

Міністерство освіти і науки України
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерної інженерії

ОНУФРАК Ольга Василівна

**«Алгоритми перетворення елементів
цифрового зображення на основі технології
морфінгу / Algorithms for digital image elements
converting based on morphing technology»**

Студент групи КІм – 21
ОНУФРАК Ольга Василівна

Науковий керівник
к.т.н., Ю.М. Батько

Тернопіль – 2020

РЕЗЮМЕ

Кваліфікаційна робота на тему “Алгоритми претворення елементів цифрового зображення на основі технології морфінгу” зі спеціальності 123 «Комп’ютерна інженерія» освітнього ступеня «магістр» написана обсягом 90 сторінок і містить 33 ілюстрації, 3 таблиць, 2 додатків та 50 джерел за переліком посилань.

Метою роботи є розробка алгоритму перетворення елементів цифрового зображення на основі технології морфінгу для комп’ютерних систем обробки зображень.

Методи досліджень. Для розв’язання поставлених задач у кваліфікаційній роботі використано: методи: лінійної алгебри та аналітичної геометрії (для створення моделей опису зображень); афінних перетворень та технології морфінгу (для перетворення елементів зображень); об’єктно-орієнтованого програмування (для проектування програмних засобів аналізу та перетворення зображень).

Результати дослідження: алгоритми перетворення елементів цифрового зображення на основі аналізу його внутрішніх сегментів та технології морфінгу, програмна система аналізу та перетворення цифрових зображень.

Результати роботи можуть бути використані в створенні нових зображень, кіно індустрії, індустрії створення комп’ютерних ігор, для наукових досліджень та в навчальному процесі.

Орієнтовні напрямки розвитку досліджень: розроблення програмних систем для індустрії розваг, віртуальних подорожей в реальному часі шляхом створення нових спецефектів та шаблонів для навчання.

КЛЮЧОВІ СЛОВА: ПЕРЕТВОРЕННЯ ЗОБРАЖЕННЯ, АФІННІ ПЕРЕТВОРЕННЯ, ПРОГРАМНА СИСТЕМА, ВІДПОВІДНІ ТОЧКИ.

RESUME

Graduate qualification work on “ Algorithms for digital image converting elements based on morphing technology ” specialty 123 - Computer Engineering is 90 pages long and contains 33 illustrations, 3 tables, 2 appendices and 50 references.

The aim of the work is to develop an algorithm for converting digital image elements based on morphing technology for computer image processing systems.

Research methods. To solve the tasks in the qualification work used: methods: linear algebra and analytical geometry (to create models of image description); affine transformations and morphing technologies (for the transformation of image elements); object-oriented programming (for designing software for image analysis and conversion).

Research results: algorithms for conversion of digital image elements based on the analysis of its internal segments and morphing technology, software system for analysis and conversion of digital images.

The results can be used in the creation of new images, the film industry, the computer game industry, for research and in the educational process.

Indicative directions of research development: development of software systems for the entertainment industry, real-time virtual travel by creating new special effects and templates for learning.

KEYWORDS: IMAGE TRANSFORMATIONS, AFFINE TRANSFORMATIONS, SOFTWARE SYSTEM, CORRESPONDING POINTS.

ЗМІСТ

Вступ.....	7
1 Програмні, апаратні та поєднані системи обробки цифрових зображень	10
1.1 Комп'ютерна графіка, основні поняття та функції	10
1.2 Аналіз, синтез та обробки цифрових зображень	17
1.3 Програмні системи для аналізу, синтезу та обробки зображень	20
1.4 Постановка задач дослідження.....	28
1.5 Висновки до розділу.....	29
2 Методи та алгоритми перетворення об'єктів на цифрових зображеннях.....	30
2.1 Методи трансформації зображень на основі афінних перетворень.....	30
2.2 Алгоритми перетворення зображень на основі технології морфінгу..	40
2.3 Алгоритм перетворення елементів цифрового зображення.....	47
2.5 Висновки до розділу.....	50
3 Програмна система аналізу та перетворення цифрових зображень.....	51
3.1 Структура програмної аналізу та перетворення зображень.....	51
3.2 Програмні модулі системи перетворення елементів зображення.....	62
3.3 Тестування та аналіз реалізованої системи.....	68
3.4 Висновки до розділу.....	72
Висновки.....	73
Список використаних джерел.....	74
Додаток А Лістинг коду програми.....	78
Додаток Б Світлокопії виданих публікацій	85

ВСТУП

Актуальність роботи. Проблема моделювання та відображення поліморфізму об'єктів є однією з найбільш складних в комп'ютерній графіці, а при виконанні даного процесу в реальному часі взагалі вимагає складних математичних обчислень та високої якості апаратного забезпечення. Таке завдання часто виникає при створенні систем віртуальної реальності, до яких відносяться віртуальні музеї, віртуальні подорожі, різні автомобільні, авіаційні, морські та інші тренажери, багато комп'ютерних ігор.

Складність створення таких систем визначається необхідністю формування реалістичного відображення сцени зі швидкістю, достатньою для підтримки достатньо високої частоти оновлення картинки на моніторі. Наприклад, для відтворення ілюзії плавного руху в навчальних тренажерах дана частота не повинна бути нижче ніж 30 кадрів в секунду. Внаслідок поставлених обмежень для використання в сучасних програмних системах необхідно використовувати спеціальні моделі та методи, що володіють достатньою швидкістю, але при цьому, забезпечують відображення зображення, візуально мало відмінне від реального. Тобто синтезоване зображення, повинно бути цілком достовірним з точки зору користувачів систем, що використовують ці моделі і методи.

У багатьох комп'ютерних системах, наприклад, морських і авіаційних тренажерах, віртуальних подорожах або комп'ютерних іграх поліморфізм об'єктів є центральним елементом сцени, і їх моделювання має велике значення. Для вирішення даної задачі в основному застосовуються імітаційні методи, що представляють собою сукупність емпірично підібраних взаємозалежностей для досягнення візуальної подібності результату. Їх використання вимагає трудомісткого підбору параметрів і можливо лише для обмеженого кола завдань. З іншого боку, застосування в точних фізичних

моделей, розроблених для неінтерактивних додатків, стримується їх високою обчислювальною складністю.

Крім отримання геометрії об'єктів окрему проблему представляє їх достовірне відображення на екрані. Для цього можна використовувати методи, візуалізують статичні сцени. Однак на відміну від статичності сцени, геометрія динамічних об'єктів постійно змінюється, що вимагає перебудування сіток і робить весь процес надто трудомістким та сповільнює роботу програмних систем.

Таким чином для побудови реалістичного відображення поліморфних об'єктів на сценах, потрібні нові методи і алгоритми, які дозволили б отримати візуально схоже зображення і задовольняли поставленим вимогам по швидкодії. Тому задача створення алгоритму перетворення цифрових зображень на основі поліморфізм є актуальною.

Метою роботи є розробка алгоритму перетворення елементів цифрового зображення на основі технології морфінгу.

Для досягнення даної мети ставились наступні завдання:

- провести аналіз та класифікацію задач комп'ютерної графіки;
- проаналізувати методи синтезу, аналізу та обробки цифрових зображень;
- провести дослідження існуючих програмних систем синтезу, аналізу та обробки цифрових зображень;
- проаналізувати існуючі алгоритми перетворення елементів цифрових зображень;
- розробити алгоритм перетворення цифрових зображень на основі технології морфінгу;
- реалізувати програмну систему перетворення цифрових зображень, провести її тестування та порівняти з програмами аналогами.

Об'єкт дослідження – процес перетворення цифрових зображень в системах обробки зображень.

Предмет дослідження – методи та алгоритми обробки цифрових кольорових зображеннях.

Наукова новизна одержаних результатів визначається наступним чином:

– проведено комплексний аналіз та класифікацію алгоритмів перетворення об'єктів на зображеннях, що дозволило виділити їх переваги та недоліки та розробити власний алгоритм перетворення зображень на основі технології морфінгу;

– розроблено алгоритм перетворення елементів цифрового зображення на основі технології морфінгу, що дозволило зменшити обчислювальну складність процесу обробки зображень.

Практична цінність одержаних результатів полягає в тому, що:

– розроблено та проведено моделювання програмної системи обробки зображень з використанням елементів технології морфінгу, що дозволило в подальшому програмно реалізувати та провести дослідження запропонованих алгоритмів;

– реалізовано програмне забезпечення для аналізу та перетворення елементів цифрового зображення на основі мови програмування Delphi та з використанням алгоритмів морфінгу зображень.

Публікації та апробація до випускної кваліфікаційної роботи. За результатами наукових досліджень, проведених у випускній кваліфікаційній роботі, підготовлено тези [1,2] доповіді «Аналіз алгоритмів обробки цифрових зображень на основі технології морфінгу» обсягом 1 сторінка на III Науково-практичній конференції молодих вчених і студентів «Інтелектуальні комп'ютерні системи та мережі», а також «Аналіз алгоритмів сегментації цифрових зображень» обсягом 1 сторінка на III Науково-практичній конференції молодих вчених і студентів «Інтелектуальні комп'ютерні системи та мережі».

1 ПРОГРАМНІ, АПАРАТНІ ТА ПОЄДНАНІ СИСТЕМИ ОБРОБКИ ЦИФРОВИХ ЗОБРАЖЕНЬ

1.1 Комп'ютерна графіка, основні поняття та функції

Представлення візуальних цифрових зображень на дисплеї комп'ютера в графічному вигляді на початку було зроблено в 50-х роках з використанням тодішньої обчислювальних машин, для використання в військових і наукових дослідженнях. З тих часів візуальний спосіб відображення інформації став основною перевагою переважної більшості комп'ютерних систем, але головними споживачами все ж таки є персональних комп'ютери. Графічний користувацький інтерфейс на сьогодні є "де-факто" стандартом для програмного додатків різних класів, включаючи й операційних системи.

Виділена спеціальна область в науці інформатики, що досліджує методи і засоби обробки та синтезу цифрових зображень на основі програмно-апаратних комп'ютерних комплексів - комп'ютерна графіка. Дана галузь охоплює всі види і напрямки представлення цифрових зображень, доступних для ознайомлення людиною або на дисплеї монітора, або у вигляді твердої копії на відповідному носії (папір, тканина, кіноплівка тощо). Без засобів комп'ютерної графіки важко уявити собі не тільки сучасний комп'ютерний, але матеріальний світ. Візуалене відображення даних використовується в різних напрямках людської діяльності. Для прикладу назвемо туризм (комп'ютерні екскурсії), наукові дослідження (візуалізація векторних полів, структури речовини та інших даних), моделювання об'єктів, дослідно-конструкторські розробки.

Залежно від підходу до формування цифрових зображень комп'ютерну графіку можна поділити на растрову, векторну та фрактальну.

Особливості колірного відображення характеризують такі поняття, як кольорова графіка або чорно-біла.

На спеціалізацію в окремих напрямках розділяють назви деяких видів: наукова графіка, комп'ютерна поліграфія, інженерна графіка, Web-графіка тощо.

На стику комп'ютерних, кіно та медіа технологій створена та набирає обертів порівняно новий напрям - комп'ютерна анімація.

Слід відмітити, що все більше місце в комп'ютерній графіці займають розвагам. З'явився термін - механізм графічного представлення даних (Graphics Engine). Пропозиція ігрових програм досягає десятків мільярдів доларів і часто є причиною для чергового скачку вдосконалення анімації і графіки.

Хоча комп'ютерна графіка є лише інструментом для реалізації більших проектів, її методи і структура засновані на сучасних розробках прикладних і фундаментальних наук: математики, фізики, статистики, програмування тощо. Це твердження буде вірним як для прикладних програмних, так і для різних апаратних розробок для створення і опрацювання цифрових зображень на комп'ютерних системах. Тому комп'ютерна графіка – це один з найбільш активно прогресуючих напрямів інформатик, що в багатьох випадках виступає "локомотивом", що стимулює за собою всю ІТ-індустрію.

Основні області застосування комп'ютерної графіки на сьогоднішній день є надзвичайно різноманітними та наведені на рисунку 1.1.

Наукова графіка. Перші комп'ютерні системи застосовувалися лише для вирішення спеціалізованих виробничих і наукових завдань. Щоб отримані результати було зручніше оброблювати та аналізувати їх представляли для графічної обробки, будували діаграми, графіки, креслення складних розрахованих конструкцій. Перші візуальні приклади графіки на машині отримували в режимі посимвольного друку. Пізніше з'явилися спеціальні пристрої для створення копій - графопобудовники (плоттери) для різних креслення графіків і креслень чорнильним пером на папері. Сучасна наукова комп'ютерна графіка надає засоби для відображення результатів обчислювальних експериментів з наочним поданням їх результатів.



Рисунок 1.1 – Класифікація комп'ютерної графіки

Ділова графіка – це галузь комп'ютерної графіки, що використовується для наочного відображення різних прикладів роботи підприємств. Звітна документація, планові звіти, статистичні зведення – ця інформація, для якої за допомогою засобів ділової графіки проектують ілюстративні матеріали. Програмні додатки для ділової графіки формують електронні таблиці.

Конструкторська графіка застосовується для вирішення задач інженерів-конструкторів, винахідників нової техніки, архітекторів. Цей напрям комп'ютерної графіки є обов'язковим фактором САПР (систем автоматизації проектування). Засоби конструкторської графіки можна можна використовувати як для плоских зображень (проекції, перетину), так і для трьоївимірних проекцій.

Ілюстративна графіка – призначена для малювання або креслення на дисплеї комп'ютера. Пакети для роботи з ілюстративною графікою - це системи прикладного програмного забезпечення загального призначення. Найпростіші програмні системи для ілюстративної графіки як правило носять назви - графічні редактори.

Рекламна і художня графіка – на сьогоднішній день, це популярний напрям завдяки телебаченню. За допомогою програмних систем створюються комп'ютерні ігри, відеоуроки, рекламні ролики, мультфільми, відеопрезентації. Графічні програми для таких цілей вимагають великих апаратних ресурсів, та вимогливі до швидкодії та обсягів пам'яті. Відмінною рисою таких графічних пакетів є можливість генерування реалістичних цифрових зображень і "рухомих відеороликів". Отримання сцен тривимірних об'єктів, їх наближення, повороти, деформації, видалення пов'язано з великим обсягом математичних обчислень. Передача характеристик об'єкта в залежності від його положення, розташування джерела світла, тіней, від фактури поверхні, вимагає складних розрахунків в реальному часі.

Комп'ютерна анімація - це напрям, що забезпечує користувача можливостями для отримання рухомих цифрових зображень на дисплеї. Користувач створює на дисплеї малюнки початкового та кінцевого положення, всі проміжні варіанти розраховує та зображує програма, виконуючи математичний опис даного виду переміщення. Отримані зображення, що виводяться послідовно на дисплей з певною частотою, створюють ілюзію відео.

Мультимедіа - це поб'єднання чіткого зображення на дисплеї комп'ютера зі паралельним звуковим супроводом. Найбільшого використання системи мультимедіа отримали в області, розваг, реклами та навчання.

Комп'ютерну графіку прийнято розділяти на три основні види. Це растрова графіка, векторна графіка і фрактальна графіка (рисунок 1.2). Вони відрізняються принципами формування зображення при відображенні на екрані монітора або при друці на папері.

У растровій графіці цифрові зображення представляється у вигляді масиву кольорових точок. Такий підхід до відображення зображення називають растровим та часто використовується для різних задач.

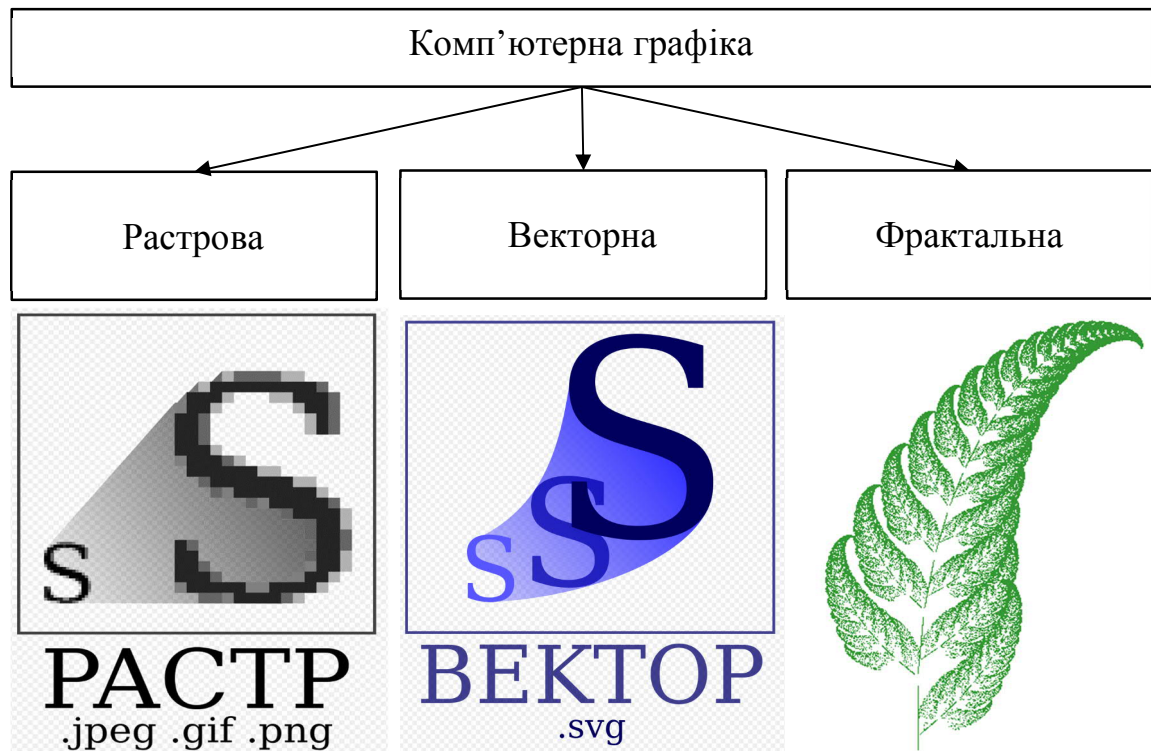


Рисунок 1.2 – Класифікація комп'ютерної графіки

Растрову графіку використовують при створення електронних (мультимедійних) і поліграфічних документів. Ілюстрації, зроблені засобами растрової графіки, як правило, створюють в ручному режимі за допомогою комп'ютерних систем, на основі сканованих ілюстрації, що попередньо підготовлені художниками або фотографії. Для генерації растрових цифрових зображень в комп'ютерній графіці широко застосовують цифрові апаратні фото- і відеокамери.

