Міністерство освіти і науки України  
НТУУ «КПІ ім. І. Сікорського»

Кафедра цифрових технологій в енергетиці

Візуалізація графічної та геометричної інформації

Розрахунково-графічна робота

«Операції на текстурних координатах»

Виконав:  
студент 5-го курсу НН ІАТЕ  
групи ТР-41мп  
Сухенко Б. О.

Перевірив:

доц. Демчишин А. А.

Київ – 2024

Завдання:

Необхідно реалізувати масштабування/обертання текстури (координат текстури) навколо визначеної користувачем точки – непарні варіанти реалізують масштабування, парні варіанти реалізують обертання. По варіанту 19 – необхідно масштабування.

Повинна бути можливість переміщати точку вздовж простору поверхні (u, v) за допомогою клавіатури, наприклад клавіші A і D переміщують точку вздовж параметра u, а клавіші W і S переміщують точку вздовж параметра v.

Теоретична інформація

Текстурні координати – це двовимірні значення, які використовуються для прив’язування текстури (зображення) до геометричної поверхні в тривимірному просторі. Вони вказують, як кожна точка текстури накладається на вершини об’єкта, забезпечуючи його місцезнаходження. Координати зазвичай позначаються як (u, v), де:

* u – координата по горизонтальній осі текстури.
* v – координата по вертикальній осі текстури.

Значення u і v зазвичай нормалізовані до діапазону [0, 1], де:

* u = 0 відповідає лівому краю текстури, а u = 1 – правому.
* v = 0 відповідає нижньому краю текстури, а v = 1 – верхньому.

Типи відображення текстур:

1. UV-Mapping – найпоширеніший метод, коли текстура проектується на поверхню об’єкта через спеціальну розкладку координат u-v.
2. Процедурне відображення – текстура генерується математичними формулами, а не реальним зображенням.
3. Сферичне, циліндричне та кубічне відображення – підходи для проектування текстур на специфічні геометричні форми.

Операції на текстурних координатах:

1. Масштабування текстури: масштабування виконується шляхом множення текстурних координат на масштабний коефіцієнт. Наприклад, якщо текстурні координати u і v збільшити вдвічі, текстура буде здаватися вдвічі меншою на об'єкті, оскільки більше її частин поміститься на поверхні.
2. Зсув (офсет): зсув координат u або v зміщує текстуру відносно поверхні. Наприклад, додавання фіксованого значення до u призводить до зміщення текстури вправо або вліво.
3. Плиткове відображення (tiling): це метод, коли текстура повторюється багаторазово для покриття всієї поверхні. Це досягається використанням координат за межами [0, 1]. При такому підході важливо правильно працювати з межами, щоб уникнути накладання текстур.
4. Дзеркальне відображення (mirroring): координати віддзеркалюються навколо центральної точки для отримання ефекту дзеркального повторення. Це часто використовується для запобігання різким переходам між плитками текстури.
5. Кліппінг (clipping): це обмеження текстурних координат у певних межах, щоб уникнути відображення текстури за межами об'єкта або запобігти небажаним артефактам.

Проблеми при роботі з текстурними координатами:

1. Шви текстури (texture seams): шви з'являються на стику текстури через неправильне вирівнювання координат або повторення текстури, коли використовуються значення за межами діапазону [0, 1]. Це часто трапляється при неправильному масштабуванні або зсуві.
2. Розтягування та стискання: коли текстура застосовується до неправильної геометрії або з неправильно розрахованими координатами, вона може розтягуватися або стискатися, спотворюючи зображення.
3. Аліазинг: це візуальний ефект, при якому текстура виглядає "смугастою" або "блоковою" через недостатнє згладжування. Для його усунення застосовуються методи фільтрації, такі як міпмапінг.

Фільтрація дозволяє поліпшити якість відображення текстури на об’єкті:

* Nearest-neighbor – використовує найближчий піксель, швидкий, але призводить до "рваних" країв.
* Bilinear і Trilinear фільтрація – використовують інтерполяцію між пікселями для більш плавного відображення.
* Anisotropic filtering – покращує якість текстур під кутом.

