high-3D-model-point-count))
(assert (high-model-lighting))
(printout t "Position the object to occupy more space in the frame for a more detailed model." crlf)
)

(defrule rule17
(declare (saliency 100))
(moving-objects-in-frame)
=>
(assert (low-3D-model-quality))
(assert (low-3D-model-point-count))
(assert (low-model-lighting))
(printout t "Remove moving objects from the frame or use special techniques to correct for moving objects to improve the model quality." crlf)
)

(defrule rule18
(declare (saliency 100))
(unt textured-surfaces-present)
=>
(assert (low-3D-model-quality))
(assert (low-3D-model-point-count))
(assert (low-model-lighting))
(printout t "Reduce the area of untextured surfaces or apply texturing methods to enhance the model quality." crlf)
)

```

```

)

(defrule rule19
  (declare (salience 100))
  (transparent-surfaces-present)
  =>
  (assert (low-3D-model-quality))
  (assert (low-3D-model-point-count))
  (assert (low-model-lighting))
  (printout t "Reduce the area of transparent or translucent surfaces to achieve a more accurate model." crlf)
)

(defrule rule20
  (declare (salience 100))
  (mirror-like-surfaces-present)
  =>
  (assert (low-3D-model-quality))
  (assert (low-3D-model-point-count))
  (assert (low-model-lighting))
  (printout t "Reduce the area of mirror-like surfaces or apply specialized techniques for modeling such surfaces with high quality." crlf)
)

(defrule rule21
  (declare (salience 100))
  (image-noise-present)
  =>
  (assert (low-3D-model-quality))
  (assert (low-3D-model-point-count))
  (assert (low-model-lighting))
  (printout t "Apply noise reduction techniques to improve the model quality when there is noise present in the image." crlf)
)

(defrule rule22
  (declare (salience 100))
  (glare-or-reflections-present)
  =>
  (assert (low-3D-model-quality))
  (assert (low-3D-model-point-count))
  (assert (low-model-lighting))
  (printout t "Apply glare or reflection compensation methods to improve the model quality and reduce their impact." crlf)
)

(defrule rule23
  (declare (salience 100))
  (direct-artificial-lighting-present)
  =>
  (assert (high-3D-model-quality))
  (assert (high-3D-model-point-count))
  (assert (high-model-lighting))
  (printout t "Use direct artificial lighting to achieve a more accurate and detailed model." crlf)
)

(defrule rule24
  (declare (salience 100))
  (insufficient-light-sources)
  =>
  (assert (medium-3D-model-quality))
  (assert (medium-3D-model-point-count))
  (assert (medium-model-lighting))
  (printout t "Ensure an adequate number of light sources in the frame to improve the model quality." crlf)
)

(defrule rule25
  (declare (salience 100))
  (direct-sunlight-present)

```

```

=>
(assert (low-3D-model-quality))
(assert (low-3D-model-point-count))
(assert (low-model-lighting))
(printout t "Reduce the impact of direct sunlight in the frame or use compensation methods to improve the model quality."
 crlf)
)

(defrule rule26
(declare (salience 100))
(diffuse-lighting-present)
=>
(assert (high-3D-model-quality))
(assert (high-3D-model-point-count))
(assert (high-model-lighting))
(printout t "Use diffuse lighting to achieve a more even and soft illumination of the model." crlf)
)

(defrule rule27
(declare (salience 100))
(low-master-parameters)
=>
(assert (low-3D-model-quality))
(assert (low-3D-model-point-count))
(assert (low-model-lighting))
(printout t "Increase the values of the master parameters for 3D model construction to obtain a more detailed and accurate
model." crlf)
)

(defrule rule28
(declare (salience 100))
(default-master-parameters)
=>
(assert (medium-3D-model-quality))
(assert (medium-3D-model-point-count))
(assert (medium-model-lighting))
(printout t "Set the master parameters for 3D model construction to optimal values to achieve a balance between model
quality and construction speed." crlf)
)

(defrule rule29
(declare (salience 100))
(high-master-parameters)
=>
(assert (high-3D-model-quality))
(assert (high-3D-model-point-count))
(assert (high-model-lighting))
(printout t "Apply high values for the master parameters for 3D model construction to achieve the highest quality and level of
detail in the model." crlf)
)