Більшість графічних програмних редакторів, використовуються для роботи з цифровими растровими ілюстраціями, та орієнтовані не стільки на синтез цифрових зображень, а в більшості випадків на їх обробку. В глобальній мережі в основному для передачі візуальної інформації поки застосовуються тільки растрові ілюстрації, оскільки їх просто отримати та вносити корективи, що значно спрощує роботу з ними та дозволяє без вилеких затрат створювати різні цифрові бібліотеки зображень.

Таблиця 1.1 - Порівняння растрової і векторної графіки

Критерій порівняння	Векторна графіка	Растрова графіка
Спосіб представлення зображення	Векторне зображення описується в вигляді масиву команд.	Растрове зображення будується з безлічі пікселів.
Подання об'єктів реального світу	Векторна графіка не дозволяє отримувати зображення природньої якості.	Растрові малюнки як правило застосовується для подання реальних образів.
Особливості друку зображення	Векторні малюнки іноді не друкуються або виглядають на папері не так, як хотілося б.	Растрові малюнки можуть бути легко надруковані на принтерах.
Якість редагування зображення	Векторні зображення можуть бути легко перетворені зі збереженням якості.	При масштабуванні і обертанні растрових картинок виникають спотворення.

Векторний метод – цей підхід в представлення зображення у вигляді набору геометричних примітивів (відрізків і дуг тощо). В даному випадку під вектором розуміють набір даних, що в повній мірі характеризують який-небудь природній об'єкт.

Програмні системи для обробки векторних зображень призначені в основному для синтезу ілюстрацій, а от для їх обробки використовуються рідко. Такі системи широко використовують в дизайнерських бюро, рекламних агенствах, видавництвах і редакціях. Оформлювальні роботи, що базуються на використанні шрифтів і найпростіших геометричних примітивів, вирішуються підходами векторної графіки набагато швидше.

Програмні системи для роботи з фрактальною графікою використовуються для генерації цифрових зображень шляхом в автоматичному режимі на основі різних математичних обчислень. Генерування фрактальної художньої композиції полягає не в оформленні або малюванні, а в програмному створенні.

Фрактальна графіка так само як і векторна - обраховується, проте відрізняється від неї іншою тим, що жодні візуальні об'єкти в пам'яті комп'ютера не зберігається. Зображення генерується за рівнянням (або на основі системи рівнянь), тобто на основі формул, які зберігати не треба. Провівши модифікацію коефіцієнтів у рівняння, можна згенерувати зовсім інше зображення.

Властивість фрактальної графіки відображати сцени живої природи обчислювальним підходом часто застосовують для автоматичного синтезу незвичних ілюстрацій.

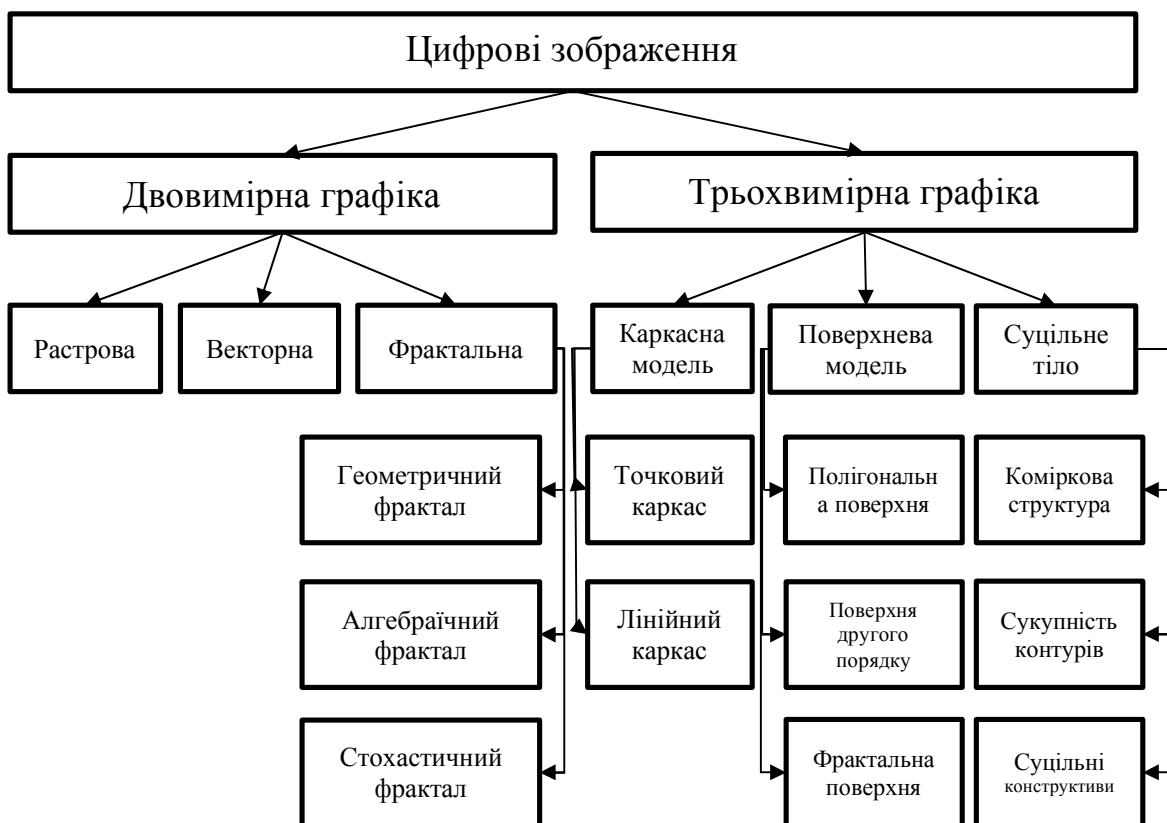


Рисунок 1.3 – Класифікація цифрових зображень

Цифрове зображення – це масив пікселів (даних), що отриманий в процесі дискретизації (аналого-цифрового перетворення) вхідного зображення (оригіналу) (рисунок 1.3). Після кодування на основі заданого алгоритму та запису на носій, цей масив даних іменується файлом.

Отож, провши аналіз та дослідивши основні сфери застосування комп'ютерної графіки можна зробити висновки, що даних напрямом в використанні комп'ютерної техніки на сьогоднішній день розвивається з великою інтенсивністю та має висорий рівень перспективності в подальших дослідженнях.

1.2 Аналіз, синтез та обробки цифрових зображень

Комп'ютерна графіка (computer graphics) - це область сучасної інформатики (науки про комп'ютери - computer sciences), що досліджує всі аспекти, пов'язані з синтезом та обробкою цифрових зображень за допомогою комп'ютерної техніки.

Обробка цифрових зображень - це процес перетворення цифрових зображень. Тобто вхідними даними в даному випадку є деяке зображення, а результатом відповідно теж зображення. Прикладом обробки цифрових зображень можуть бути: підвищення чіткості, контрасту, кореляція відтінків, згладжування, зменшення шумів, редукція кольорів тощо. основою для обробки можуть бути різноманітні цифрові зображення: радіолокаційні, космічні знімки, інфрачервоні зображення, відскановані зображення тощо. Основним завданням обробки цифрових зображень є як поліпшення деякого певного критерію, так і спеціальне спотворення, яке кардинально змінює саме зображення. В такому випадку обробка цифрових зображень фактично є проміжним етапом для подальшого процесу розпізнавання зображення. Для прокладу розглянемо таку ситуацію, перед розпізнаванням досить часто необхідно підкреслити контури

об'єктів, створювати бінарне відображення, просегментувати за кольорами. Існуючі алгоритми обробки зображення дозволяють вирішити різні задачі та залежать від того, яким чином отримане зображення: синтезовано системою комп'ютерної графіки чи це є результат оцифровки кольорової або чорно-білої фотографії.

Образотворча комп'ютерна графіка. В даному підході під об'єктами розуміють синтезовані зображення. Завдання даного напрямку це - побудова моделі об'єкта та подальша генерація зображення, в тому числі з перетворенням моделі, ідентифікація і отримання необхідної додаткової інформації.

Обробка та аналіз цифрових зображень. Під об'єктами будемо розуміти дискретне, числове відображення фотографій. Завдання даного напрямку є підвищення якості, оцінка вхідного зображення, в тому числі і визначення форми, розмірів, місця розташування та інших характеристик необхідних об'єктів. Окремим завданням є розпізнавання образів - виділення і класифікація властивостей об'єктів.

Отже, в основі аналізу та обробки цифрових зображень знаходяться підходи представлення, обробки і аналізу цифрових зображень, а також що є природньо, образотворча комп'ютерна графіка хоча б для того, щоб представити результати.

Комп'ютерна графіка – це сучасна технологія синтезу та аналізу графічних цифрових зображень засобами обчислювальної техніки. В своєму розрізі вона вивчає методи документування цифрових зображень на основі невізуальних даних або даних, створених безпосередньо користувачем.

Існують дві суміжні області інженерної діяльності. Перша - це розпізнавання образів (computer vision), завдання при цьому прямо протилежна завдання комп'ютерної графіки, тобто за наявним зображенню об'єкта отримати його опис. Друга - Обробка цифрових зображень (image processing). Цей напрямок пов'язаний з вирішенням проблем зберігання, стиснення, передачі, поліпшення зображення.

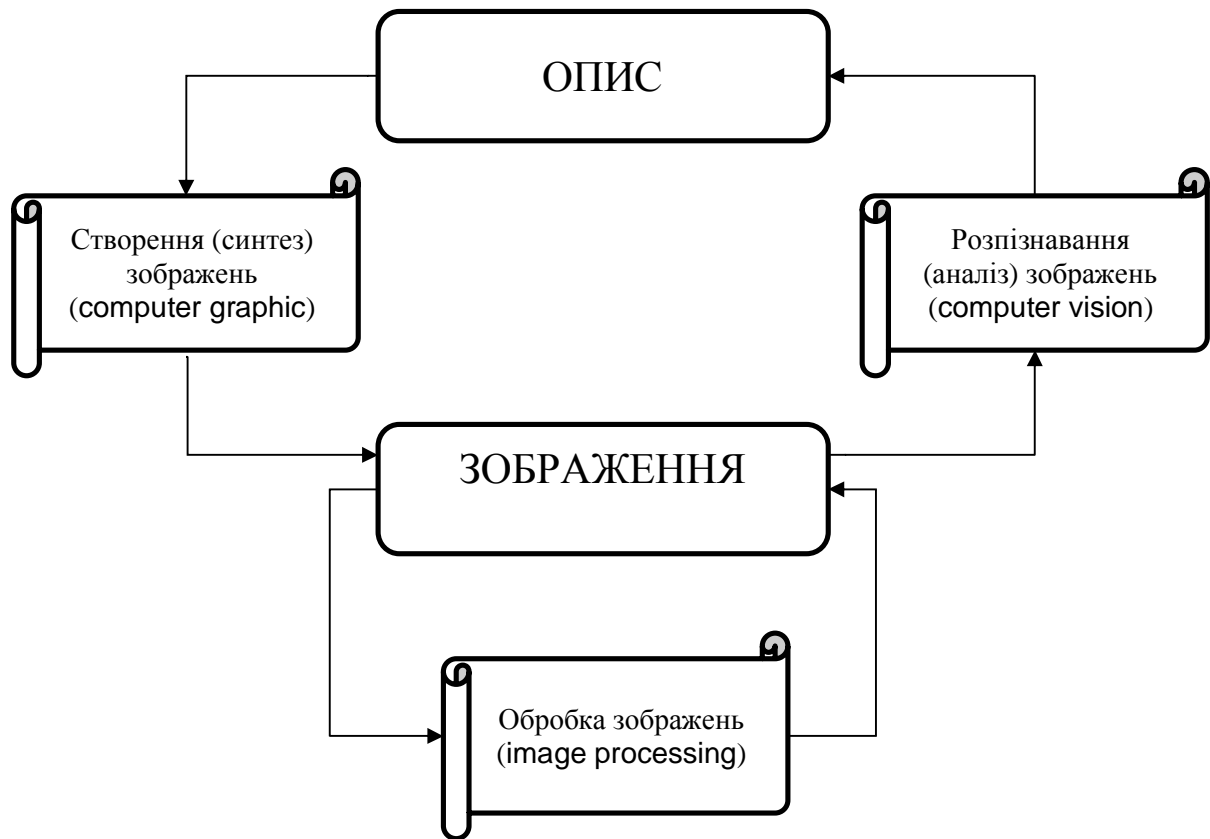


Рисунок 1.4 – Предметна область цифрової графіки та суміжних дисциплін

Раніше ці напрямки були абсолютно незалежні дисципліни. Зараз області їх перетину стають все ширше, а їх межі - все більш умовними.

Завданням комп'ютерної (машинної) графіки є візуалізація, (тобто створення зображення), що проводиться на основі інформації необразотворчого характеру. Під терміном «зображення» тут розуміється безліч точок, ліній і т.п., що виводяться на графічний пристрій (монітор, принтер та ін.). Підводячи підсумок наведеному короткому огляду можна відзначити наступне. Існує багато різноманітних застосувань комп'ютерної графіки, проте в будь-якому випадку для побудови зображення використовується одна або кілька обчислювальна машина з відповідними технічними засобами. Обчислювальна машина виконує програму, яка відповідно до деякого описом управляє висновком цифрових зображень різних.

Таким чином, Комп'ютерна графіка - інженерна дисципліна, яка вивчає прийоми, методи і алгоритми створення, зберігання і обробки описів об'єктів і їх цифрових зображень за допомогою комп'ютерної техніки.

1.3 Програмні системи для аналізу, синтезу та обробки зображень

На сьогоднішній день на ринку присутня велика кількість програмних систем для синтезу, редагування та подальшої обробки цифрових зображень. Такі системи дозволяють значно автоматизувати процес роботи з зображеннями проте все ще значну частину маніпуляцій треба робити вручну. Тривимірне зображення простіше створити на основі векторної графіки. Дана технологія надає можливість вказувати програмній системі команди, на основі яких будуть генеруватись зображення на основі реалізованих алгоритмів. Цей підхід схожий на креслення, причому в більшості випадків - тривимірне. На основі векторної графіки елементи сцени будуються з так званих "примітивів" - ліній, кубів, кіл, сфер, кривих тощо. При цьому елемент не потрібно малювати, а користувач просто повинен задати певний набір параметрів (наприклад: координати центру, кількість граней, радіус тощо), а вже комп'ютер сам на основі закладених алгоритмів проведе відображення. Складні об'єкти можна скласти з множини примітивів, на основі кривих (сплайнів) або, багатокутників (полігонів) причому сплайнові моделі при такому підході мають більш гладку форму, ніж полігональні. Потім користувач задає матеріали (текстури) та активує процес візуалізації (Rendering). Даний процес досить тривалий, проте після закінчення на виході отримується реалістичне зображення на основі заданої каркасної моделі і вказаними параметрами.

За своїм "професійним" призначенням програми системи для комп'ютерної анімації та графіки можна розділити на такі основні групи:

- програми двовимірної анімації, що використовуються для спецефектів;

- пакетні обробники комп'ютерної графіки для поліграфії;
- програми для двовимірного і тривимірного моделювання;
- програми двовимірної комп'ютерної живопису;
- пакети тривимірної анімації, для створення реклами або кінофільмів;
- презентаційні пакети;
- комплекси для обробки відеоцифрових зображень;
- програми для наукової візуалізації.

Програми комп'ютерної анімації та графіки є основними для поліграфістів і кінематографістів, художників і дизайнерів, кліп-мейкерів і вчених, розробників комп'ютерних ігор і творців освітніх програм, а також будь-яких користувачів, яким необхідно синтезувати, аналізувати або обробляти зображення різних типів.

Пакети комп'ютерної графіки які використовуються для створення поліграфічних зображень дозволяють поєднувати текст та ілюстрації різного походження, при цьому утворюючи дизайн веб-сторінок або документів з високою якістю.

Adobe Photoshop є еталонним прикладом для порівняння з іншими програмами системи для обробки растрової графіки. Ця програмна система надає можливості для корекції оцифрованих зображень, ретушування фотографій, надає можливість для застосування різних спецефектів та масок, необхідних для редагування або монтажу растрових цифрових зображень. Photoshop набув широкої популярності серед користувачів Macintosh, однак існують його версії для Windows, комп'ютерів Sun та Silicon Graphics (SGI). Наявні версії даної системи підтримують багат шарову структуру представлення цифрових зображень, а також містять масив інструментів для редагування і створення векторних зображень. Photoshop забезпечує користувачів різними фільтрами та великою кількістю масок, надає широкі можливості для роботи з кольором, відтінками, тінями, проекціями та для створення спецефектів.

Adobe Illustrator. Даний векторний пакет від розробників попередньої системи та розроблений для Windows, Macintosh та PowerMacintosh. Він забезпечує модивість для створення ілюстрацій та для розробки загального дизайну документів і є орієнтованим на відображення готових цифрових зображень з дуже високою роздільною здатністю. Ця програма дозволяє створювати символи та об'єкти довільної форми, а потім проводити процеси масштабувати, деформації та поворотів. Окрім цього, в ньому передбачено широкий спектр засобів для роботи з текстовими документами. Як правило, цю програму застосовують для роботи з фільтрами і спецефектами.