Практичне застосування текстурних координат:

* Градієнтне фарбування: текстурні координати можуть використовуватися для створення градієнтів або переходів кольору на об'єктах.
* Альфа-маппінг: текстури застосовуються для прозорості, коли деякі ділянки об'єкта мають різний ступінь прозорості.
* Бамп-маппінг: використання текстур для імітації дрібних деталей рельєфу на поверхні.

Текстурні координати є важливою частиною процесу рендерингу, забезпечуючи правильне і гармонійне з'єднання візуального вигляду текстури з тривимірною геометрією об'єкта.

Деталі реалізації

За основу взяти контрольне завдання (лабораторна робота №3), де було реалізовано текстурування з 3 текстурами: дифузна, відбивання(дзеркальна) і нормалей.

Першим ділом в index.html файл в розмітку було додано слайдер масштабу в діапазоні від 0.1 до 5 з кроком 0.1 і стандартним значенням 1 (тобто без масштабування).

В main.js додано змінну textureScale, яка оновлюється при кожній зміні слайдера. При її зміні відбувається також оновлення поверхні і заново буферизуються дані (рис. 1).

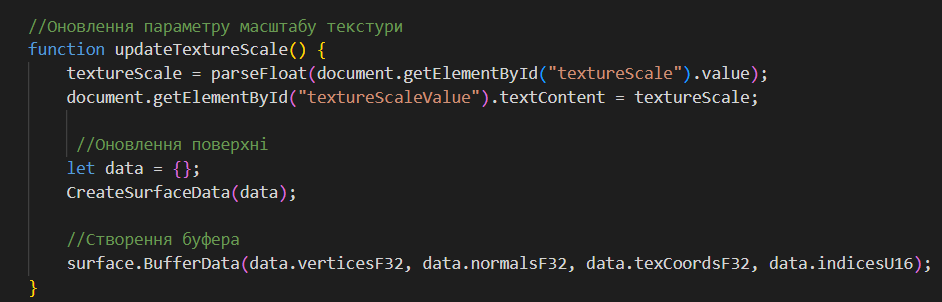


Рис. 1 – Оновлення масштабу текстури

Потім додано змінні uOffset та vOffset, які відповідатимуть за координати точки, відносно якої виконується масштабування, і івентЛістенер для натискань, який при натисненні кнопок W та S (з увімкненою англійською розкладкою) змінює координату u точки, в при натисненні кнопок D та A – відповідно координату v. Зміна відбувається з кроком 0.01 за натиснення. При зміні також відображається значення u-v поточного місцезнаходження цієї точки біля слайдера. Також додано перевірки щоб не виходити з діапазону [0; 1]. Також, при натиснені будь-якої з цих кнопок після зміни uOffset та vOffset відбувається перерахунок і перебуферизація поверхні. На рисунку 2 показано основну частину відповідного коду.

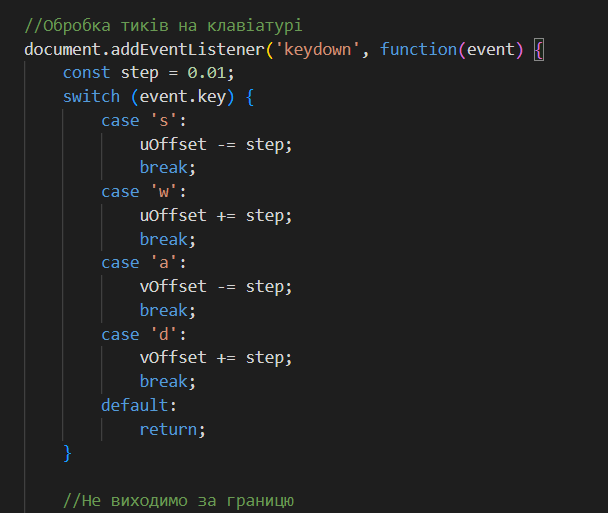


Рис. 2 – Обробка натиснень на клавіатурі

Саме ж масштабування текстури відносно точки полягає у зміні текстурних координат таким чином, щоб текстура виглядала збільшеною або зменшеною навколо заданої точки. Як і написано раніше, ця точка визначається зміщенням uOffset та vOffset. Це зміщення вказує, де саме на текстурі розташована точка, яка залишається на місці під час масштабування, забезпечуючи ефект масштабування навколо цієї точки. u = 0 відповідає лівому краю текстури, u = 1 – правому краю, а v = 0 відповідає нижньому краю текстури, в той час як v = 1 – верхньому.