(defrule rule30
(declare (salience 100))
(texture-issues-present)
=>
(assert (low-3D-model-quality))
(assert (low-3D-model-point-count))
(assert (low-model-lighting))
(printout t "Check the presence and quality of textures to improve the visual appearance of the model. Apply texture
corrections or add details if necessary." crlf)
)

(defrule rule31
(declare (salience 100))
(incorrect-model-scaling)
=>

```

```

(assert (medium-3D-model-quality))
(assert (medium-3D-model-point-count))
(assert (medium-model-lighting))
(printout t "Check the scaling of the model and ensure it corresponds to the real-world proportions. Adjust or change the
scaling if necessary to achieve correct proportions and visual appearance." crlf)
)

(defrule rule32
(declare (saliency 100))
(incompatible-file-formats)
=>
(assert (medium-3D-model-quality))
(assert (medium-3D-model-point-count))
(assert (medium-model-lighting))
(printout t "Ensure that all file formats used for the model and textures are compatible with each other. Convert file formats
or ensure their compatibility for proper model rendering." crlf)
)

(defrule rule33
(declare (saliency 100))
(insufficient-object-detail)
=>
(assert (low-3D-model-quality))
(assert (low-3D-model-point-count))
(assert (low-model-lighting))
(printout t "Add more points and details to individual objects in the model to improve their appearance and realism." crlf)
)

(defrule rule34
(declare (saliency 100))
(shadows-missing)
=>
(assert (medium-3D-model-quality))
(assert (medium-3D-model-point-count))
(assert (medium-model-lighting))
(printout t "Add realistic shadows to the model to enhance its visual appearance and simulate lighting effects." crlf)
)

(defrule rule35
(declare (saliency 100))
(object-overlapping)
=>
(assert (low-3D-model-quality))
(assert (low-3D-model-point-count))
(assert (low-model-lighting))
(printout t "Check for object overlapping in the model, which can distort their appearance. Adjust the position or size of
objects if necessary to achieve a more accurate representation." crlf)
)

(defrule rule36
(declare (saliency 100))
(incorrect-model-size)
=>
(assert (medium-3D-model-quality))
(assert (medium-3D-model-point-count))
(assert (medium-model-lighting))
(printout t "Verify the size of the model and its suitability for the project requirements. Change the model size if necessary to
achieve the desired proportions and scale." crlf)
)

```

## Лістинг 2. Друга версія

```

(deftemplate Decision
(slot action)
)

```

```

(deffacts initial-facts
)

(defrule rule1
  (declare (saliency 100))
  (not (inconsistent-facts))
  (quality-high)
  (point-count-high)
  (model-lighting-high)
  (model-construction-speed-high)
  =>
  (assert (Decision (action "No additional actions required.")))
)

(defrule rule2
  (declare (saliency 100))
  (not (inconsistent-facts))
  (quality-low)
  (point-count-low)
  (model-lighting-low)
  (model-construction-speed-low)
  =>
  (assert (Decision (action "Reposition the cameras for even coverage of the object.")))
)

(defrule rule3
  (declare (saliency 100))
  (not (inconsistent-facts))
  (point-count-low)
  (model-lighting-low)
  (model-construction-speed-low)
  (test (< (count-facts ((?f Photo)) TRUE) 25))
  =>
  (assert (Decision (action "Increase the number of photos to obtain a more detailed model.")))
)

(defrule rule4
  (declare (saliency 100))
  (not (inconsistent-facts))
  (point-count-low)
  (model-lighting-low)
  (model-construction-speed-low)
  (test (>= (count-facts ((?f Photo)) TRUE) 25))
  =>
  (assert (Decision (action "Add additional photos to improve the quality of the model.")))
)

(defrule rule5
  (declare (saliency 100))
  (not (inconsistent-facts))
  (point-count-low)
  (model-lighting-low)
  (model-construction-speed-low)
  (test (or (< (image-resolution-low) 3000) (< (image-resolution-low) 3000)))
  =>
  (assert (Decision (action "Use higher-resolution photos for a more detailed model.")))
)