Corel Draw. Це досить поширена система серед пакетів ілюстративної графіки для операційної системи Windows від корпорації Corel Corp., що отримала статус класичною програмою додатку для векторного малювання. Corel Draw призначений не стільки для створення, підготовки графіків, а для обробки растрових цифрових зображень. В ньому присутні відмінні функції управління файлами та реалізовані можливості показу слайд-шоу на екрані комп'ютера, функції малювати від руки та модифікації шарів цифрових зображень, підтримує різні спецефекти, в тому числі тривимірні, та присутні гнучкі можливості для обробки текстів.

Комп'ютерна графіка для створення двовимірної графіки на SGI і Macintosh представлена наступними програмними продуктами як: PixelPaint Pro Collage, і StudioPaint 3D. Серед групи розробок для обробки зображень на платформі Macintosh варто відмітити пакет Collage фірми Specular International, що надає модивість компоувати зображення з високою роздільною здатністю, які були попередньо створені в середовищі Photoshop, і пакет для обробки растрового живопису на цифрових зображень PixelPaint Pro фірми Pixel Resources.

Серед множини програм комп'ютерної графіки для SGI окреме місце посідає пакет StudioPaint 3D фірми Alias|Wavefront, який дозволяє користуватись різними інструментами в режимі реальному часі при роботі на тривимірних моделях. Даний програмний продукт працює з необмеженою

кількістю шарів та забезпечує 30 рівнів скасування попередньої дії. StudioPaint 3D може використовуватись на спеціалізованих планшетах з чутливим пером, що дає можливість користувачу зробити традиційний ескіз зображення від руки, а потім переконвертувати зображення в тривимірні пакети для проведення моделювання або надання анімації та побудувати тривимірну модель по ескізу.

Astound. Paket Astound, один з основних пакетів що були створені останнім часом. Він є простий у використанні, надає можливість проведення синхронізації звуку з кліпи, рухомі зображення та інші ертефакти мультимедіа, в системі присутній великий інструментарій для створення анімації різних тривимірних діаграм, містить множину готових презентацій та шаблоні для популярних тем, забезпечений інструментами для роботи з різною графікою, динамічними зображеннями, а також звуком.

Adobe Persuasion, Demo-it, Novell Presentations, Director. Даний набір презентаційних пакетів містить програму Adobe Persuasion для OS/2 та Macintosh, що включає достатньо широко застосовані засоби компоунання, а також дизайну, програму Novell Presentations для операційної системи Windows з великим набором засобів для створення та зручним управлінням, а також власний пакет для слайд-шоу Demo-it фірми Lifeboat Publishers для Windows і відкритий для загального кристування презентаційний пакет Director фірми Macromedia, версії якого існують і для платформ Windows і Macintosh.

Двовимірна та трьохвимірна анімація, створюється як за допомогою традиційних (без використання обчислювальних систем), так і сучасними комп'ютерними методами, проте і ті і ті базуються на одному і тому ж принципі. А саме, якщо ряд статичних цифрових зображень буде відображатись в достатньо швидкому темпі, то в такій ситуації людське око скомпоує їх разом і буде сприймати як безперервне відео. Для того, щоб декілька цифрових відображень об'єкта були сприйняті людським оком як плавний імітацію руху цього об'єкту, може вистачити і невеликої швидкості, приблизно 8 комп'ютерних екранів/секунду. На відміну від традиційної анімації, при якій

кожен кадр промальовується в ручному режимі, у комп'ютерній 2D-анімації основну частину рутинної роботи забирає на себе програмна система. Можна, наприклад, задати рух по деякій траєкторії (програма створить на основі математичних розрахунків відповідні проміжні кадри) або плавно змінити кольорову палітру на протязом зміни декількох кадрів (наприклад, поступово освітлити зображення або прибрати частину з можливих кольорів).

Animo. Програма Animo від фірми Cambridge Animation Systems, відтворює техніку анімації "один до одного", тобто традиційну анімацію, що забезпечує її популярність серед професійних мультиплікаторів, які звикли створювати кадри за допомогою стандартних підходів. Серед інших професійних пакетів двовимірної анімації слід відмітити пакет Animation Stand, який розробила фірма Rainbow Technologies для Macintosh. Можливості даної програми, ще не дотягують до функціоналу програми Animo, але вже перевершують ресурси пакета Animator Studio.

Elastic Reality. Ще оодин програмний пакет для створення анамаційних файлів - Elastic Reality фірми AD SG, його функції призначені для двовимірних деформацій а також перетворень (морфінгу) відеоматеріалу. Аналізований пакет працює із незамкненими і замкнутими кривими, дозволяє керувати ступенем прозорості різних об'єктів, проводити згладження їх контурів та застосовувати різні спецефекти.

Програми двовимірного і тривимірного комп'ютерного моделювання застосовуються для різноманітних дизайнерських і інженерних задач. Крім того: вони можуть бути використані для доповнення програм тривимірної анімації, а також як поліграфічні і презентаційні пакети.

AutoCAD. Серед програм моделювання для операційної системи Windows один з лідерів є потужна система для машинного проектування AutoCAD фірми відомої фірми Autodesk. AutoCAD часто розглядають як деяке графічне ядродля розробки систем автоматизованого проектування (САПР). А саме, даний програмний продукт реалізує основні операції по написанню та редагуванню різних ліній, дуг і тексту, створює 2D- і 3D-моделі, проводить автоматизацію

рішень багатьох прикладних задач, що виникають під час проведення проектування, дозволяє користувачам адаптувати та налаштувати систему під конкретні додатки, при цьому також розроблений механізм створення власних сценаріїв і макрокоманди. Пакет включає в себе вбудовану мову програмування AutoLISP, що надає користувачам можливості формувати нові команди, а також використовувати різні мови програмування високого рівня.

Моделювання (2D і 3D) на SGI: Designer, Studio і AutoStudio. На компютерах під управлінням Silicon Graphics одними з потужніших програмних пакетів для дизайну та моделювання є серія програм фірми AliaWavefront - Designer, Studio і AutoStudio. Відмінною особливістю дизайнерських програмних розробок від Alias є можливість працювати з одночасну роботу з 2D і 3D моделями, та забезпечує гармонійну інтеграцію в існуючі на ринку системи автоматизованого проектування. Пакет Designer підтримує моделювання на високому рівні з використанням сплайнів, та надає ресурси для оцінки властивостей різних геометричних об'єктів, забезпечує зручні інструменти для проведення анімації і якісний модуль для рендерингу відео. Програми Studio і Auto Studio можуть включати в себе також різні засоби для роботи на Silicon Graphics, що базуються на багатопроцесорних моделях. Серед інших відомих систем автоматизованого проектування для SGI слід відмітити програму Vislab, що була розроблена фірми Engineering Animation, яка надає функціонал для створення візуальних рішень як для інженерних цілей так і для дизайну. Серед представлених на ринку систем автоматизованого проектування необхідно згадати програмний пакет Vislab фірми Engineering Animation. Дана розробка дозволяє створювати різноманітні візуальні рішення для інженерних цілей і дизайну.

Тривимірна анімація за технологією та принципом створення нагадує «лялькову»: користувач створює відповідні моделі (каркаси) об'єктів, накладає на них текстури, поєднує все це в єдину сцену, встановлює параметри освітлення і та характеристика камери, а на останньому етапі задає кількість кадрів у відео та напрям рух предметів. Подивитися, що відбувається можна за

допомогою камери, яка теж перебуває в русі. Переміщення об'єктів в тривимірному просторі задається на основі різних траєкторій, а також ключовим кадрам та з використанням формул, що будуть задавати рух окремих частин складних конструкцій. Підбравши необхідний темп та напрям руху, освітлення та текстури, користувач повинен запустити процес візуалізації. Програма прораховує всі необхідні проміжні кадри та повертає готовий фільм. Недоліком даного підходу є те що зображення буде мати надмірну гладкість форм і поверхонь, а також буде присутня деяка механістичність руху об'єктів.

Технологія синтезу реалістичних тривимірних цифрових зображень, на відміну від двовимірної анімації, при якій більшість може бути намальовано від руки, вимагає значних ресурсів. В тривимірній анімації об'єкти занадто гладкі, їхня форма при відображенні є занадто правильна і рухи йдуть по занадто "геометричним" траєкторіях, проте сучасні програмні продукти здатні вирішити більшість з перелічених недоліків. У анімаційних програмних пакетах покращуються засоби для візуалізації, оновлюються існуючі інструменти для створення різних спецефектів та збільшуються бібліотеки текстур. Для створення більш природніх об'єктів, наприклад, диму або волосся, використовується сучасна технологія формування об'єкту з множини незалежних частинок. Додається інверсна кінематика, а також інші техніки оживлення, розробляються нові методи співставлення відеозапису й анімаційних ефектів, що надає модивість зробити сцени і рухи у більшій мірі реалістичними. Крім того, підхід на основі відкритих систем дозволяє працювати відразу з декількома різними пакетами. Можна створити спеціалізовану модель в одному пакеті, розфарбувати її в іншому, а надати їй принципи руху в третьому, доповнити відеозаписом у четвертому. Ну і звичайно, функції багатьох професійних пакетів можна сьогодні розширити та розвинути за допомогою додаткових модулів, написаних спеціально для базового пакету.

3D Studio і 3D Studio MAX. Один з найпоширеніших пакетів 3D-анімації - це 3D Studio фірми Autodesk. Програма працює для операційної системи

Віндовс, забезпечує всі етапи створення тривимірного фільму, а саме: моделювання різних об'єктів і формування сцени, анімацію і візуалізацію, функції роботи з відео. Крім того, існує широкий напрям прикладних програмних систем (IPAS-процесів), написаних спеціально для програми 3D Studio. Інтерфейс даної програми єдиний для всіх модулів та реалізує високий ступінь інтерактивності. 3D Studio MAX реалізує в собі розширені можливості по управлінню анімацією, дозволяє зберігати історію для кожного з роботчих об'єктів і дозволяє генерувати різноманітні світлові ефекти, підтримує 3D-акселератори і головне має відкриту архітектуру, тобто надає модивість третім фірмам інтегрувати в систему додаткові додатки.

Electric Image, SoftImage. Для створення тривимірної анімації на комп'ютерах типу IBM і Macintosh звичайно зручно використовувати пакет Electric Image Animation System, що включає в себе надзвичайно великий комплекс анімаційних засобів, функцій, спецефекти, інструментарій для роботи зі звуком та світлом, генератор шрифтів з параметри яких можна налаштовувати від потреб користувача. Хоча у цієї програми немає засобів моделювання, але зате є можливість імпорту понад тридцять різних форматів моделей. Пакет також підтримує роботу з ієрархічними об'єктами і засобами інверсної кінематики. Вона підтримує моделювання на базі стандартних полігонів і сплайнів, забезпечує створення спецефектів, роботу з різними частинками і технологію перенесення активного руху з живих акторів на комп'ютерних персонажів або обекти.

Infini-D для Macintosh. Infini-D підтримує три основні процеси: візуалізацію, створення тривимірних об'єктів, а також анімацію. Пакет допускає під час робти шість базисних елементів, на основі яких користувач можете конструювати свої тривимірні об'єкти різної складності, а також додатковий вбудований редактор, що дозволяє розробляти власні "будівельні блоки". Власні бібліотеки програми Infini-D містять великий вибір типів для заливки та текстури, а, крім того, користувач можете створювати нові кольорові гами і імпортувати матеріали (за допомогою зовнішніх апаратних пристроїв).

PowerAnimator на SGI. Засоби даного пакету дозволяють проводити моделювання зі складними ієрархічними об'єктами і поверхнями, що кодуються та взаємодіють між собою. Це один з найбільш потужних і зручних програмних комплексів інструментів для проектування та створення об'єктів на основі полігонів і, головне, сплайнів. Пакет надає можливість спроектувати реалістичні образи, зокрема, завдяки відмінним можливостям освітлення - спрямованого або розсіяного, з використанням блікі і інших сучасних ефектів.

Анімація в пакеті створюється на основі послідовності ключових кадрів, включаючи інверсну кінематику, а також, технологію перенесення руху з живих акторів на комп'ютерних персонажів, що забезпечує отримати дуже природні рухи об'єктів. Крім того, пакета PowerAnimator підтримує важливі засоби анімації м'язів і міміки, а також інструменти більш точної синхронізації звуку і зображення.

Програми обробки цифрових відеоцифрових зображень і створення багат шарових композицій з використанням двовимірної і тривимірної графіки на сьогоднішній день дозволяють замінити комбіновані зйомки, обробляти відзнятий матеріал в реальному часі засобами комп'ютерної графіки, суміщати відзнятий матеріал та елементи комп'ютерної анімації та виводити результати на кіно- і відеоплівку.

1.4 Постановка задач дослідження

В даному розділі проаналізовано задачі та сфери застосування методів та підходів комп'ютерної графіки. Проведено методів та алгоритмів обробки цифрових зображень різного виду, а також виділено їх основні характеристики. Проаналізовано програмні комплекси для синтезу, аналізу та обробки цифрових зображень.

Для досягнення поставленої мети необхідно розв'язати наступні задачі.

- провести аналіз та класифікацію задач комп'ютерної графіки;
- проаналізувати методи синтезу, аналізу та обробки цифрових зображень;
- провести дослідження існуючих програмних систем синтезу, аналізу та обробки цифрових зображень;
- проаналізувати існуючі алгоритми перетворення елементів цифрових зображень;
- розробити алгоритм переворення цифрових зображень на основі тезнології морфінгу;
- реалізувати програмну систему перетворення цифрових зображень, провести її тестування та порівняти з програмами аналогами.

1.5 Висновки до розділу

Проведено дослідження та класифікацію сфер застосування комп'ютерної графіки, що дозволило виділити основні напрямки та алгоритми які використовуються перетворення цифрових зображень.

Проаналізовано алгоритми обробки цифрових зображень, що дозволило вибрати групу алгоритмів для проведення перетворень цифрових зображень, а також їх переваги та недоліки.

Проведено аналіз програмних систем обробки цифрових зображень який дозволив виділити основні структурні модулі, а встановити інтерфейси обміну даними між ними.

2 МЕТОДИ ТА АЛГОРИТМИ ПЕРЕТВОРЕННЯ ОБ'ЄКТІВ НА ЦИФРОВИХ ЗОБРАЖЕННЯХ

2.1 Методи трансформації зображень на основі афінних перетворень

Афінні властивості геометричних фігур - це такі, які зберігаються при афінних перетвореннях. Термін «афінний» утворений від латинського слова *affinis* - «споріднений». Афінний перетворення - взаємно однозначне відображення площини або простору на себе, при якому трьом точкам, лежачим на одній прямій, відповідають три точки, також лежать на одній прямій.

Бачення про проекційні властивості геометричних фігур можна отримати з перспективних зображень. Геометрично процес перспективного зображення складається з двох частин: з точки O проводяться прямі лінії до кожної точки геометричної фігури F ; ці прямі лінії, які виходять з точки O , перетинаються деякої площиною. Проведення прямих ліній, що з'єднують точку O зі всіма точками фігури F , називається проектуванням фігури F з точки O . Якщо сукупність прямих, що виходять з точки O , перетинається площиною, то сукупність точок перетину цих прямих ліній з площиною називається перетином прямих ліній площиною. Проектування і перетин є одними з основних побудов проєктивної геометрії. За допомогою цієї побудови кожній точці фігури F ставиться у відповідність певна пряма, що проходить через точку O , і, взагалі кажучи, кожної прямої, що проходить через точку O , ставиться у відповідність певна точка площині за деякими винятками.

Прикладом афінного перетворення площини є рівномірне стиснення площини до прямої з коефіцієнтом до >0 , при якому будь-яка точка прямої a залишається на місці, а будь-яка точка A , що не лежить на цій прямій, переходить в точку A' , яка лежить на тому ж перпендикуляр до прямої a , що і точка A , але на відстані від a , збільшеному $c>1$ або зменшеному $d<1$ раз.

Покажемо, що при стисненні площини до прямої a всяка пряма переходить в пряму (рисунок 2.1). Нехай пряма OA перетинає пряму a в точці

O , і нехай $d < 1$. Точка O при такому перетворенні залишається на місці. Точка A переходить в точку A' . Але і будь-яка точка B прямої OA перейде в деяку точку B' прямої OA . Так як:

$$A'T = d \cdot AT, B'P = d \cdot BP, \text{ то } A'T/AT = B'P/P. \quad (2.1)$$

Значить, для всіх точок A' і B' , в які перейдуть різні точки прямої OA , тангенс кута, утвореного прямою a з прямою OA' і з прямою OB' , один і той же.

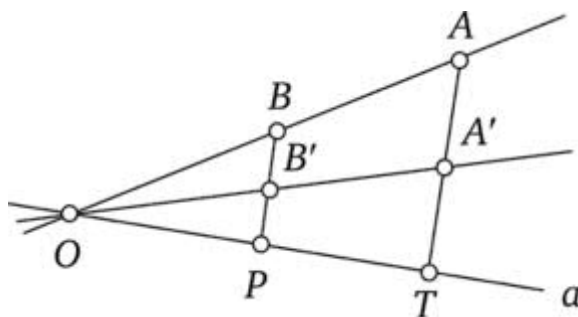


Рисунок 2.1 – Проекція прямої на основі відображення

Покажемо, що стиснення площині до прямої зберігає відношення паралельності прямих, тобто паралельні прямі переходять також в паралельні. Нехай прямі AT і BK паралельні (рисунок 2.2). Дійсно, якщо тангенси кутів, які утворюють прямі AT і BK з прямою a однакові, в силу того що AT і BK паралельні, то і їх образи AT' і BK' утворюють з прямою a кути, тангенси яких відрізняються тільки множителем d , тобто однакові.