Спочатку текстурні координати кожної вершини поверхні перетворюються в нормалізований діапазон від 0 до 1. Це дозволяє працювати з координатами незалежно від розмірів самої поверхні. Потім виконуємо зсув цих координат таким чином, щоб точка з фіксованими координатами залишалася незмінною. Наступним кроком є застосування коефіцієнта масштабування, який змінює відстань між кожною точкою текстури і точкою, навколо якої виконується масштабування. Якщо коефіцієнт більше 1, текстура збільшується, а якщо менше 1 – зменшується, що показано на рисунку 3.

Важливим аспектом є те, що після масштабування текстурні координати можуть виходити за межі допустимого діапазону [0, 1]. Для вирішення цієї проблеми застосовується остача від ділення на 1 – будь-які значення координат, які виходять за межі діапазону, циклічно повертаються в цей діапазон. Це дозволяє уникнути викривлень або непередбачуваних ефектів на межах текстури:

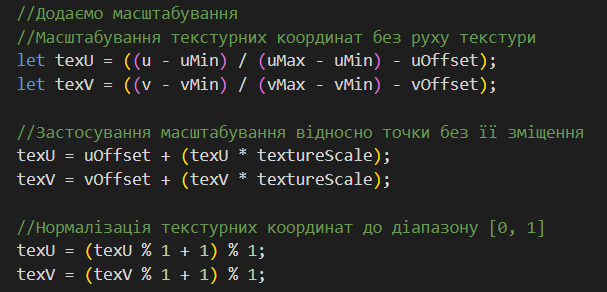


Рисунок 3 – Масштабування відносно точки

Отже, загальний процес виглядає так: спочатку визначається точка масштабування і застосовується до всіх текстурних координат. Потім враховується коефіцієнт масштабування, який збільшує або зменшує текстуру, залишаючи фіксовану точку незмінною. Завершується процес нормалізацією координат, щоб вони залишались у межах 0 і 1, що забезпечує коректне відображення текстури.

Інструкція користувача

Запустивши програму, користувач бачить мінімальну поверхню Річмонда з 3 накладеними текстурами: дифузна, нормалей та дзеркальною. Внизу знаходяться слайдери, за допомогою яких можна редагувати щільність сітки об’єкту по u та v осям (його гранульованість). Навколо об’єкту рухається джерело світла по колу, тим самим показуючи всі особливості накладених текстур і створюючи гарний ефект переливання (рис. 4). З допомогою руху мишки під час затискання ЛКМ користувач може обертати фігуру навколо її центру.