(defrule rule6
  (declare (saliency 100))
  (not (inconsistent-facts))
  (point-count-low)
  (model-lighting-low)
  (model-construction-speed-low)
  (test (and (>= (image-resolution-low) 3000) (>= (image-resolution-low) 3000)))

```



```

=>
(assert (Decision (action "Use high-resolution photos for a more detailed model.")))
)

(defrule rule7
(declare (saliency 100))
(not (inconsistent-facts))
(point-count-low)
(model-lighting-low)
(model-construction-speed-low)
(test (< (frame-overlap) 60))
=>
(assert (Decision (action "Increase frame overlap for a more complete model.")))
)

(defrule rule8
(declare (saliency 100))
(not (inconsistent-facts))
(point-count-low)
(model-lighting-low)
(model-construction-speed-low)
(test (>= (frame-overlap) 60))
=>
(assert (Decision (action "Reduce frame overlap to improve model quality.")))
)

(defrule rule9
(declare (saliency 100))
(not (inconsistent-facts))
(point-count-high)
(model-lighting-high)
(model-construction-speed-high)
(all-views-captured)
=>
(assert (Decision (action "No additional actions required.")))
)

(defrule rule10
(declare (saliency 100))
(not (inconsistent-facts))
(point-count-low)
(model-lighting-low)
(model-construction-speed-low)
(not (all-views-captured))
=>
(assert (Decision (action "Capture photos of the object from all views to improve model quality.")))
)

(defrule rule11
(declare (saliency 100))
(not (inconsistent-facts))
(quality-high)
(point-count-high)
(model-lighting-high)
(test (object-in-center))
=>
(assert (Decision (action "Place the object of interest in the center of the frame for a more accurate and balanced model.")))
)

(defrule rule12
(declare (saliency 100))
(not (inconsistent-facts))
(quality-low)
(point-count-low)
(model-lighting-low)
(test (not (object-in-center)))

```

```

=>
(assert (Decision (action "Place the object of interest in the center of the frame to improve the model quality.")))
)

(defrule rule13
(declare (saliency 100))
(not (inconsistent-facts))
(quality-high)
(point-count-high)
(model-lighting-high)
(object-in-focus)
=>
(assert (Decision (action "Ensure the object of interest is always in focus for a more detailed model.")))
)

(defrule rule14
(declare (saliency 100))
(not (inconsistent-facts))
(quality-low)
(point-count-low)
(model-lighting-low)
(not (object-in-focus))
=>
(assert (Decision (action "Ensure the object of interest is always in focus to improve the model quality.")))
)

(defrule rule15
(declare (saliency 100))
(not (inconsistent-facts))
(quality-low)
(point-count-low)
(model-lighting-low)
(object-occupies-less-area)
=>
(assert (Decision (action "Maximize the object coverage in the frame for a more detailed model.")))
)

(defrule rule16
(declare (saliency 100))
(not (inconsistent-facts))
(quality-high)
(point-count-high)
(model-lighting-high)
(object-occupies-more-area)
=>
(assert (Decision (action "Position the object of interest to occupy more area in the frame for a more detailed model.")))
)

(defrule rule17
(declare (saliency 100))
(not (inconsistent-facts))
(quality-low)
(point-count-low)
(model-lighting-low)
(moving-objects-present)
=>
(assert (Decision (action "Remove moving objects from the frame or use special techniques to correct for moving objects for better model quality.")))
)

(defrule rule18
(declare (saliency 100))
(not (inconsistent-facts))
(quality-low)
(point-count-low)
(model-lighting-low)

```

```
(non-textured-surfaces-present)
=>
(assert (Decision (action "Reduce the area of non-textured surfaces or apply texturing methods to improve model quality.")))
)
```

```
(defrule rule19
(declare (saliency 100))
(not (inconsistent-facts))
(quality-low)
(point-count-low)
(model-lighting-low)
(transparent-surfaces-present)
=>
(assert (Decision (action "Reduce the area of transparent or semi-transparent surfaces for a more accurate model.")))
)
```

```
(defrule rule20
(declare (saliency 100))
(not (inconsistent-facts))
(quality-low)
(point-count-low)
(model-lighting-low)
(reflective-surfaces-present)
=>
(assert (Decision (action "Reduce the area of highly reflective surfaces or use special methods to model such surfaces with high quality.")))
)
```