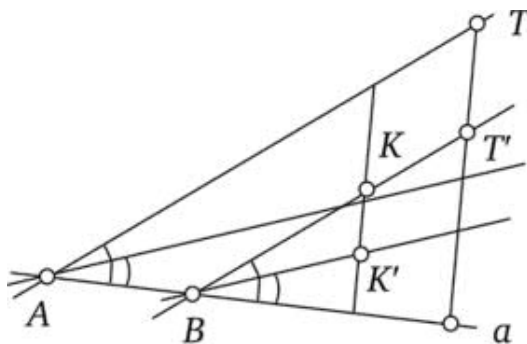


Рисунок 2.2 – Проекція паралельних прямих на основі відображення

Ставлення паралельності прямих зберігається при будь-якому афінному перетворенні. Якщо на площині взяти пучок паралельних прямих, то при афінному перетворенні він перейде також в пучок паралельних між собою прямих. При цьому може змінитися їх напрямок, відстані між ними, але так, що всі відстані змінюються однаково, так як відстань між будь-якими двома точками множиться на одне й те саме додатне число. Це означає, що при афінному перетворенні дана сітка рівних паралелограмів перетворюється в деяку іншу сітку рівних же паралелограмів. Наочний образ афінного перетворення дає малюнок 2.3.

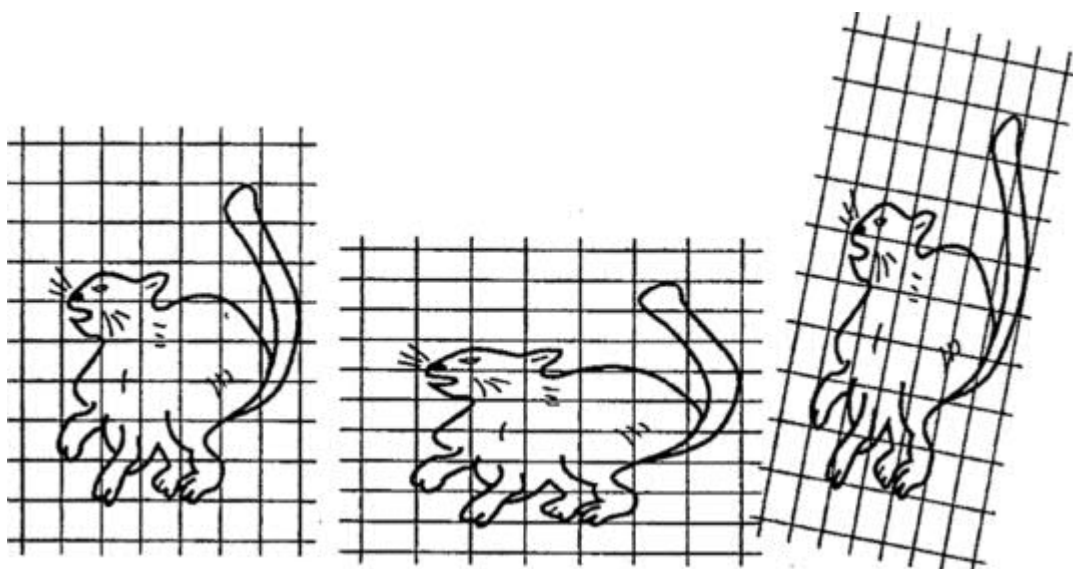


Рисунок 2.3 – Приклади афінних перетворень зображення

Афінну площину можна отримати з проєктивної площині, якщо довільну пряму проєктивної площині взяти в якості нескінченно віддаленої і по цій прямій зробити її «розріз». Тоді всі прямі, що перетиналися в точках нескінченно віддаленої прямий на отриманій таким чином афінній площині, стають паралельними.

Відзначимо, що при афінному перетворенні пересічні прямі переходять також в пересічні прямі. Дійсно, афінні перетворення на площині є взаємно однозначним. Тому образ точки перетину двох прямих є точкою, що належить

образам цих прямих, і, отже, афінніє образи цих прямих також мають спільну точку, то є перетинаються.

Афінний перетворення площині повністю визначено, якщо дані образи будь-яких трьох точок площини, які не лежать на одній прямій.

Це означає, що якщо дані образи трьох точок, які не лежать на одній прямій, то цим визначається образ будь-якої іншої точки площині при цьому перетворення.

Нехай точки A, B, C при деякому афінному перетворенні перейшли в точки A', B', C' відповідно. Так як точки A, B, C лежать в одній площині, то вони є вершинами трикутника. Їхні образи – точки A', B', C' також вершини трикутника, так як пересічні прямі при будь-якому афінному перетворенні переходять в прямі, які перетинаються (рисунок 2.4).

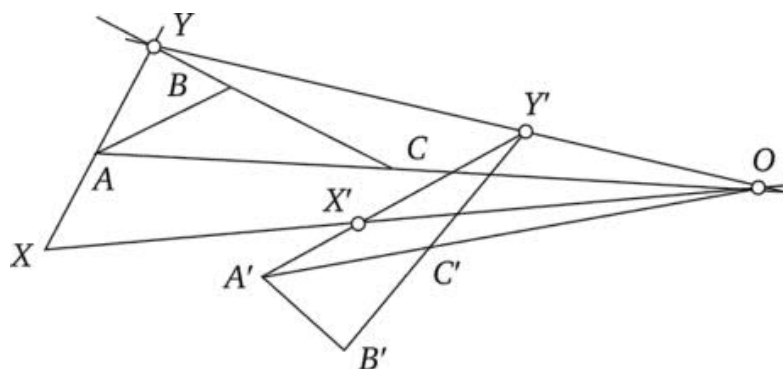


Рисунок 2.3 – Приклади афінних перетворень на площині

Візьмемо на площині довільну точку X і побудуємо її образ при цьому перетворення. Проведемо пряму AX . Вона перетинає хоча б одну з прямих, на яких лежать боку трикутника ABC . Нехай AX перетинає пряму BC в точці Y . Побудуємо точку O - точку перетину прямих AC і $A'C'$. При цьому афінному перетворенні точка O переходить сама в себе, так як належить прямій AC і її образу $A'C'$, тобто є нерухомою точкою даного перетворення. Образ точки Y повинен лежати на прямій OY , але він також повинен лежати на прямій $B'C'$, так як Y належить прямій BC . Отже, образ точки Y - точка Y' є точка перетину прямих OY і $B'C'$. Образ точки X повинен належати як прямій OX , так і прямій

$A'Y'$, в силу того що X належить OX і $A'Y'$. Перетин прямих OX і $A'Y'$ є образ точки X – точка X .

Окремим випадком афінних перетворень є таке, при якому будь-яка пряма переходить в пряму, їй паралельну. Таке перетворення називається гомотетія.

Розглянемо приклад перетворення гомотетії трикутника. Так як при афінному перетворенні все відстані змінюються однаково, то трикутники $A'B'C'$ і ABC такі, що $A'B'C'$ є гомотетичним чином трикутника ABC , відповідні сторони повинні бути паралельні і пропорційні. Безпосередньою побудовою можна виявити, що прямі AA' , BB' , CC' перетинаються в одній точці, яка називається центром гомотетії (рисунок 2.4).

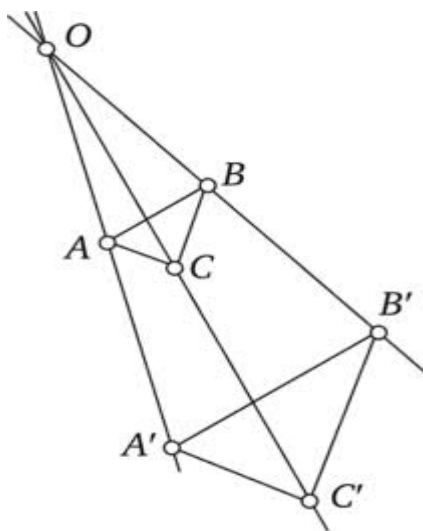


Рисунок 2.4 – Приклад афінних перетворень лінії в паралельну лінію

Якщо точка O - центр гомотетії, то кожній точці M площини ставиться у відповідність точка M' така, що точки M та M' лежать на промені OM і $OM' = d \cdot OM$, де d - *const*, відмінне від нуля число, зване коефіцієнтом гомотетії.

Проекційні перетворення. Нехай F деяка фігура, яка проектується з точки O , що не лежить в площині фігури. У перетині проектують прямих деякої площиною виходить нова фігура F' . Встановлена таким чином відповідність між елементами фігур F і F' називається перспективним перетворенням або

перспективною відповідністю. Якщо тепер проектувати фігуру F' з нового центру O' на третю площину, то отримаємо нову фігуру G , яка виходить з фігури F за допомогою двох послідовних перспективних перетворень.

Результат деякої послідовності перспективних перетворень називається проєктивним перетворенням або проєктивним відповідністю. Ті властивості фігур, які не змінюються при проєктивних перетвореннях, є проєктивними властивостями фігур.

Звідси безпосередньо випливає, що поняття паралельності прямих, рівності відстаней і кутів не є проєктивними властивостями. З іншого боку, точки і прямі при проєктивному перетворенні завжди переходять відповідно в точки і прямі, незалежно від того, скільки операцій проектування і перетину було вироблено при здійсненні проєктивного перетворення; при цьому якщо точка A лежить на прямій l фігури F , то відповідна їй точка A' лежатиме на прямій t , що відповідає прямій l . Двом пересічним прямим будуть відповідати дві пересічні прямі, трикутнику буде відповідати трикутник. Зі сказаного ясно, що властивості, що стосуються тільки приналежності точок і прямих, є проєктивними властивостями, а ті властивості, які пов'язані з вимірюванням, тобто метричні властивості, проєктивними, взагалі кажучи, не є.

Осьова симетрія або відображення від осі. Нехай на площині дана деяка пряма a . Відображенням від осі a називається таке перетворення площини, при якому кожній точці A площини ставиться у відповідність точка A' , яка лежить на перпендикуляр до прямої a , опущеному з A на пряму a , і знаходиться від a на тій же відстані, що і точка A . Пряма a називається віссю симетрії. Ясно, що при такому перетворенні всі крапки на прямій a переходять в себе, тобто пряма a залишається нерухомою.

Покажемо, що відображення від осі є рухом. Для цього потрібно встановити, що якщо A і B - будь-які дві точки площини, а точки A' і B' - їх образи при відображенні від осі, то $A'B' = AB$.

Це твердження очевидно, якщо точки A і B лежать на осі a . Якщо A і B не лежать на прямій a , то можливі три випадки:

- пряма AB паралельна прямій a ;
- пряма AB перпендикулярна прямій a ;
- пряма AB ні паралельна, ні перпендикулярна прямій a .

У разі якщо $ABA'B'$ - паралелограм (рисунок 2.5), так як $AA'=BB'$ і AA' паралельна прямій BB' за визначенням осової симетрії, отже, $AB=A'B'$.

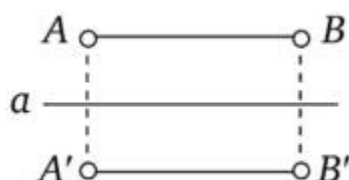


Рисунок 2.5 – Приклад відображення лінії

У випадку якщо точки A' і B' також лежать на прямій AB . Нехай C - точка перетину прямої a з прямою AB , тоді $AB=AC-BC$, $A'B'=A'C'-B'C'$. Але $AC = A'C$, $BC=B'C$, отже, $AB=A'B'$.

У разі якщо буде третій випадок, то проведемо з точок A і A' перпендикуляри AC і $A'C'$ до прямої BB' (рисунок 2.6). Точки C і C' симетричні відносно прямої a й, по доведеному, $AC= A'C'$, $BC=B'C'$, отже, трикутники ABC і $A'B'C'$ рівні за двома катетами, звідси $AB=A'B'$.

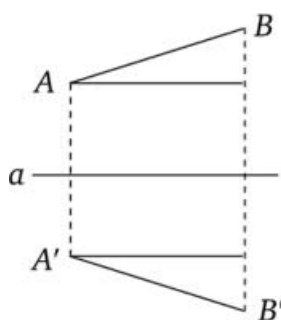


Рисунок 2.6 – Приклад відображення лінії

Поворот площини навколо даної точки. Нехай O - деяка точка площині, звана центром повороту, ϕ - градусна міра деякого кута, званого кутом повороту.

Поворотом площини навколо точки O на кут ϕ називається перетворення площини, при якому точка O переходить сама в себе, а кожна точка A , відмінна від O , переходить в точку A' , таку що $OA = OA'$, градусна міра кута AOA' дорівнює ϕ , причому всі точки площини, відмінні від O , повертаються навколо точки O в одному і тому ж напрямку - або за годинниковою стрілкою, або проти годинникової стрілки.

Покажемо, що поворот є рух, тобто що $AB = A'B'$, якщо A' і B' - образи точок A і B при повороті площини навколо точки O на кут ϕ .

Якщо точки O, A і B лежать на одній прямій (рисунок 2.6), то $AB = OA - OB$, $A'B' = OA' - OB'$, але $OA = OA'$, $OB = OB'$ за визначенням повороту, отже, звідси випливає, що $AB = A'B'$.

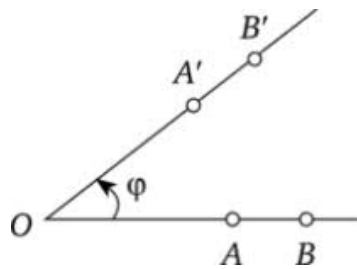


Рисунок 2.6 – Приклад повороту відносно деякої точки

Якщо точки O, A, B не лежать на одній прямій, то трикутник OAB дорівнює трикутнику $OA'B'$ по двох сторонах і куту між ними. Кут AOB дорівнює куту $A'OB'$, так як кожен з них дорівнює різниці між кутом ϕ і кутом $A'OB$. Це проілюстровано на рисунку 2.7.

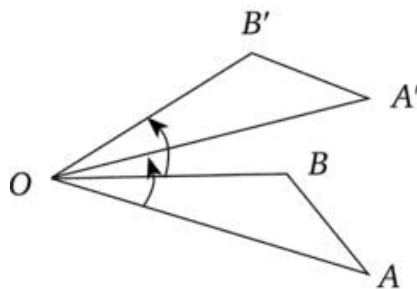


Рисунок 2.7 – Приклад повороту площини відносно деякої точки

Симетрія плоских фігур. Нехай F деяка плоска фігура. Рух, під час якого кожна точка фігури F переводить в точку цієї ж фігури, називається перетворенням симетрії цієї фігури.

Таким чином, симетрія фігури характеризується сукупністю рухів, які переводять фігуру в себе. Покажемо, що сукупність перетворень симетрії даної фігури утворює групу щодо операції композиції рухів. Дійсно:

1) якщо два рух у площині є перетвореннями симетрії даної фігури, то їх композиція також є перетворенням симетрії даної фігури, так як кожне з перетворень фігури переводить будь-яку її точку в точку, що належить цій же фігурі, і, отже, послідовне виконання двох таких перетворень має тим же властивістю;

2) тотожне перетворення також є перетворенням симетрії (кожна точка фігури залишається на місці), з іншого боку, композиція будь-якого перетворення симетрії з тотожним перетворенням є те ж саме перетворення симетрії даної фігури;

3) перетворення, зворотне будь-якого перетворення симетрії даної фігури, також є її перетворенням симетрії, так як кожен рух повертає в початкове положення, тобто в точку даної ж фігури.

Відзначимо, що паралельний перенос і поворот площині навколо даної точки є рухами площини «по собі», в той час як відображення від прямої як би «виносить» будь-яку точку з площини, повертаючи її назад на площину. Паралельний перенос і поворот площині навколо даної точки - це руху першого роду, а відображення від прямої - рух другого роду.

У тому випадку, коли відбиття від деякої прямої є перетворенням симетрії деякої фігури, цю пряму називають віссю симетрії фігури. У фігури може бути більше однієї осі симетрії, наприклад, у прямокутника дві осі симетрії, кожна з яких проходить через середини протилежних сторін прямокутника. Справедливо наступне твердження: якщо деяка фігура має дві осі симетрії, то вона має і поворот як перетворення симетрії навколо точки

перетину осей на кут, рівний подвоєному куту, під яким перетинаються осі симетрії цієї фігури.

Дійсно, нехай прямі a і b - осі симетрії фігури F , точка O - точка перетину осей, які перетинаються під кутом ϕ (рисунок 2.8).

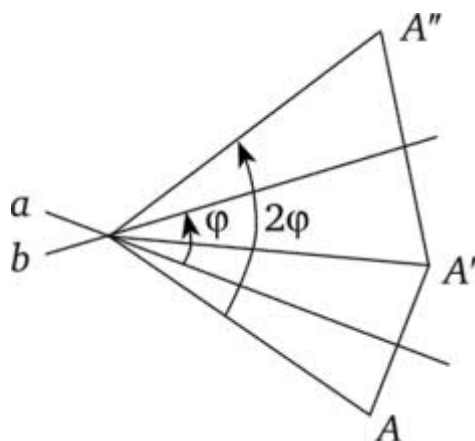


Рисунок 2.8 – Приклад семетричного переносу плоских фігур

Нехай A' - образ точки A при відображенні від осі a , точка A - образ A' при відображенні від прямої. Тоді трикутники AOA' і $A'O A''$ рівнобедрений в силу визначення відображення від осі. AA' перпендикулярна a й в точці перетину прямої a ділиться навпіл. Аналогічно AA'' точкою перетину прямої видання ділиться навпіл, і пряма видання перпендикулярна AA'' . Отже, $OA=OA'=OA''$. Кут ϕ ділиться прямою OA' на дві частини, одна з яких дорівнює половині кута AOA' , а інша дорівнює половині кута $A'O A''$. Отже, кут AOA' дорівнює 2ϕ , так як кут AOA'' дорівнює сумі кутів AOA' і $A'O A''$. Це означає, що точка A'' є образом точки A при повороті площини на кут 2ϕ , що й треба було довести.

Найбільш важливі афінніє перетворення - це стиснення площині до прямої, зрушення, гомотетия, симетрія. Таким чином, всі перетворення, що вивчаються в елементарній геометрії середньої школи, є окремими випадками афінних перетворень.