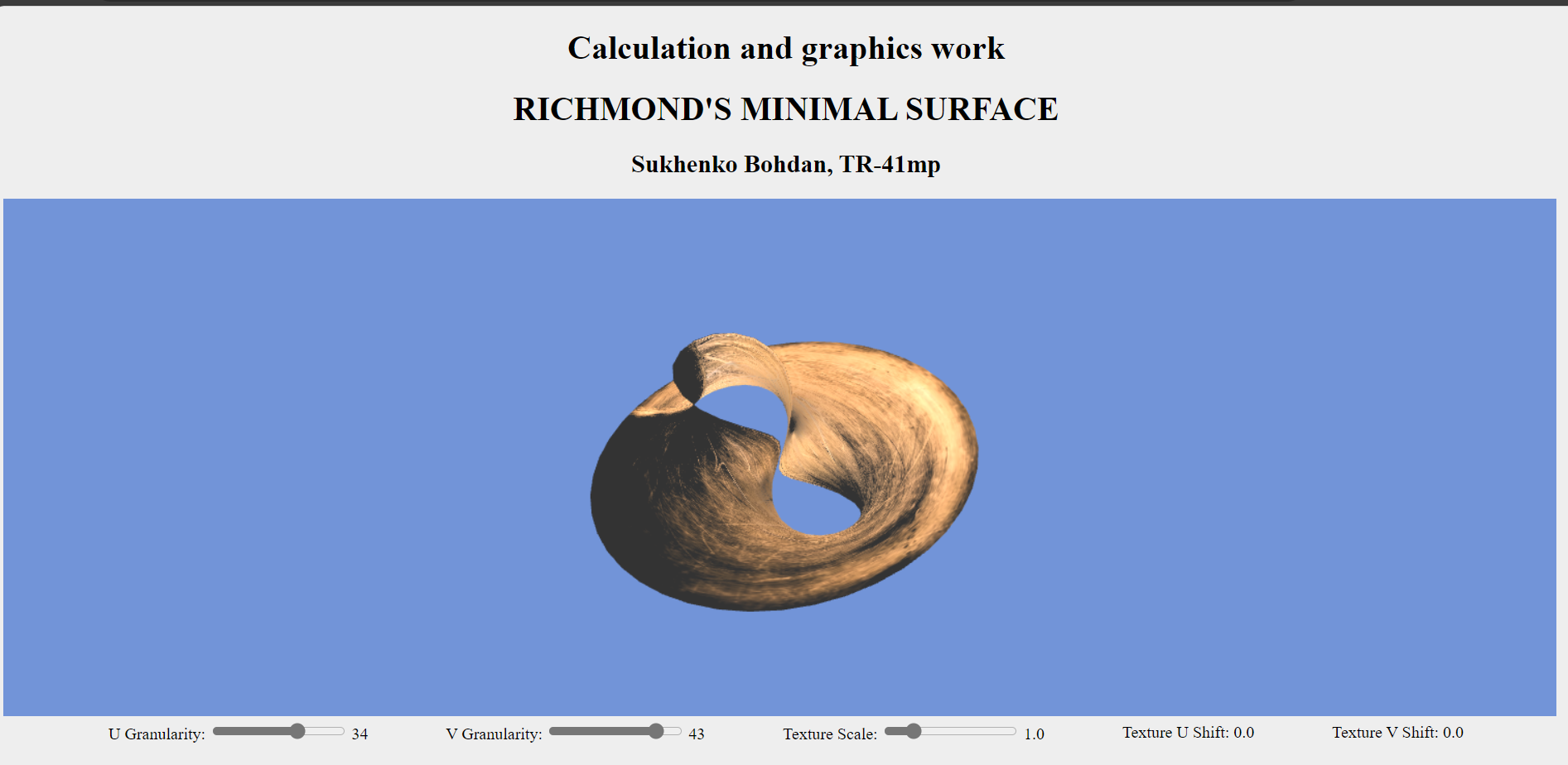


Рис. 4 – Мінімальна поверхня Річмонда з збільшеною гранульованістю

За допомогою третього слайдера внизу користувач може змінювати масштаб текстури навколо вказаної точки. Координати точка знаходяться далі – вони вказані в u-v координатах (координатних текстурах), де u = 0 відповідає лівому краю текстури, u = 1 – правому краю, а v = 0 відповідає нижньому краю текстури, в той час як v = 1 – верхньому (рис. 5):