```
(defrule rule21
(declare (saliency 100))
(not (inconsistent-facts))
(quality-low)
(point-count-low)
(model-lighting-low)
(image-noise-present)
=>
(assert (Decision (action "Apply denoising techniques to improve model quality in the presence of image noise.")))
)
```

```
(defrule rule22
(declare (saliency 100))
(not (inconsistent-facts))
(quality-low)
(point-count-low)
(model-lighting-low)
(glare-or-reflection-present)
=>
(assert (Decision (action "Apply glare or reflection compensation methods to improve model quality and reduce their impact.")))
)
```

```
(defrule rule23
(declare (saliency 100))
(not (inconsistent-facts))
(quality-high)
(point-count-high)
(model-lighting-high)
(direct-artificial-lighting-present)
=>
(assert (Decision (action "Use direct artificial lighting to achieve a more accurate and detailed model.")))
)
```

```
(defrule rule24
(declare (saliency 100))
(not (inconsistent-facts))
(quality-medium)
```

```

(point-count-medium)
(model-lighting-medium)
(insufficient-light-sources)
=>
(assert (Decision (action "Ensure an adequate number of light sources in the frame to improve model quality.")))
)

(defrule rule25
(declare (salience 100))
(not (inconsistent-facts))
(quality-low)
(point-count-low)
(model-lighting-low)
(direct-sunlight-present)
=>
(assert (Decision (action "Reduce the impact of direct sunlight in the frame or use compensation methods to improve model quality.")))
)

(defrule rule26
(declare (salience 100))
(not (inconsistent-facts))
(quality-high)
(point-count-high)
(model-lighting-high)
(diffuse-lighting-present)
=>
(assert (Decision (action "Use diffuse lighting to achieve a more uniform and soft illumination of the model.")))
)

(defrule rule27
(declare (salience 100))
(not (inconsistent-facts))
(quality-low)
(point-count-low)
(model-lighting-low)
(low-master-builder-parameters)
=>
(assert (Decision (action "Increase the values of the master builder parameters to obtain a more detailed and accurate model.")))
)

(defrule rule28
(declare (salience 100))
(not (inconsistent-facts))
(quality-medium)
(point-count-medium)
(model-lighting-medium)
(default-master-builder-parameters)
=>
(assert (Decision (action "Set optimal values for the master builder parameters to achieve a balance between model quality and construction speed.")))
)

(defrule rule29
(declare (salience 100))
(not (inconsistent-facts))
(quality-high)
(point-count-high)
(model-lighting-high)
(high-master-builder-parameters)
=>
(assert (Decision (action "Apply high values for the master builder parameters to achieve the highest model quality and level of detail.")))
)

```

```
(defrule rule30
  (declare (salience 100))
  (not (inconsistent-facts))
  (quality-low)
  (point-count-low)
  (model-lighting-low)
  (texture-issues-present)
  =>
  (assert (Decision (action "Check for the presence and quality of textures to improve the visual appearance of the model. Apply texture corrections or add details if necessary.")))
)
```

```
(defrule rule31
  (declare (salience 100))
  (not (inconsistent-facts))
  (quality-medium)
  (point-count-medium)
  (model-lighting-medium)
  (incorrect-model-scaling)
  =>
  (assert (Decision (action "Check the scaling of the model and ensure it corresponds to real-world proportions. Restore or modify the scale if necessary to achieve correct proportions and appearance.")))
)
```

```
(defrule rule32
  (declare (salience 100))
  (not (inconsistent-facts))
  (quality-medium)
  (point-count-medium)
  (model-lighting-medium)
  (file-format-incompatibility)
  =>
  (assert (Decision (action "Ensure all file formats used for the model and textures are compatible with each other. Convert file formats or ensure their compatibility for correct model rendering.")))
)
```

```
(defrule rule33
  (declare (salience 100))
  (not (inconsistent-facts))
  (quality-low)
  (point-count-low)
  (model-lighting-low)
  (insufficient-detailing-of-objects)
  =>
  (assert (Decision (action "Add more points and details to individual objects in the model to enhance their appearance and realism.")))
)
```