При афінному перетворенні площині еліпс переходить в еліпс (зокрема, в коло); тому еліпс і коло - афінно-родинні фігури (або афінно рівні). Парабола

при афінному перетворенні переходить в будь-яку іншу параболу, гіпербола - в гіперболу. Тому всі еліпси складають один афінний тип (клас) кривих другого порядку, параболи - інший афінний тип і гіперболи- третій тип кривих. При афінному перетворенні крива одного афінного типу не може перейти в криву іншого афінного типу.

Афінний перетворення часто використовується як метод вирішення геометричних задач; при цьому дану фігуру афінно перетворюють в більш просту, де знаходять вказану властивість, а потім виконують зворотнє Афінний перетворення і знаходять шукану властивість даної фігури.

2.2 Алгоритми перетворення зображень на основі технології морфінгу

Морфінг (morphing) - технологія комп'ютерної графіки, що створює плавний перехідний ряд зображень від одного об'єкта до іншого. Сам термін походить від слова metamorphosing - проведення перетворення, в якому один образ поступово перетворюється в інший. Даний метод вперше був використаний в 1990 році для створення спец-ефектів в кіноіндустрії. З тих пір Морфірованіє зображень набуло широкого застосування в різних областях діяльності людини. Існує кілька різновидів морфінга, в тому числі лінійний, сегментна і зважений [18].

Лінійний морфінг. Процедура лінійного морфінга дозволяє з пари вихідних $2D$ зображень A і B побудувати третю, що володіє в заданій ступеня властивостями кожного з вихідних зображень. Для вирішення цього завдання на вихідних зображеннях визначається набір взаємно відповідних ключових точок, що утворюють триангуляційну сітку. У разі коли вихідні зображення є особи, ключові точки розставляються відповідно до їх анатомічним пристроєм, наприклад, відповідно до розташування очей, брів, рота, носа, вух тощо.

На основі побудованої сітки розраховується відображення, що трансформує кожен клітинку сітки зображення A в відповідному полі сітки зображення B і таким чином перебудовується зображення A в зображення B , тобто кожній точці зображення $A(x_1, y_1)$ можна поставити у відповідність точку зображення $B(x_2, y_2)$. За допомогою рівнянь, що описують процес трансформації, розраховується позиція і яскравість точок проміжного зображення, відповідних тій або іншій мірі завершеності процесу. Якщо, наприклад, процес трансформації проведений на 30%, то проміжне зображення буде являти собою морф, що складається з 70% A і 30% B . В результаті при досить детальному наборі ключових точок процедура морфінга дозволяє побудувати плавний перехідний ряд заданої довжини між A і B , характеризується постійною ступенем візуальною відмінності між сусідніми зображеннями.

У той же час дана процедура має цілу низку додаткових, які представляють певний інтерес можливостей, як, наприклад, застосування техніки нелінійних сіток, коли ключові точки замість відрізків прямих з'єднуються кривими Безьє, що в результаті дозволить істотно поліпшити якість морфінга при одночасному скороченні числа необхідних ключових точок .

Зважений морфінг дозволяє трансформувати одну базову модель відразу в дві морфінг-мішені. Головна перевага цієї технології полягає в можливості здійснення синхронізації руху губ натурника як з його промовою, так і з мімікою обличчя в цілому. Розглянемо принципову відмінність зваженого морфінга від інших його видів. При морфінгу базового об'єкта A на 50% за допомогою цільового об'єкта - плаче рот (з виваженим коефіцієнтом 20%) виходить верхня модель колонки B (рисунок 2.9). При обробці цього ж базового об'єкта ще на 50%, але з іншим цільовим об'єктом - сміється рот (з коефіцієнтом 70%) - виходить нижня модель стовпчика B .

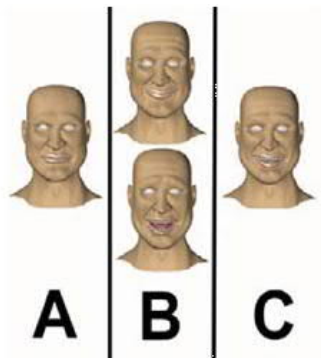


Рисунок 2.9 – Приклад зваженого морфінгу

Інструментарій зваженого морфінга дозволяє для кожного цільового об'єкта встановлювати свої вагові коефіцієнти або відсоток морфінга, створюючи таким чином безліч варіантів мішеней на основі декількох цільових об'єктів. Результат суміщення обох «портретів» представлено в колонці С.

Якщо потрібне створення будь-яких інших варіантів зображення того ж особи, необхідно лише змінити процентне співвідношення «вкладів» різних морфінг-мішеней в остаточне зображення: 20/80, 60/40 тощо. Сума відсотків не повинна перевищувати 100, хоча у багатьох програмах дана умова можна порушити і отримати досить цікаві результати. Таким чином, зважений морфінг дозволяє створювати з одного базового об'єкта кілька змінених з подальшим створенням необхідного виразу обличчя.

Варпінг зображень. Варпінг(Warping) - технологія варпінга аналогічна сегментному морфінгу і в якомусь сенсі тотожна останньому, але відрізняється тим, що не пов'язана жорстко з набором цільових об'єктів (рот, очі, ніс тощо). При використанні варпінга, так само як і при морфінгу, оператор (дослідник) формує лише опорні фігури або задає ключові точки, які дозволяють комп'ютеру виконати правильний перехід. Використовуючи варпінг (варп-морфінг), можна заданим чином змінити вихідне зображення, наприклад, накласти на обличчя не тільки пряму (сегментний морфінг), але і криву посмішку, розширити або звузити очі, змінити внутрішню структуру частин особи незвичним способом і технологія варпінга дає можливість створювати оригінальний стомлений матеріал, не вдаючись до копіткою пошуків

відповідних моделей, наприклад, моделювати особи з певними ознаками або конструювати химерні особи або незвичайні композитні особи.

У методиці, спрямованої на вивчення перетворень станів вигляду обличчя людини, з використанням технології варпінга трансформувалися енцефалометричні індекси внутрішньої структури особи: ширина очей, висота очей або величина чола, довжина носа, висота рота або величина підборіддя, які і виступили в ролі основних варійованих змінних. Згідно з дослідженням, графічна схема, максимально відповідна експресії радості, характеризується високим чолом, широким розташуванням очей, середнім розташуванням верхнього краю носа і укороченим його кінчиком, верхнім розташуванням рота.

Прототипування (prototyping) - технологія комп'ютерної графіки, що створює зображення з усередненими характеристиками з безлічі заданих об'єктів. Підсумкове зображення загального або усередненого типу особи виступає в якості прототипу тієї чи іншої групи. Такими групами можуть бути, наприклад, всі чоловіки, або чоловіки однієї сім'ї, або представники різних етносів, або прототип однієї людини, виділений з різних знімків його особи, зроблених ще і в різні роки. Одним з перших, хто розробляв ідею прототипу, був Френсіс Гальтон. У другій половині XIX століття він робив фотознімки і накладав їх один на одного, щоб отримати усереднений тип особи.

Використання різних видів прототипів дозволяє вивчати відмінності середніх характеристик загальних груп, наприклад, чому саме відрізняються один від одного особи чоловіків і жінок.

Можливості комп'ютерної графіки також дозволяють, зберігаючи індивідуальні властивості особи натурника, змінювати різні його характеристики, такі, як вік, привабливість, маскуліність-фемінність, «дитячість особи» і багато інших.

Карикатурування (Caricaturing) - це вид перетворення, в якому перебільшуються деякі риси обличчя, щоб підкреслити характерні для цієї людини відмітні ознаки і таким чином зробити його обличчя повністю не схожим на інші. Вперше комп'ютерну технологію створення карикатур

застосувала Сьюзан Бреннан. Та ж технологія застосовується і для створення прототипические осіб. На фотозображення наноситься набір ключових точок, промальовує основні структури особи. Ці точки з'єднують, і утворюється форма особи, що складається з ліній. Якщо зняти такий же набір точок з інших осіб, а потім все їх з'єднати, вийде лінійна форма усередненого типу особи. Щоб створити карикатуру, необхідно спотворити і перебільшити розташування нетипових структур особи. Експерименти з карикатурними особами кінозірок показали, що карикатури розпізнаються швидше, ніж нормальні особи. Крім того, карикатурному особі надається перевага як найбільш схожим на його володаря при виборі з двох альтернатив - карикатурного і природного. Приклад спотворення деяких елементів зображення наведено на рисунку 2.10.



Рисунок 2.10 – Приклад перетворення деяких елементів зображення

Якщо карикатури легко розпізнаються, то антикаркатури, навпаки, менш впізнавані. Створюючи антикаркатури особи, дослідник наближає точки, що відповідають за його особливість, до точок прототипу особи. Можливе створення зображень антиобличчя, тобто особи, яка має протилежні особливості по відношенню до початкового. Товсті губи перетворюються в тонкі, маленькі очі - в великі тощо. Так само можна конструювати «місячне» обличчя і робити багато інших перетворення.

Модель морфіруемого синтезу тривимірних зображень. В останні 15 років спостерігається істотний прогрес у створенні інтерактивних тривимірних зображень. У сучасній комп'ютерній графіці тривимірне зображення розглядається як комбінація об'ємної полігональної сітки і натягнутою на неї

текстури, яка описує колір зображення. Це означає, що техніки морфинга і особливо варпінга природним чином застосовні для створення тривимірних зображень. Використання досягнень в розробках тривимірної графіки є перспективний напрямок в психологічних дослідженнях людини по виразу його обличчя.

На рисунку 2.11 показана процедура тривимірного варпінга особи голлівудської кінозірки Тома Хенкса за допомогою комп'ютерної моделі морфіруемого синтезу тривимірних зображень особи. Вона дозволяє з простого двовимірного зображення особи (1) створити його тривимірних форму (2) і текстуру (3). Форма і текстура промальовувалися на базі даних тривимірних осіб, створеної шляхом тривимірного лазерного сканування, яке здійснюється наступним чином: людина, особа якого скануватиметься, сідає на стілець, що обертається, який знаходиться перед пучком лазерного світла. У лазерному світловому потоці відображаються всі подробиці форми і текстури особи людини і записуються таким чином, що особа постає у вигляді великого числа лазерних зліпків-профілів. Опорні точки сусідніх профілів з'єднують разом з утворенням каркасної моделі з полігональної сіткою, яка графічно являє собою об'ємні криволінійні поверхні голови і обличчя людини. Кожній точці присвоюється коефіцієнт, величина якого визначає ступінь її впливу на частину поверхні, що проходить поблизу точки. Від взаємного розташування крапок і величини коефіцієнтів залежить форма і «гладкість» поверхні в цілому. Конструюються особа утримує набір таких модельних коефіцієнтів, на основі яких шляхом поточечного зіставлення в векторному просторі проб форм і текстур створюються нові природно сприймаються особи. Модель особи, екстрагований з двовимірне зображення, можна обертати або перетворювати в необхідному ракурсі. Так, особа Т. Хенкса трансформувалося в повне (4) і худе (5), а також набувало похмуре вираз (6) і веселе (7). Кожній точці присвоюється коефіцієнт, величина якого визначає ступінь її впливу на частину поверхні, що проходить поблизу точки. Від взаємного розташування крапок і величини коефіцієнтів залежить форма і «гладкість» поверхні в цілому. Конструюються

особа утримує набір таких модельних коефіцієнтів, на основі яких шляхом поточечного зіставлення в векторному просторі проб форм і текстур створюються нові природно сприймаються особи.

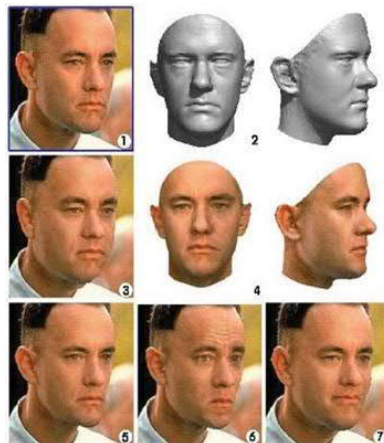


Рисунок 2.11 - Процедура перетворення в моделі морфуючого синтезу тривимірного зображення особи, де 1 - вихідне зображення, 2 - трьохвимірний форма особи, 3 - текстура особи, 4-7 - перетворене особа (4 - повне, 5 - худе, 6 - похмуро, 7 - усміхнене)

Основну частку ринку програмних засобів обробки тривимірної графіки займають три пакети: 3D Studio Max (компанії Kinetix), Softimage 3D (компанії Microsoft) і Maya (консорціум відомих компаній Alias, Wavefront, TD1). На сьогоднішній день Maya є найбільш передовим пакетом в класі засобів створення і обробки тривимірної графіки для персональних комп'ютерів. Що стосується програм для роботи з двовимірними зображеннями, то їх представлено трохи більше десятка. Найвідоміші з них: Abrosoft Fantamorph, FunMorph, MorphMan, WinMorph, Morph Buster, Magic Morph, Morpheus, Morpher тощо.

Використання тривимірних зображень голови людини в психологічних дослідженнях вкрай перспективно, але стримується низкою факторів. Техніка, що дозволяє здійснювати безпосереднє тривимірне сканування, вкрай дорога, а процедури реконструкції тривимірних моделей голови по фотозображення

анфас і в профіль не дозволяють досягти портретної схожості з моделлю. Можна припустити, що подальший прогрес в області вивчення сприйняття виразів обличчя буде пов'язаний з використанням анатомічно коректних тривимірних моделей голови людини, що включають опис вісцерального черепа, облягаючих його м'язових і шкірних покривів.

Розглянуті методи комп'ютерної графіки значно полегшують вивчення психологічних особливостей і станів людини по виразу його обличчя. Моделювання необхідних зображень дозволяє виявляти нові закономірності процесу міжособистісного сприйняття. З практичної точки зору розвиток комп'ютерної графіки стимулює появу більш адекватних і зручних методів реконструкції особи, які використовуються, наприклад, в криміналістиці, а також володіють можливістю маніпуляцій його частинами, змін співвідношень між ними, масштабних уточнень з метою створення більш реалістичного зображення особи.

Використання технології морфінгу та застосування програмно-апаратних комплексів для проведення обробки цифрових зображень надає нові можливості для створення принципово нових технологій та отримання нових результатів. Тому даний напрямок розвитку комп'ютерної графіки має великі перспективи для застосування в різних галузях життя.

2.3 Алгоритм перетворення елементів цифрового зображення

Провівши аналіз відомих алгоритмів та виділивши їх основні переваги та недоліки було спроектовано алгоритм перетворення зображень на основі технології морфінгу. Запропонований алгоритм ґрунтується на розділенні вхідного зображення на окремі області за допомогою накладання сітки. З подальшим процесом перетворення окремих сегментів за допомогою афінних перетворень. Блок-схема алгоритму наведена на рисунку 2.12.



Рисунок 2.12 – Блок-схема алгоритму перетворення елементів на зображенні

Серед особливостей запропонованого алгоритму слід відмітити поєднання алгоритмі попередньої обробки, фільтрації, афінних перетворень та правил взаємодії між характерними точками на різних етапах роботи.

Запропонований алгоритм ґрунтується на твердженні, що кожне зображення може бути вписане в прямокутну область на яку в подальшому буде накладено поверхневу сітку з характерним точками. Дані точки будуть мати відповідників на іншому зображенні. В результаті процесу перетворення

через задану кількість кроків всі сегменти першого зображення передуй в зображення номер два.

Розмірність сітки, що буде накладена на зображення та кількість проміжних перетворень обирається від того:

- наскільки велике зображення;
- наявні апаратні ресурси для проведення математичних обчислень;
- плавність переходу обного зображення в інше.

Даний набір правил достатній для того щоб обрати всі параметри для роботи алгоритму. Для уточнення параметрів роботи алгоритму достатньо провести декілька додаткових запусків алгоритму з різними параметрами.

Запропонований алгоритм складається з такої послідовності кроків.

Крок 1. Завантаження вхідного зображення.

Крок 2. Проведення попередньої обробки зображення. На даному етапі проводиться перетворення вхідного зображення з кольорового у градації сірого. Проводиться видалення сторонніх шумів тощо.

Крок 3. Встановлення розмірів $N \times M$ сітки. Початкові розміри задаються з розрахунку 1 точка на 25 пікселів зображення, дані параметри були встановлено на основі проведених експериментів, як такі що оптимальними по параметру математичні операції/ якість;

Крок 4. Встановлення точок відповідності на двох зображеннях. Даний крок носить важливий характер та проводиться користувачем в ручному режимі від результату його проведення залежить результат роботи програми вцілому. Відповідні точки повинні відповідати тим областям зображення в яке має перейти область з першого зображення в область з другого зображення.

Крок 5. Послідовна порівняння обриманих сегментів зображення, якщо сегменти подіні між собою, то такі сегменти відкидаються з подальшої обробки.

Крок 6. Якщо відповідні сегменти були визнані такими, що не є подібними за параметрами площі (первинний параметр) або форми (вторинний

параметр), то проводиться процедура розрахунку та визначення K проміжних відображень сегмента.

Крок 7. Якщо всі сегменти пройшли процес перетворення переходим на крок 8, в іншому випадку, в іншому випадку крок 5.

Крок 8. Вивід результатів перетворення у вигляді послідовних проміжних фреймів з заданою частотою.

Дана послідовність кроків забезпечує повноцінний аналіз вхідного зображення та дозволяє виділити відповідні точки та розити зображення на відповідні сегменти, що в кінцевому варіанті дозволить провести процедуру передворення зображення на основі технології морфінгу.

До переваг даного алгоритму слід віднести його простоту та швидкість роботи. До недоліків: необхідність проведення процесу налаштування програми перед її роботою, оскільки для коректного та швидкого функціонування необхідно встановити початкові параметри роботи алгоритму (розміри сітки, відповідні точки).

2.5 Висновки до розділу

Проведено аналіз алгоритмів перетворення зображень за допомогою афінних перетворень, що дало можливість обрати математичне обґрунтування для проведення перетворень окремих елементів цифрового зображення.

Розроблено алгоритм перетворення елементів цифрових зображень на основі технології морфінгу, що дозволило спроектувати програмний додаток для обробки цифрових зображень.