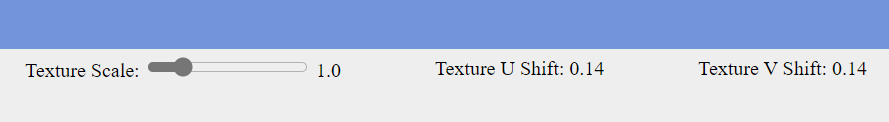
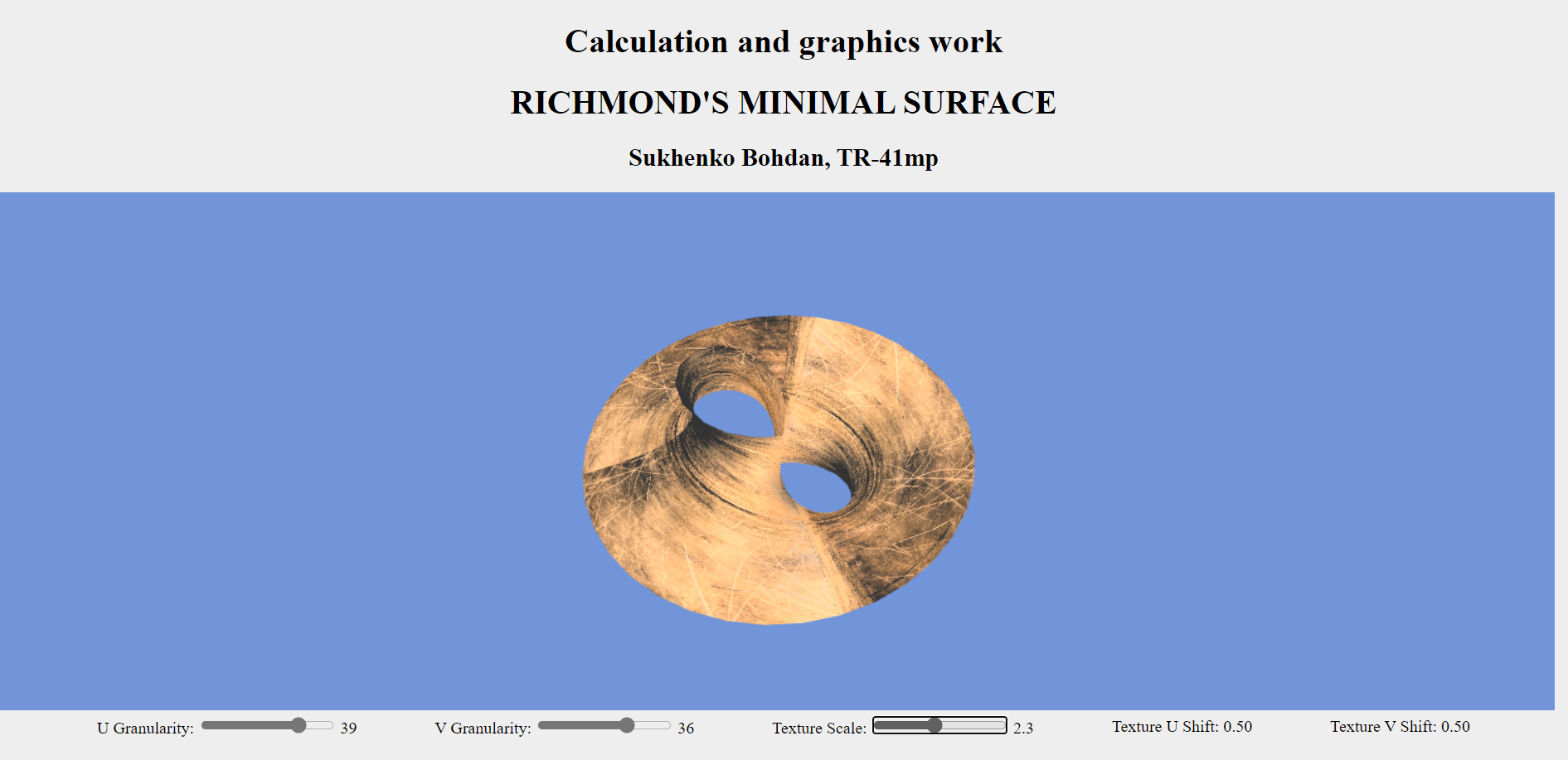


Рис. 5 – Інформація щодо зміни масштабу текстури

При натисненні кнопок W та S з увімкненою англійською розкладкою користувач може змінювати координату u точки, відносно якої відбувається масштабування. А кнопками D та A – відповідно координату v. Зміна відбувається з кроком 0.01 за натиснення.

На наступних скріншотах наведено об’єкт з текстурами, масштабованими відносно (0.5; 0.5) – тобто центру в текстурних координатах (дивлячись на об’єкт – він справа по центру).