```
(defrule rule34
  (declare (salience 100))
  (not (inconsistent-facts))
  (quality-medium)
  (point-count-medium)
  (model-lighting-medium)
  (lack-of-shadows)
  =>
  (assert (Decision (action "Add realistic shadows to the model to improve its visual appearance and reproduce the effect of lighting.")))
)
```

```
(defrule rule35
  (declare (salience 100))
  (not (inconsistent-facts))
  (quality-low)
  (point-count-low)
  (model-lighting-low)
```

```

(object-overlap)
=>
(assert (Decision (action "Check for object overlap in the model, which can distort their appearance. Adjust the position or size
of objects if necessary to achieve a more accurate representation.")))
)

(defrule rule36
(declare (salience 100))
(not (inconsistent-facts))
(quality-medium)
(point-count-medium)
(model-lighting-medium)
(incorrect-model-size)
=>
(assert (Decision (action "Check the size of the model and ensure it meets the project requirements. Modify the model's size if
necessary to achieve the desired proportions and scale.")))
)

(defrule decision-rule
(declare (salience -10))
(Decision (action ?decision))
=>
(printout t "Recommendation: " ?decision crlf)
)

(defrule print-recommendations
(declare (salience -30))
(exists (Decision))
=>
(printout t "Recommendations:" crlf)
(bind ?count 1)
(do-for-all-facts ((?d Decision)) TRUE
  (printout t "- " ?d:action crlf)
  (bind ?count (+ ?count 1))
)
(printout t "Total recommendations: " ?count crlf)
)

(defrule inconsistency-check
(declare (salience -100))
(exists (inconsistent-facts))
=>
(printout t "Inconsistent facts detected. Please check the input data." crlf)
)

(defrule reset-inconsistent-facts
(declare (salience -200))
?f <- (inconsistent-facts)
=>
(retract ?f)
(assert (initial-fact))
)

(defrule check-consistency
(not (inconsistent-facts))
(not (initial-fact))
=>
(assert (inconsistent-facts))
)

(defrule initialize
(initial-fact)
(not (initial-facts))
=>
(printout t "Initializing the expert system..." crlf)
(assert (initial-facts))
)

```

```

(halt)
)

(defrule stop
(initial-facts)
(not (initial-fact))
=>
(printout t "Expert system initialized." crlf)
(assert (initial-fact))
(halt)
)

(defrule run-expert-system
(initial-fact)
(initial-facts)
(not (control-fact))
=>
(retract (initial-fact))
(printout t "Running the expert system..." crlf)
(assert (control-fact))
)

(assert (initial-fact))
(assert (initial-facts))
(run-expert-system)

```

### Лістинг 3. Третя версія програми

```

(deftemplate Decision
(slot action)
)

(deffacts initial-facts
)

(defrule rule2
(declare (salience 100))
(lack-of-external-model-texturing)
=>
(assert (low-3D-model-quality))
(assert (low-3D-model-point-count))
(assert (low-model-lighting))
(assert (medium-model-construction-speed))
(assert (need-external-texturing))
(assert (Decision (action "Add external texturing to the model.")))
)

(defrule rule3
(declare (salience 100))
(low-3D-model-quality)
=>
(assert (low-3D-model-point-count))
(assert (low-model-lighting))
(assert (medium-model-construction-speed))
(assert (need-improve-model-quality))
(assert (Decision (action "Improve the quality of the 3D model.")))
)

(defrule rule4
(declare (salience 100))
(low-model-lighting)
=>
(assert (low-3D-model-quality))
(assert (low-3D-model-point-count))
(assert (medium-model-construction-speed))
(assert (need-improve-lighting))
)

```

```
(assert (Decision (action "Improve the lighting of the model.")))
)

(defrule decision-rule
  (declare (salience -10))
  (Decision (action ?decision))
  =>
  ;; (printout t "Recommendation: " ?decision crlf)
  )

(defrule print-recommendations
  (declare (salience -30))
  (exists (Decision))
  =>
  (printout t "Recommendations:" crlf)
  (bind ?count 1)
  (do-for-all-facts ((?d Decision)) TRUE
    (printout t "- " ?d:action crlf)
    (bind ?count (+ ?count 1))
  )
  (printout t "Total recommendations: " ?count crlf)
  )

(reset)
(run)
```