3 ПРОГРАМНА СИСТЕМА АНАЛІЗУ ТА ПЕРЕТВОРЕННЯ ЦИФРОВИХ ЗОБРАЖЕНЬ

3.1 Структура програмної аналізу та перетворення зображень

Обробка зображень - це процес маніпулювання піксельними даними, щоб зробити його придатним для програм комп'ютерного зору або зробити його придатним для подання їх людям. Наприклад, виділення тих особливостей на зображенні. Що будуть необхідні для провдення того чи іншого етапу дослідження чи подільшого аналізу.

На сьогоднішній день на ринку присутня велика кількість програмних бібліотек для обробки зображень, що містять велику кількість функцій необхідних для провдення процедури обробки на різних стаціях його аналізу. Проте однією з найбільш розвинутих та використовуваних є бібліотека Halcon.

HALCON визначає сучасний рівень програмного забезпечення для машинного зору. Вона забезпечує всебічне розуміння та завжди базується на новітніх та найсучасніших технологіях. Незалежно від завдання, Halcon вирішить це, швидко і з найвищою точністю.

Набір функцій для створення програмних додатків з компютерної графіки містять професійний інструмент обробки зображень, що є не просто бібліотекою операторів обробки зображень. Вирішення завдань обробки зображень - це лише одна частина повного рішення, яке включає інше програмне забезпечення такі компоненти, як управління процесами або доступ до бази даних, та апаратні компоненти від освітлення до керування пристроями для збору зображень та багато інших механічних компонентів. Тому важливо, щоб система обробки зображень була проста у використанні і може бути гнучко інтегрована в цикл обробки даних.

Для цього в Halcon реалізовано багато функцій, що забезпечують про всі важливі аспекти:

– розробка програмного забезпечення підтримується IDE halcon (інтегроване середовище розробки), що складається з HDevelop та HDevEngine. HDevelop - це високоінтерактивний інструмент розробки, який дозволяє швидко розробити завдання обробки зображень. За допомогою HDevEngine ви можете безпосередньо виконати програми та процедури HDevelop із програми C ++, C #, Visual Basic або C. Крім того HDevelop також може експортувати програми та процедури на вашій мові програмування;

– проблемно-орієнтована документація охоплює всі рівні, від швидкого доступу до важливої інформації аж до детального обговорення передових тем;

– описи поєднуються з сотнями прикладів для інтуїтивного розуміння рішення, які можуть служити шаблонами для скорочення часу розробки;

– бібліотека halcon пропонує відкриті інтерфейси для ефективного обміну даними для інтеграції власних операторів, або для доступу до спеціалізованого обладнання, що завершує систему.

Halcon відповідає всім вимогам професійної бібліотеки для створення програмних додатків з функцією компютерного зору:

– вона включає методи для всіх стандартних та вдосконалених типів обробки зображень, починаючи від отримання зображень з багатьох різних пристроїв і до вдосконаленого відповідності фігур.

– окрім функціональних можливостей обробки зображень, Halcon пропонує інструменти, які зазвичай потрібні в контексті програм машинного зору, наприклад, для зв'язку через сокети або послідовний інтерфейс, обробка файлів, аналіз даних, арифметичні операції або класифікація;

– Halcon пропонує гнучкі способи розпаралелювання для використання багатопроцесорного або багатоядерного обладнання для прискорення програми;

– бібліотека Halcon, яка використовується в додатку, не буде видимою для кінцевого користувача і вимагає лише мінімальні ресурси в установці, що

робить її ідеальною для розробок програмних додатків на персональних комп'ютерах.

При проектуванні структурної архітектури програмної системи був використаний ієрархічно-модульний підхід, оскільки даний підхід надає можливість максимально швидко проводити маніпуляції з окремими модулями системи: додавання, корегування, заміни та виділення окремих структурних елементів програмної системи без втрати її працездатності в цілому.

Спрощену структуру програмної системи наведено на рисунку 3.1.

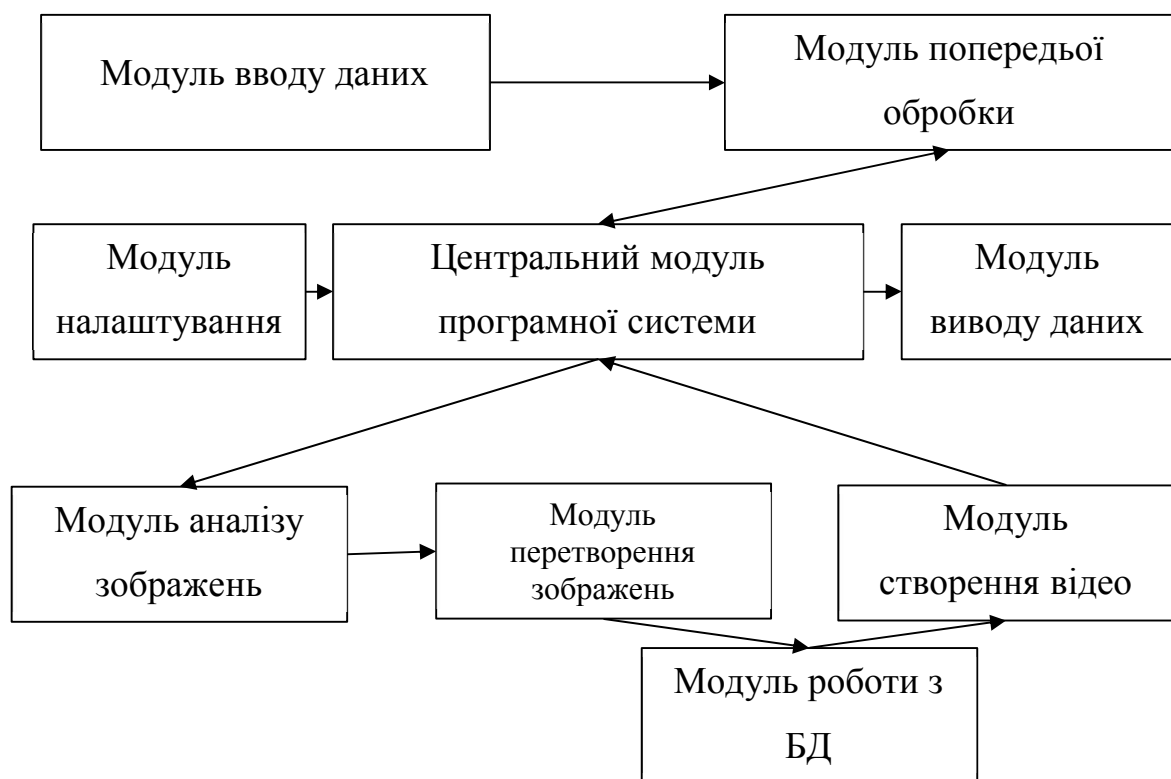


Рисунок 3.1 – Узагальнена структура програмного додатка аналізу та перетворення зображень

Запропонована структура є включає усі стандартні програмні модуля, які характерні для програмних розробок даного класу. Серед основних модулів слід відмітити такі:

Центральний модуль програмної системи – даний програмний модуль забезпечує взаємодію між окремими структурними елементами реалізованої

програмної системи, до його функцій відносяться функції перевірки цілісності програмного додатку, наявність оновлень чи відключення пошкоджених блоків програми, контроль та переконвертація даних під час передачі інформації між окремими мобілями системи. Вхідними даними даного модуля є вектор параметрів роботи програми. Вихідними – корекційні правки пакетів обміну даних, повідомлення для користувача при необхідності внести тих чи інших змін в роботу програмного додатку в цілому.

Модуль вводу даних – даний модуль має низький пріоритет надійності, оскільки містить функції для отримання вхідних зображень з різних пристроїв, серед яких зовнішні та внутрішні носії даних, фотоапаратура, бази даних цифрових зображень. На даному етапі робота програмного додатку не проводиться якась додаткова перевірка чи обробка інформації. Результат роботи даного модуля є пара вхідних зображень у форматі, що підтримується програмною системою та команда активації для всіх інших програмних модулів системи.

Модуль попередньої обробки – в даному модулі зібрані функції, що забезпечують підвищення якості подальшої обробки зображень на більш пізніх етапах аналізу та перетворення зображень. Більшість функцій були взяті з зовнішніх цифрових бібліотек, оскільки під час виконання індивідуального завдання перед виконавцем не ставились завдання розробки та проектування алгоритмів попередньої обробки зображень. Даний модуль зокрема виконує попередню перевірку чи формат та кодування вхідних зображень можуть бути обпрацьовані засобами розробленого програмного засобу, проведені попередній аналіз по визначенню розміру типу кольору, перевірка чи дані зображення не є ідентичними та чи в програмі є попередньо збережена сітка характерних точок для даної пари зображень. Результатом роботи даного модуля є встановлення параметрів початкової роботи програмного додатку.

Модуль налаштування – невеликий модуль функції якого забезпечують можливість налаштування програмної системи під потребу користувача, встановлення роботи самої системи та окремих її алгоритмів. Серед параметрів,

які можна встановити слід відмітити: тип та розмір шрифтів інтерфейсу та модулі виводу звітів, папарметрів підключення та роботи фото/веб/відеокамер, встановлення параметрів роботи алгоритмів аналізу зображень (розміри ковзного вікна та крок його зміни).

Модуль аналізу зображень – цей програмний модуль розроблений з метою проведення аналізу вхідної пари зображень, а саме встановлення розмітаної сітки, та визначення координат окремих її вузлів. Корекція окремих розташування окремих вузлів реалізована у ручному режимі та надає користувачеві можливість корегувати результати роботи програмного додатку на етапі підготовки роботи програми та встановлення параметрів процесу морфінгу. Оскільки програмно не завжди можливо в автоматичному режимі визначити відповідні області на вхідних зображеннях (в деяких випадках користувачі навмисне хочуть отримати спотворене відображення) тому саме таке вирішення даної задачі є максимально ефективним та дозволяє користувачеві у повній мірі отримати результат в залежності його потреб. Результатом роботи даного модуля є розмічена сітка з точками відповідності на двох зображеннях.

Модуль перетворення зображення – в даному модулі зібрані функції, що виконують основне завдання по виконанню задач перетворення зображень. А саме в даному модулі відбувається перевірка подібності відповідних сегментів зображення за двома критеріями подібності: площею (оскільки дану характеристику можна достатньо просто обчислити) та формою сегменту. Форма сегмента порівнюється як взаєморозташування сусідніх відрізків, які утворюють чотирикутник сегменту та кутів між ними. Другий критерій використовується якщо по першому критерію сегменти відрізняються менше ніж на 5%. Якщо і другий критерій не підтверджує подібність обраної пари відповідних сегментів, то над даною парою сегментів проводиться процедура перетворення, а саме створення проміжних відображень сегменту першого зображення, яке переходить в друге. Кількість проміжних відображень обирається користувачем в ручному режимі. Якщо ж сегменти були визначені

за двома критеріями як подібні, то вони визнають такі ми що не потребують перетворення і в кінцевому варіанті використовується сегмент з другого вхідного зображення.

Модуль роботи з БД – допоміжний модуль, що забезпечує можливість роботи з базою даних проміжних фреймів, та проводити зберігання тимчасової інформації (віділиних областей на зображенні), з метою подальшої обробки.

Модуль створення відео – допоміжний модуль програмної системи, що повинен забезпечити об'єднання усіх проміжних сегментів у відповідні цілісні фрейми з подальшим формуванням відпоередності. Відеопослідовність формується на основі встановлених користувачем параметрів зміни окремих кадрів та може коригуватись під час відображення результатів роботи програми.

Модуль виводу даних – призначений для відображення результатів роботи програми у вигляді відеофайлу, на якому відбувається послідовне перетворення першого вхідного зображення в друге, а потім навпаки. При цьому створюється відеофайл який демонструє послідовне перетворення одного зображення в інше в безперервному вигляді.

Спроектвана та розроблена ієрархічно-модульна структура програмного додатку аналізу та перетворення зображень у мовній мірі формує цілісну систему та має всі можливості для виконання поставлених завдань по отриманні, попередній обробці, аналізі, перетворенні та виводі результатів обробки вхідних зображень. Окрім того використаний модульний підхід дозволяє реагувати на зміни в роботі програмної системи додаючи навіть модулі чи модифікуючи вже присутні, при цьому роботоздатність та цілісність системи не буде пошкоджена.

Перед процесом реалізації розглянутих алгоритмів було проведено моделювання розробленого програмного додатку за допомогою сучасних програмних пакетів. Серед недоліків використання універсальної мови моделювання слід відмітити:

- часові затрати. Потрібно багато часу, щоб діаграма була розумною і синхронізована з фактичним кодом. Діаграми UML не запускаються, але

вимагають багато часу. Тому вони корисні тільки в тому випадку, якщо розробник може ними керувати.

– можливі втрати інформації. Не можливо представляти кожну умову в діаграмі послідовності. Тому діаграми стану повинні передавати основні факти, а не всі можливі результати.

– грошові витрати. Гарне програмне забезпечення UML коштує грошей, і для правильного освоєння потрібен якийсь час.

На першому кроці моделювання проведено аналіз набору функцій та можливих дій користувача при роботі з розробленою системою. Приклад діаграми прецедентів наведено на рисунку 3.2.



Рисунок 3.2 – Діаграма прецедентів програмної системи

Як видно з наведеної діаграми користувач має обмежені можливості для втручання в роботу самої програмної системи. Проте йому було надано можливість встановлення параметрів роботи програмної системи, а також він повноцінно може впливати на встановлення та налаштування вузлів сітки відповідностей, що в свою чергу підвищує якість результату роботи програмної системи. Заборона доступу користувача до внутрішніх параметрів роботи

системи підвищує надійність роботи програми та зменшує кількість хибних спрацювань від втручання в роботу непрофесійних користувачів або користувачів з низьким рівнем навиків роботи з програмними продуктами даного типу. Проте, достатньо високий рівень автоматизації роботи програмного додатку забезпечує можливість використання її для роботи та навчання користувачів без досвіду роботи. А сам процес опанування програмою є достатньо короткий та не вимагає від нових користувачів глибоких попередніх знань комп'ютерної техніки.

На другому кроці моделювання було проведено аналіз послідовності взаємодії між основними групами процесів та об'єктів, що будуть відбуватись в процесі роботи програми. Результат моделювання наведено на рисунку 3.3.

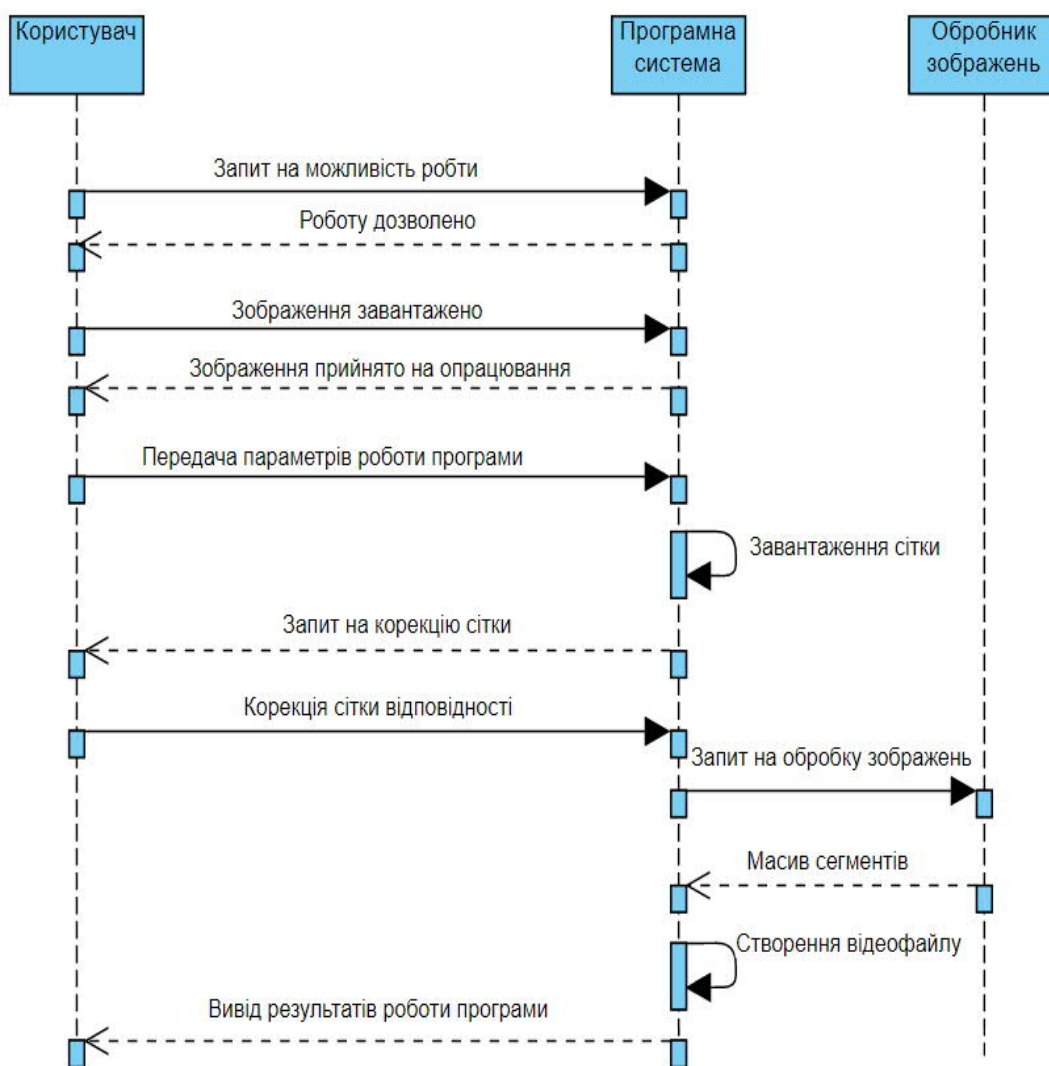


Рисунок 3.3 – Діаграма послідовності програмної системи

Як видно з наведеної діаграми, процес функціонування програмного додатку має послідовний характер та виконує усі дії послідовно, що з однієї сторони сповільнює процес опрацювання зображення, проте запропонований алгоритм дозволяє при наявності відповідних апаратних засобів провести процес розпаралелення задачі опрацювання зображення та прискорити процедуру опрацювання зображення. Проте в даній реалізації для використання програмного додатку для його роботи буде достатньо робочої станції з середніми технічними показниками

На етапі моделювання було розглянуто основні фактори. Які можуть впливати на роботу системи також функції, що має виконати користувач для досягнення результату.