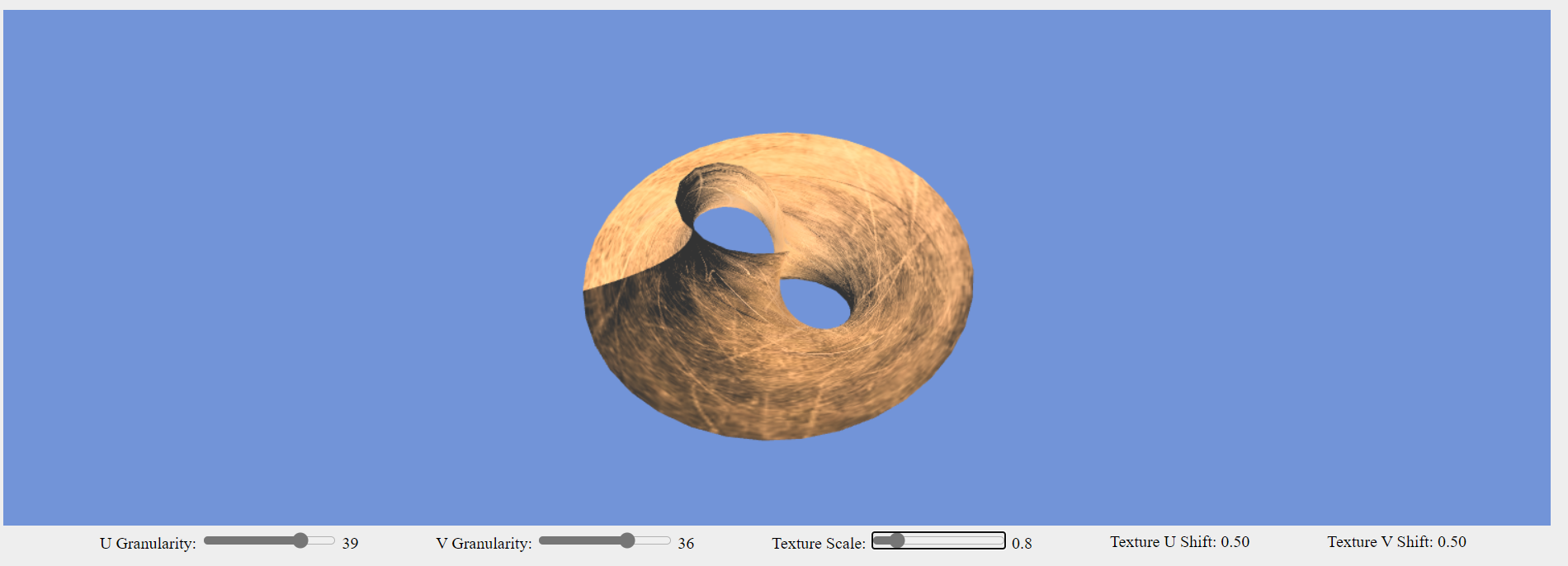


Рис. 6 – Різний масштаб текстури навколо вказаної точки

Далі, для прикладу, наведено скріншот масштабування текстури відносно точки (0; 0), тобто лівого нижнього краю.

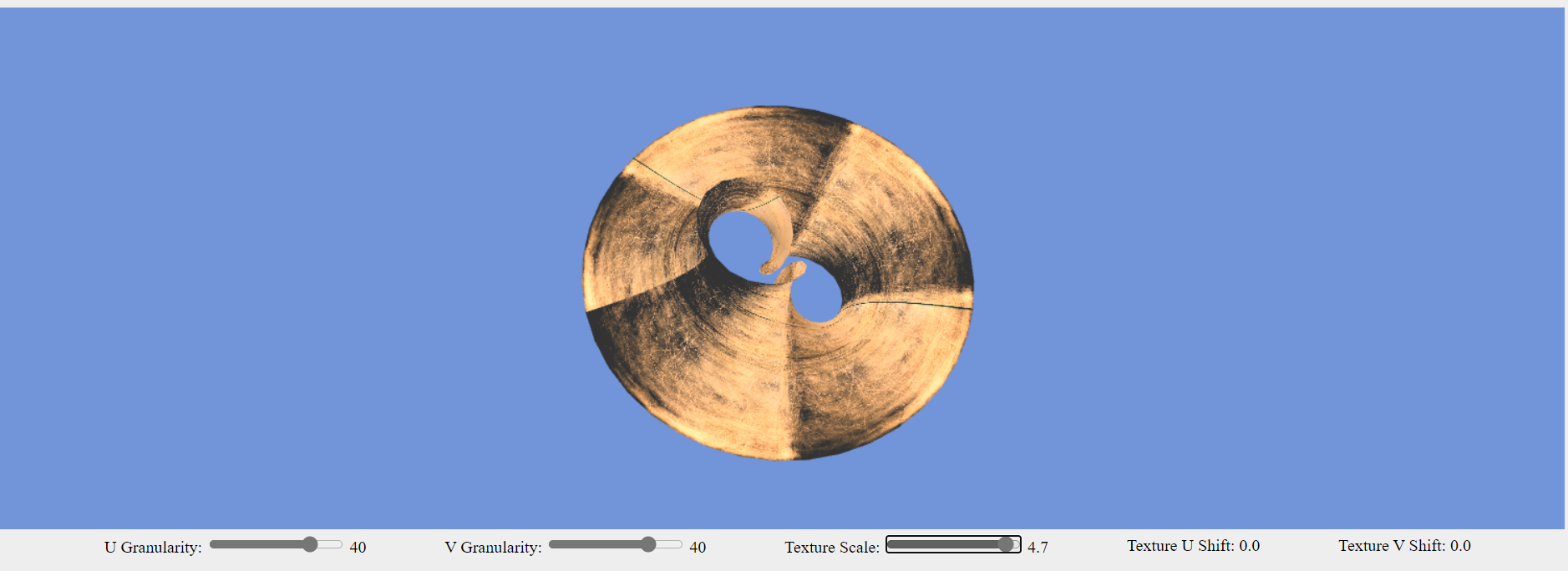


Рис. 7 – Масштабування відносно (0; 0)

Програмний код

//Текстурки мої текстурочки

let diffuseTexture, specularTexture, normalTexture;

//Масштаб текстури

let textureScale = 1.0;

//Зсув для масштабування, u-v координати точки

let uOffset = 0.0;

let vOffset = 0.0;

//Обробка тиків на клавіатурі

document.addEventListener('keydown', function(event) {

    const step = 0.01;

    switch (event.key) {

        case 's':

            uOffset -= step;

            break;

        case 'w':

            uOffset += step;

            break;

        case 'a':

            vOffset -= step;

            break;

        case 'd':

            vOffset += step;

            break;

        default:

            return;

    }

    //Не виходимо за границю

    if(uOffset > 1)

        uOffset = 1;

    if(vOffset > 1)

        vOffset = 1;

    if(uOffset < 0)

        uOffset = 0;

    if(vOffset < 0)

        vOffset = 0;

    //Оновлення показу координат

    textureShiftUValue.textContent = uOffset.toPrecision(2);

    textureShiftVValue.textContent = vOffset.toPrecision(2);

    //Оновлення поверхні

    let data = {};

    CreateSurfaceData(data);

    //Створення буфера

    surface.BufferData(data.verticesF32, data.normalsF32, data.texCoordsF32, data.indicesU16);

});

//Обчислення вершин поверхні

//З попередньої лаби

for (let i = 0; i <= uGranularity; i++) {

    let u = uMin + i \* uStep;

    for (let j = 0; j <= vGranularity; j++) {

        let v = vMin + j \* vStep;

        //Функція поверхні

        let x = -Math.cos(v) / (2 \* u) - (u \* u \* u \* Math.cos(3 \* v)) / 6;

        let y = -Math.sin(v) / (2 \* u) - (u \* u \* u \* Math.sin(3 \* v)) / 6;

        let z = u \* Math.cos(v);

        //Нормалізовані текстурні координати

        //Додаємо масштабування

        let texU = Math.abs(textureScale \* ((u - uMin) / (uMax - uMin) - uOffset)) % 1;

        let texV = Math.abs(textureScale \* ((v - vMin) / (vMax - vMin) - vOffset)) % 1;

        vertices.push(new Vertex([x, y, z], [texU, texV]));

    }

}