Іншим етапом проектування та розробки програмної системи є етап розробки користувацького інтерфейсу програмної системи. Під час проектування дизайну головного вікна програми були враховані усі можливості для користувачів до швидкого доступу до окремих функцій та пунктів меню. А з іншої сторони вдалось уникнути значного завантаження головного вікна програми. При цьому основний простір на головному вікні було віддано областям для відображення вхідних зображень та область для відображення результатів роботи програми. Результат проектування графічного інтерфейсу користувача наведено на рисунку 3.4.

При проектуванні зовнішнього інтерфейсу було враховано те, що програмна система повинна працювати з мінімальним втручанням ззовні, через те для користувача доступна мінімальна кількість активних функціональних елементів. Основне функції, що необхідно виконувати в процесі налаштування роботи програми реалізовані у вигляді інтерактивних точок, що можна переміщувати по зображенні в ручному режимі з метою отримати сітку відповідності з максимальною точністю відображення відповідних точок. Мінімізація кількості активних компонентів на екрані дозволила збільшити корисну площу на головному вікні програми, що в свою чергу позитивно вплинуло на зручність при роботі з програмним додатком.

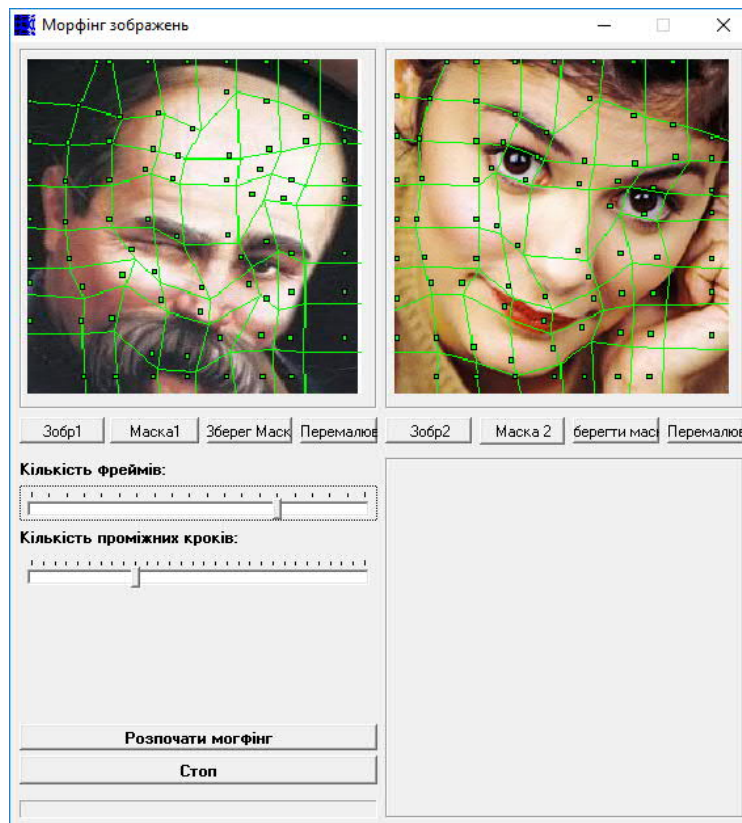


Рисунок 3.4 – Зовнішній вигляд програмної системи

Сам інтерфейс складається з трьох складивих елементів:

- головного вікна програмної системи для візуалізації вхідних зображень. Дана область також забезпечує можливість користувача переглядати поточні сітку відповідностей та проводити корегування її вузлів в залежності від оцікуваних результатів та поставлених задач. ;

- вікно відображення результатів роботи програми (результати візуалізуються у вигляді послідовності зображень, що змінюються з певною частотою, утворюючи при цьому неперервний відео-шоу) Користувач може зупинити відображення результатів активувавши відповідний пункт меню;

- панель активних перемикачів (набір перемикачів та активаторів функції програмної системи, необхідних для запуску, зупинки або корекції роботи програмної системи). Серед основних перемикачів слід відмітити кнопки: «Зобр1» яка активує функцію завантаження першого вхідного зображення, «Маска1» - для завантаження маски відповідності для першого

зображення, якщо її було попередньо збережено за допомогою кнопки «Зберегти маску», а також кнопка «Перемалювати» яка дозволяє нанести нову сітку на вхідне зображення та активувати процес налаштування даної сітки по новій. Аналогічний набір кнопок присутній і для другого вхідного зображення. Окрім того користувачу надається модифікація встановлення в ручному режимі кількості проміжних фреймів, що буде створено під час процесу перетворення та швидкість зміни фреймів під час відображення результатів роботи.

Даний зовнішній вигляд програмної системи достатньо зручний та швидкий в опануванні, що є безперечною перевагою під час роботи з програмним продуктом.

Для роботи з програмою користувачеві необхідно виконати просту послідовність кроків, візуально відображену на рисунку 3.5:

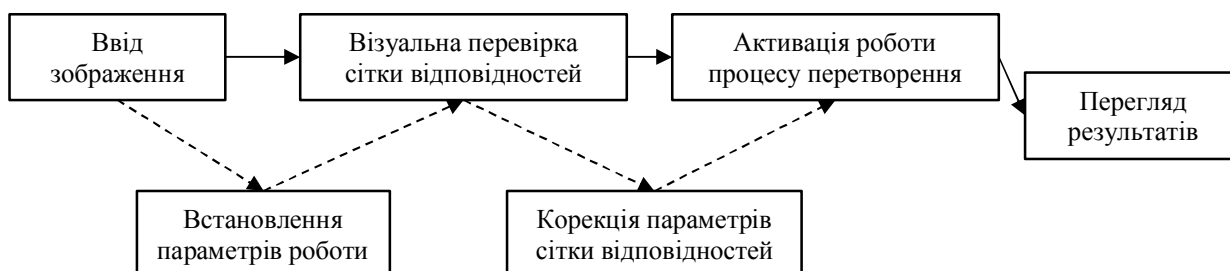


Рисунок 3.5 – Послідовність дій користувача при роботі з програмною системою перетворення елементів зображення

Як показано зі схеми дій користувача, при роботі користувач повинен виконати три основні дії з програмною системою, що включають активацію програмного додатку шляхом завантаження вхідних зображень, перевірки коректності сітки відповідності між двома зображеннями та перегляду отриманих результатів в процесі перетворення зображень. Окрім того користувач може виконати ще дві додаткові дії, при необхідності внесення змін при негативному або не достатньо хорошому результаті роботи програми.

3.2 Програмні модулі системи перетворення елементів зображення

Для реалізації програмного додатку було використано технічні можливості програмного середовища Delphi. За допомогою технологій обробки зображень даної мови програмування можна написати програмне забезпечення для керування яскравістю зображення, контрастом, знаходженням крайової характеристики, надання ефекту тиснення або змінити зображення на вигляд олійного живопису. Обробка зображень займається маніпуляцією кольором, піксельною операцією, навіть кадровою операцією. Основні переваги даної мови це:

- операція над окремими пікселями - це операція, при якій кожен піксель у зображенні працює самостійно, тобто значення пікселя не впливає на інші значення пікселів. А залежить виключно від попереднього значення та параметрів роботи;

- глобальні операції, коли глобальна характеристика зображення використовується для вирахування параметра для роботи кожного пікселя. Загальна характеристика, як правило, визначається за допомогою статистичного методу;

- багатокadroва операція, коли зображення працює з іншими зображеннями, щоб отримати результат. Наприклад, це генерація перехідного зображення наступної картини уповільненого руху, що саме потрібно нам при візуалізації результатів роботи програмного додатку;

- геометричні операції, де форма, розмір чи орієнтація модифікуються. Прикладом є збільшення розміру та перетворення;

- операція над сусідніми пікселями, де результат роботи пікселя залежить не лише від попереднього значення, а й від інших значень оточуючих пікселів;

– морфологічні операції - це операції, орієнтовані на конкретну область зображення. Ця операція тісно пов'язана з аналізом зображень, оскільки вона має справу з алгоритмом виявлення об'єктів/морфів.

Аналіз можливостей даної системи розробки та проектування програмних додатків показав, що дана система в повній мірі підходить для вирішення поставлених завдань, оскільки підтримує реалізацію основних зачач, які необхідно вирішити під час виконанні індивідуального завдання (отримання, обробка як окремих пікселів так і областей зображення, перетворення частини зображення).

За сонову було обрано клас `Bitmap`. Даний клас ьнкапсулює точковий малюнок `GDI+`, що складається з даних пікселів графічного зображення і атрибутів малюнка. Об'єкт `Bitmap` використовується для роботи з зображеннями, які визначаються даними пікселів. Точковий малюнок складається з точкових даних для графічного зображення і його атрибутів. Існує безліч стандартних форматів для збереження растрового зображення в файл. `GDI+` підтримує такі формати файлів: `BMP`, `GIF`, `EXIF`, `JPG`, `PNG` і `TIFF`.

Розробник можете створювати зображення з файлів, потоків та інших джерел за допомогою одного з `Bitmap` конструкторів і зберігати їх в потік або в файлової системі за допомогою `Save` методу. Зображення малюються на екрані або в пам'ять за допомогою `DrawImage` методу `Graphics` об'єкта. Для завантаження зображення було використано стандартний метод даного класу:

```
procedure TForm1.LoadAClick(Sender: TObject);
begin
  opendialog1.Filter := '*.jpg|*.jpg';
  if OpenDialog1.Execute then begin
    if (MidStr((TSpeedButton(Sender)).Name,5,1) = 'A') then begin
      LoadJpg(OpenDialog1.FileName,OrigBmpA);
    end else begin
      LoadJpg(OpenDialog1.FileName,OrigBmpB);
    end;
  end;
  Point_lines(ImageA,OrigBmpA,1);
  Point_lines(ImageB,OrigBmpB,2);
end;
```

Серед основних властивостей класу слід зазначити такі (Таблиця 3.1):

Таблиця 3.1- Основні властивості класу Bitmap

Властивість	Значення параметру
Height	Повертає висоту об'єкта Image в пікселях.
HorizontalResolution	Повертає горизонтальне розширення об'єкта Image в пікселях на дюйм.
Palette	Задає палітру кольорів, яка використовується для об'єкта Image.
PhysicalDimension	Повертає ширину і висоту даного зображення.
PixelFormat	Повертає формат пікселів для цього об'єкта Image.
RawFormat	Повертає формат файлу цього об'єкта Image.
Size	Повертає ширину і висоту зображення в пікселях.
Tag	Повертає або задає об'єкт, що надає додаткові дані про зображення. (Успадкованих від Image)
VerticalResolution	Повертає вертикальне розширення об'єкта Image в пікселях на дюйм.
Width	Повертає ширину об'єкта Image в пікселях.

Використання даних параметрів класу в повній мірі дозволяє спростити процес попередньої обробки зображення, оскільки половину інформації про властивості вхідного зображення буде отримано під час його завантаження у відповідний екземпляр класу.

Після отримання зображення програмна система перейде на наступний етап своєї роботи та спробує накласти сітку відповідності на конкретне зображення (рисунок 3.6). Сітка реалізована як масив окремих точок, що будуть віртуально з'єднуватись між собою за допомогою ліній, створюючи при цьому видимість нерозривної сітки. При цьому змінюючи положення окремої вершини сітки, користувач може змінювати весь її зовнішній вигляд.

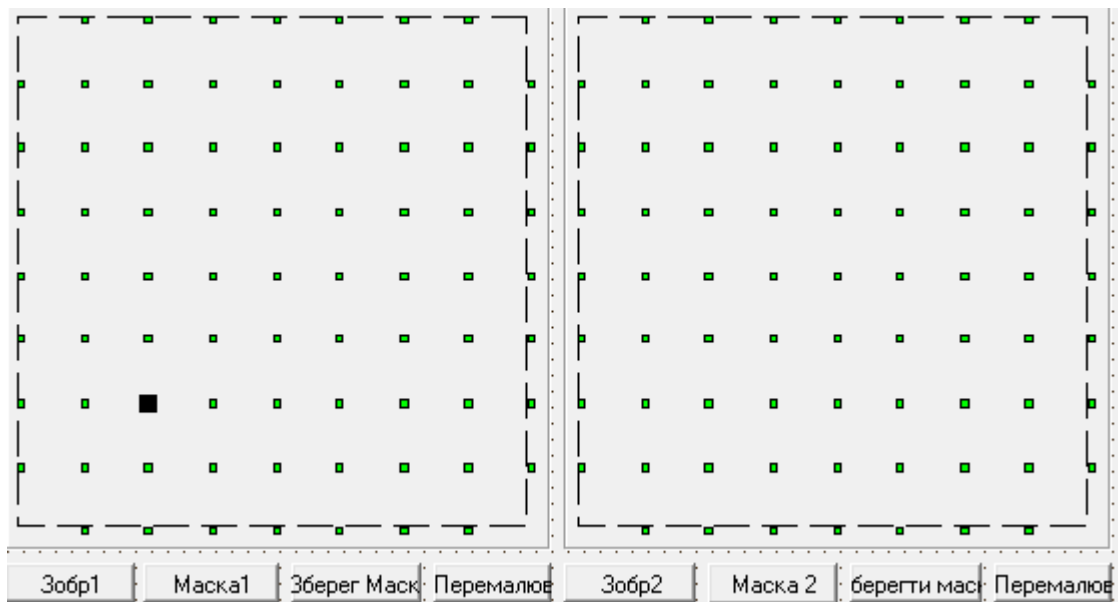


Рисунок 3.6 – реалізації сітки відповідностей за допомогою масиву відокремлених точок

В програмі реалізовано механізм при якому зміна місце розташування однієї з точок призводить до зміни форми сегментів зображення, що будуть утворені за її допомогою. При цьому сусідні точки не піддають переміщенню, а отже змінюють форму тільки чотири сегменти, які безпосередньо дотикаються до даної вершини.

```

procedure TForm1.ShapeAMove(Sender: TObject; Shift: TShiftState;
X,
Y: Integer);
var
  newX,newY : integer;
begin
  if (PointA_ismoving) then begin
    newX := (TShape(Sender)).Left + X;
    newY := (TShape(Sender)).top + Y;
    IsAllowToMove(newX,newY,PointA_cursel_AB);
    (TShape(Sender)).Left := newX;
    (TShape(Sender)).top := newY;

LesPointsA_x[PointA_cursel_x][PointA_cursel_y][PointA_cursel_AB]
:= newX;

LesPointsA_y[PointA_cursel_x][PointA_cursel_y][PointA_cursel_AB]
:= newY;
  end;end;

```

Для відображення отриманого результату скористаємось засобами, що нам надає програмне середовище. Для цього використаємо наступний програмний код:

```
procedure DrawPoly(x1,y1,x2,y2,x3,y3,x4,y4:integer);
var
  minX,minY,maxX,maxY : integer;
begin
  minX := x1;   minX := math.min(minX,x2);   minX :=
math.min(minX,x3); minX := math.min(minX,x4);
  minY := y1;   minY := math.min(minY,y2);   minY :=
math.min(minY,y3); minY := math.min(minY,y4);
  maxX := x1;   maxX := math.max(maxX,x2);   maxX :=
math.max(maxX,x3); maxX := math.max(maxX,x4);
  maxY := y1;   maxY := math.max(maxY,y2);   maxY :=
math.max(maxY,y3); maxY := math.max(maxY,y4);

  tab_bool_w := maxX-minX+1;
  tab_bool_h := maxY-minY+1;

  tmp_min_x := minX;
  tmp_min_y := minY;

  SetLength(tab_bool,tab_bool_w,tab_bool_h);
  tab_bool_erase;

  tab_bool_drawline(x1-minX,y1-minY,x2-minX,y2-minY);
  tab_bool_drawline(x2-minX,y2-minY,x3-minX,y3-minY);
  tab_bool_drawline(x3-minX,y3-minY,x4-minX,y4-minY);
  tab_bool_drawline(x4-minX,y4-minY,x1-minX,y1-minY);

  tab_bool_fill;
end;
```

В результаті виконання даного коду проводиться очислення всі вузли сітки відповідності будуть попарно з'єднані відрізками, при цьому буде враховуватись, які вузли є сусідніми. В результаті цього на зображенні буде відображатись граф, вузли якого з'єднані в певному порядку. Це з однієї сторони надасть користувачеві можливість оцінити форму майбутніх сегментів та внесе при потребі зміни для підвищення точності передачі інформації про окремі сектори. Даний підхід значно підвищить простоту роботи з програмою та візуалізує результати дій користувача.

Для створення візуального відео-відображення було використано принцип зміни окремих фреймів в наперед заданій послідовності за з деякою затримкою, дану реаліацію наведено наступному програмному коді:

```
procedure TForm1.Timer1Timer(Sender: TObject);
begin
  if frame_sens then begin
    frame_c := (frame_c+1);
    if (frame_c = frame_m-1) then frame_sens := not frame_sens;
  end else begin
    frame_c := (frame_c-1);
    if (frame_c = 0) then frame_sens := not frame_sens;
  end;

  image3.Picture.Bitmap.Assign(Lanim[frame_c]);
end;
```

Відображення масиву фреймів відбувається до того часу поки користувач на зупинить процес візуалізації активувавши відповідну кнопку в меню програмного додатку.

При встановленні інтервалу відображення зображень було одрано інтервал у розмірі 10 мс. Даний мінімальний крок зміни інтервалу дозволяє достатньо гнучко встановлювати параметри роботи програмного додатку.

```
procedure TForm1.TrackBar2Change(Sender: TObject);
begin
  timer1.Interval := trackbar2.Position * 10;
end;
```

Наведений перелік реалізованих функцій демонструє внутрішню будову програмного додатку та механізмів, що відбуваються під час його функціонування. Як показав процес написання програмного коду, обрано мова програмування та набір бібліотек у повній мірі забезпечив можливість створення програмного додатку для вирішення усіх завдань, що були поставлені при виконанні магістерської роботи.

3.3 Тестування та аналіз реалізованої системи

Для тестування програмної системи перетворення зображень було використано в якості робочої станції персональний комп'ютер з відповідним набором технічних характеристик:

- корпус Zalman Z1 Black + блок живлення Chieftec APS-550SB;
- HDD WD Caviar 500GB;
- відеокарта Asus PH-GTX1060-3G;
- ОЗУ 4Gb 2400GHz ;
- МП AMD Ryzen 3 2200G BOX 120;
- motherboard Asus B350M-E 90.

Технічні характеристики пристрою відображення зображень:

- діагональ монітора 19";
- роздільна здатність 1920x1080;
- тип матриці - TN;
- частота оновлення кадрів монітора 65 Гц;
- інтерфейси HDMI;
- відношення сторін 16: 9.

Технічні параметри комп'ютера є достатніми для проведення тестування розробленої програми та дозволить отримати результат для проведення аналізу коректності роботи програмної системи.

При проведенні тестування було обрано різні вхідні зображення, що відрізнялись між собою за такими критеріями:

- розмір зображення (обидва зображення приблизно одного розміру (рисунок 3.7а,б));
- розмір зображення (зображення значно відрізняються в розмірах одне від одного, більше ніж у два рази по висоті та ширині);

- вміст зображення (зображення містять приблизно однаковий вміст. Наприклад обличчя людей або подібні за формою квіти);
- вміст зображення (зображення містять кардинально різний вміст. Наприклад обличчя людей та елементи живої природи (рисунок 3.7в,г)).



а)



б)



в)



г)

Рисунок 3.8 – Приклад різноманітності вхідних зображень

В результаті аналізу можливих чинників впливу результати проведення тестування також було згруповано за декількома різними критеріями, для того щоб визначити параметри встановлення фотофіксуючої апаратури для максимально коректної роботи розробленої системи.

Для проведення першої групи тестів було відібрано зображення, що мало чим відрізняються один від одного, вони є приблизно однакові за розміром та

фрмою, на них зображені об'єкти подібної структури, сітки відповідності приблизно ідентичні (рисунок 3.9).

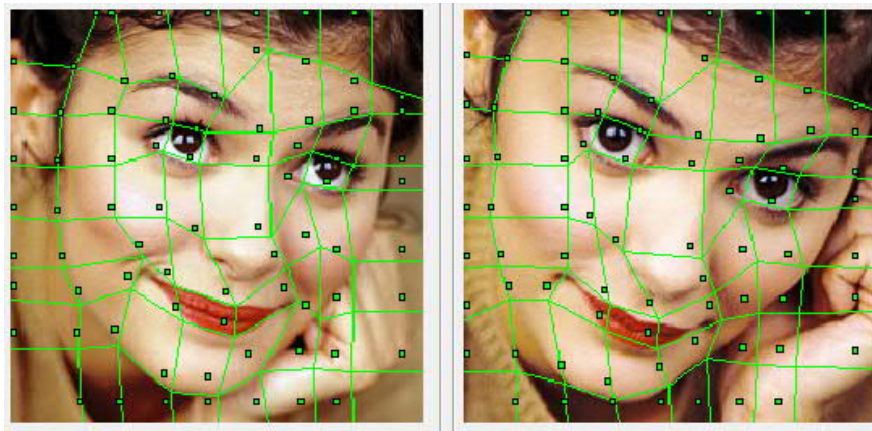


Рисунок 3.9 – Приклад вхідних зображень з мінімальними відмінностями

В результаті тестування даної групи було виявлено, що для проведення перетворення необхідно змінити мінімальну кількість сегментів, кількість фреймів для переборення таких зображень знаходиться в межах від 3 до 6, при цьому візуально не спостерігається якісь нетипові спотворення зображення під час трансформації. Кількість вузлів для коректної роботи програми – мінімальна, та знаходиться в межах 40-60 точок на один вузол.

Під час тестування другої групи зображень, в якості тестової вибірки були одрані зображення які значно відрізняються одне від одного розміром. Результати тестування показали, що якість проведення перетворення залищається візуально на високому рівні, проте для проведення тестування необхідно створити більше проміжних фреймів. Під час самопроцесу перетворення виникають явні неконтрольовані спотворення окремих сегментів, що приводять до виникнення негативних артефактів у відеоряді. Кількість вершин є неоднозначною, оскільки розміри зображень начно відрізняються. Для підвищення якості перетворення необхідно провести додаткове попереднє оброблення зображень, приведення до однакового розміру, з метою встановлення більш точнішої сітки відповідностей.

Третьою групою зображень для тестування було обрано зображення, що відрізняються за своїм змістом проте їх розміри та вміст подібні. Приклад такого порівняння наведено на рисунку 3.10.

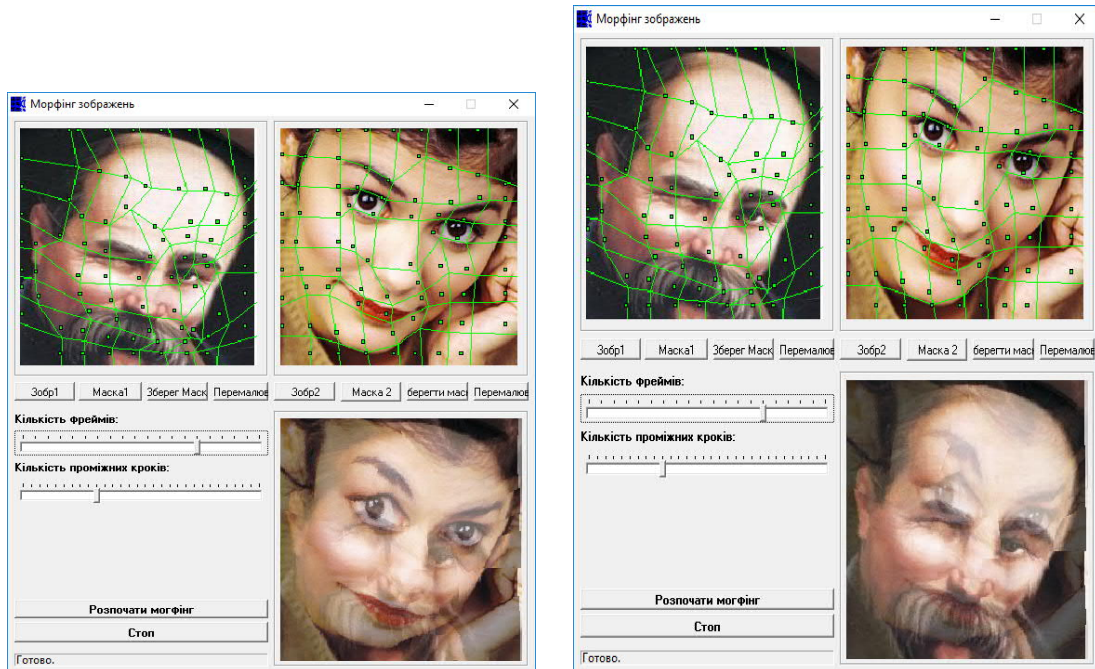


Рисунок 3.10 – Приклад морфінгу зображень

Як видно з наведених вище прикладів, програмна система успішно виконує аналіз та обробку вхідних зображень, проводить прорахунок проміжних фреймів та встановлює параметри роботи. Додатково надає можливість користувачеві проводити налаштування процесу перетворення в ручному режимі.

В результаті проведених експериментів та на основі отриманих результату можна сформулювати основні пари зображень які відрізняються між собою різними характеристиками, проте можуть бути попарно перетворені одне в одне. Отримані результати підтвердили припущення, які були зроблені на етапі модулювання, що запропонований алгоритм перетворення має високу швидкість, та пороводить обробку зображень з різними характеристиками, при цьому забезпечуючи відповідну якість результатів роботи. Результати досліджень наведено в таблиці 3.2.

Таблиця 3.2 – Узагальнена таблиця результатів тестування

Параметри вхідних зображень	Подібність зображень	Кількість фреймів, шт	Кількість точок на вузол	Візуальна оцінка перетворення
Розмір: однаковий Вміст: однотипний	дуже висока	4-6	50-75	Без спотворень
Розмір: різний Вміст: однотипний	висока	6-10	різна (впливає розмір)	Присутні спотворення деяких фреймів
Розмір: однаковий Вміст: різний	нормальна	5-8	25-50	Присутні мінімальні спотворення деяких фреймів
Розмір: різний Вміст: різний	мінімальна	20+	різна (впливає розмір)	Присутні видимі спотворення

В результаті проведених експериментів було виділено пари зображень для проведення процесу морфінгу, а також встановлено параметри автоматичного налаштування початкових характеристик роботи програмного додатку.

3.4 Висновки до розділу

Розроблено та проведено моделювання програмної системи перетворення зображень на основі алгоритму морфінга, що дозволило програмно реалізувати систему обробки зображень.

Проведено тестування розробленої програмної системи обробки та перетворення зображень, на основі різних тестових груп зображень, що дозволило виділити основні параметри зображень для попарного перетворення одне в одне.

ВИСНОВКИ

На основі аналізу сучасних програмних систем обробки цифрових зображень та аналізу алгоритмів перетворення зображень на основі морфінгу можна розбити наступні висновки:

1. Проведено дослідження та класифікацію сфер застосування комп'ютерної графіки, що дозволило виділити основні напрямки та алгоритми які використовуються перетворення цифрових зображень;

2. Проаналізовано алгоритми обробки цифрових зображень, що дозволило вибрати групу алгоритмів для проведення перетворень цифрових зображень, а також їх переваги та недоліки;

3. Проведено аналіз програмних систем обробки цифрових зображень який дозволив виділити основні структурні модулі, а встановити інтерфейси обміну даними між ними;

4. Проведено аналіз алгоритмів перетворення зображень за допомогою афінних перетворень, що дало можливість обрати математичне обґрунтування для проведення перетворень окремих елементів цифрового зображення;

5. Розроблено алгоритм перетворення елементів цифрових зображень на основі технології морфінгу, що дозволило спроектувати програмний додаток для обробки цифрових зображень;

6. Розроблено та проведено моделювання програмної системи перетворення зображень на основі алгоритму морфінга, що дозволило програмно реалізувати систему обробки зображень та провести її тестування.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Баженова И.Ю. Язык программирования С++ АО "Диалог-МИФИ", 2017. 366с.
2. Бартлетт Н. Программирование на Delphi Путеводитель. The Coriolis Group, Inc., 1996, Издательство НИПФ "ДиаСофт Лтд.", 2016. 116с.
3. Вебер Дж. Технология С++ в подлиннике. QUE Corporation, 2016, "ВНУ-Санкт-Петербург", 2017. 256с.
4. Волш А. И. Основы программирования на С++ для World Wide Web. IDG Books Worldwide, Inc., 1996, Издательство "Диалектика", 2016. 458с.
5. Марков А. С. «Базы данных. Введение в теорию и методологию. Финансы и статистика». 2016. С.24-35.
6. Абрамов С. А. Задачи по программированию. М.: Наука, 2018. 256с.
7. Березин Б.И., Начальный курс Delphi. М.: ДИАЛОГ-МИФИ, 2016. 331с.
8. Бондарев В.М. Основы программирования. Харьков: Фолио, Ростов н/Д: Феникс, 2017. 446с.
9. Вирт Н. Алгоритмы и структуры данных. М.: Мир, 2019. 345с.
10. Гладков В. П. Задачи по информатике на вступительном экзамене в вуз и их решения: Учебное пособие. Пермь: Перм. техн. ун-т, 2014. 516с.
11. Грогоно П. Программирование на языке Delphi. М.: Мир, 2012. 216с.
12. Дагене В.А. 100 задач по программированию. М.: Просвещение, 2013. 106с.
13. Джамса К. Библиотека программиста Java. Jamsa Press, 2016, ООО "Попурри", 2016. 656с.
14. Марков А. С. «Базы данных. Введение в теорию и методологию. Финансы и статистика». 2016. Р. 24-35.
15. Заварыкин В.М. Основы информатики и вычислительной техники. М.: Просвещение, 2019. 556с.

16. Касаткин В. Н. Информация. Алгоритмы. ЭВМ. М.: Просвещение, 2011. 219с.
17. Кен А. Язык программирования Delphi. Addison-Wesley Longman, U.S.A., 1996, Издательство "Питер-Пресс", 2017. 378с.
18. Керниган Б. Язык программирования Delphi. Пер. с англ. М.: Финансы и статистика, 2012. 391с.
19. Ляхович В.Ф. Руководство к решению задач по основам информатики и вычислительной техники. М.: Высшая школа, 2014. 127с.
20. Мейнджер Дж. Delphi Основы программирования. McGraw-Hill, Inc., 1996, Издательская группа ВНУ, Киев, 2017. 346с.
21. Миков А. И. Информатика. Введение в компьютерные науки. Пермь: Изд-во ПГУ, 2018. 442с.
22. Могилев А. В. Информатика: Учеб. пособие для студ. пед. Вузов. М.: Изд. центр «Академия», 2019. 629с.
23. Нотон П. JAVA: Справ. руководство. М.: БИНОМ: Восточ. Кн. Компания, 2016: Восточ. Кн. Компания. 447с.
24. Нотон П. Полный справочник по Java. McGraw-Hill, 1997, Издательство "Диалектика", 2017. 556с.
25. Ренеган Э. Дж. 1001 адрес WEB для программистов : Новейший путеводитель программиста по ресурсам World Wide Web: Пер.. Минск: Попурри, 2017. 512с. ил.
26. Родли Дж. Создание Java-апплетов. The Coriolis Group, Inc., 1996, Издательство НИПФ "ДиаСофт Лтд.", 2016. 466с.
27. Секреты программирования для Internet на Java. Ventana Press, Ventana Communications Group, U.S.A., 2016, Издательство "Питер Пресс", 2017. 396с.
28. Семакина И. Г. Информатика. Задачник-практикум: В 2 т.. М.: Лаборатория Базовых Знаний, 2019. 476с.
29. Сокольский М.В. Все об Intranet и Internet. М.: Элиот, 2018. 254с.
30. Тассел Д. Стиль, разработка, эффективность, отладка и испытание программ. М.: Мир, 2011. 56с.

31. Тюрин Ю.Н. Анализ данных на компьютере. М.: ИНФРА-М, Финансы и статистика, 2015. 384с.
32. Флэнэген Д. Java in a Nutshell. O'Reilly & Associates, Inc., 1997, Издательская группа BHV, Киев, 2018. 473с.
33. Чен М.С. Программирование на C++:1001 совет:Наиболее полное руководство по Java и Visual J++ :Пер.с англ. Минск:Попурри, 2017. 640с.ил.
34. Эферган М. C++: справочник. QUE Corporation, 2017, Издательство "Питер Ком", 1998. 256с.
35. G. Yang. «Human face detection in a complex background. Pattern Recognition », 27 (1): 2014. P.53-63.
36. Kotropoulos C. «Acoustics, Speech, and Signal Processing», 2017. ICASSP-97, 2017. IEEE International Conference on pp.2537-2540 v. 4
37. Leung TK. «Finding Faces in Cluttered Scenes Using Random Labeled Graph Matching» 2015.p P.83-95.
38. Yow KC. Feature-based human face detection. Image and vision computing 15 (9), 2017. P.713-735.
39. Sinha, P. Perceiving and Recognizing threedimensional forms. PhD thesis, Massachusetts Institute of Technology, 2016. 278p.
40. Lanitis, A «Image Anal. Classifying variable objects using a flexible shape model »Image Processing and its Applications, 2015., P.70-74.
41. Viola P. «Rapid Object Detection using a Boosted Cascade of Simple Features», proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2001), 2011., vol. 1, 518p.
42. Jones MJ. «Robust real-time face detection», International Journal of Computer Vision, vol. 57, no. 2, 2014., P.137-154.
43. Buchatskiy AN. «Selection of the Optimal Color Space for Reducing False Positives Rate in the Viola-Jones Method», Актуальні проблеми інфотелекомунікацій в науці та освіті, II Міжнародна науково-технічна та науково-методична конференція. Санкт-Петербург, 2013.

44. Ethan R. ORB: an efficient alternative to SIFT or SURF. Computer Vision (ICCV), IEEE International Conference on. IEEE, 2011. P. 2564–2571.

45. Stefan L. BRISK: Binary Robust Invariant Scalable Keypoints. Computer Vision (ICCV), 2011. P. 2548–2555.

46. Pablo F. Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces. In British Machine Vision Conference (BMVC), 2013.

47. Martin A. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Comm. Of the ACM24: 2001. P.381–395,.

48. Патин М.В. Сравнительный анализ методов поиска особых точек и дескрипторов при группировке изображений по схожему содержанию. Молодой ученый. 2016. №11. С. 214-221.

49. Гонсалес Р. Цифровая обработка изображений в среде MATLAB //М.: Техносфера. 2016. 616с.

50. Березький О.М., Дубчак Л.О., Мельник Г.М. Методичні рекомендації до виконання кваліфікаційної роботи з освітнього ступеня “Магістр”. Спеціальність: 123 - Комп’ютерна інженерія. Магістерська програма - Комп’ютерна інженерія"/ Під ред. О.М. Березького. Тернопіль:ЗУНУ,2020.32 с.

51. Гураль І.В., Дубчак Л.О. Методичні вказівки до оформлення курсових проектів, звітів про проходження практики, випускних кваліфікаційних робіт для студентів спеціальності «Комп’ютерна інженерія»/Під ред. О.М. Березького. Тернопіль: ТНЕУ, 2019. 33 с.