

## РЕДАКТОР 3D-ЗОБРАЖЕННЯ ОБЛИЧЧЯ, СТВОРЕНИЙ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ KINECT 2.0

В. Грабовський, О. Мартинович

*Львівський національний університет імені Івана Франка  
вул. Ген. Тарнавського, 107, 79017 Львів, Україна  
volodymyr.grabovskyi@franko.lviv.ua*

Представлені результати розробки з використанням технології Kinect 2.0 програмного додатку, який працює в ОС Windows і призначений для створення 3D-моделі обличчя. Розглянуті деякі ситуації та проблеми, що виникають під час створення такої моделі, а також можливі засоби та методи їх вирішення. Проаналізована можливість вибору моделі, яка найкраще підходить для конкретного типу обличчя, 3D-зображення якого потрібно побудувати, а також особливості припасування вихідного 2D-зображення обличчя до його 3D-моделі і факторів, що впливають на точність цього процесу. Для оцінки якості припасування застосований метод на основі використання “розтяжних коефіцієнтів”, правильний підбір значень яких дозволяє стискати або розтягувати ділянки оригінального зображення на величину, необхідну для найбільш адекватного «обтікання» ними обраної тривимірної моделі. Застосування розробленого програмного продукту проілюстровано на реальному прикладі створення 3D-моделі обличчя.

*Ключові слова:* 3D-зображення обличчя; 3D-модель обличчя; розпізнавання облич; помилки припасування; Kinect 2.0; Software Development Kit Kinect.

### Вступ

Проблема ідентифікації образів, зокрема – розпізнавання облич, вирішення якої ґрунтується на використанні їх 2D- та 3D-зображень, сьогодні є дуже актуальною. Причиною цього є не тільки те, що розпізнавання облич в наш час є одним з найпопулярніших методів біометричної ідентифікації, але і те, що розроблені інструменти розпізнавання отримали широке використання в багатьох наукових, технічних і побутових галузях, зокрема з метою підвищення безпеки, для захисту електронних пристроїв та комп’ютерних програм від несанкціонованого доступу, тощо.

Для процедури розпізнавання облич розроблено багато методів [1, 2], при реалізації яких використовуються різні підходи – як фундаментальні статистичні (у т. ч. з застосуванням методів математичного очікування та коваріаційних матриць, лінійного дискримінантного аналізу [3]), так і (особливо останнім часом) штучні нейронні мережі [4].

В основі більшості з нині існуючих методів розпізнавання лежать методи біометричних вимірювань [5], зокрема – визначення віддалей між т. з. ключовими точками обличчя. Початкові, до комп’ютерні технології розпізнавання використовували прямі вимірювання і тому важко піддавалися автоматизації. Непряма антропометрія розпочалася з впровадження цифрових 2D-зображень та доступу до великих обчислювальних потужностей; останнє дозволило архівувати зображення для подальшого його використання, прямо вимірювати похибку експерта і перейти до автоматизації самого процесу розпізнавання [6].

Автоматичне 2D-розпізнавання облич стало можливим у 60-х роках ХХ ст. і знай-

шло широке застосування і у біометриці, і в технологіях розпізнавання образів та комп'ютерних спільнот [7, 8]. Для того, щоб система 2D-розпізнавання обличчя була ефективною, вона повинна справлятися з можливими ускладненнями, що можуть мати місце при фіксуванні 2D-образу зразка для розпізнавання, і які виникають як за рахунок особливостей самого реєструючого пристрою (наприклад, притаманних йому кольорової гами, роздільної здатності, спотворень, які вносяться об'єктами реєструючого пристрою, тощо), так і суб'єктів, які реєструються (від їх пози, особливостей волосяного покриву обличчя, його виразу, міміки). Важливу роль відіграють також умови, за яких відбувається зйомка, у т. ч. освітлення, масштабування, ракурс зйомки, фон, на якому здійснюється фіксування особи. Зокрема, варіації пози та освітлення під час фіксування зображення і на сьогодні залишаються проблемами, що важко вирішуються при 2D-розпізнаванні. Суттєвим недоліком, притаманним 2D-розпізнаванню, також є те, що обличчя, яке власне є 3D-об'єктом, при перетворенні в його двовимірний образ втрачає важливу інформацію – насамперед, дані про його глибину. Саме ті додаткові знання, які надає тривимірна форма обличчя, виявилася дуже корисними для усунення деяких зі згаданих вище недоліків [6].

Використання технології 3D-розпізнавання обличчя, яка нині стала доступною завдяки збільшенню обчислювальних потужностей сучасної комп'ютерної техніки та нових технологій створення 3D-об'єктів, дозволяє позбутися деяких з вказаних вище недоліків, які мають місце при 2D-розпізнаванні. Це зумовлено тим, що, по-перше, тривимірне зображення обличчя є (до певної міри) незмінним щодо змін освітлення під час запису та порівняння. По-друге, 3D-вимірювання повністю зберігають об'ємні характеристики обличчя, а інформацію про глибину легко можна використовувати ще і для відтворення фонових зображень. Нарешті, вплив варіацій пози реєстрованого об'єкта можна усунути обертянням та зміною положення обличчя в 3D-просторі. Завдяки цьому, об'єктивне та автоматизоване розпізнавання обличчя вже значною мірою перемістилося з 2D на 3D підходи [9-11].

Проблеми, які мають місце при розпізнаванні, часто мають об'єктивний характер. Однією з них є те, що обличчя людини – не жорстка, незмінна в часі структура; воно може змінюватися не лише з часом (наприклад, за рахунок старіння або зміни маси тіла), але і за рахунок міміки [12]. Зокрема, зміни вигляду лица, обумовлені мімікою, були визнані однією з найважчих проблем розпізнавання. Крім того, значні труднощі в розпізнаванні відіграє наявність маскувальних факторів (таких як окуляри, шарфи, шапки, наявність макіяжу, гриму, тощо), які впливають на вигляд спостережуваного, що ще більше ускладнює завдання розпізнавання. Саме тому раніше, оскільки ефективного вирішення цієї проблеми не було, вони часто були нерозв'язні; в даний час вони тією чи іншою мірою отримують своє вирішення завдяки 3D-розпізнаванню [6, 13].

Проведення якісного 3D-розпізнавання обличчя потребує наявності насамперед якісного його тривимірного зображення, яке часто створюється, виходячи з двовимірного з використанням спеціально створених засобів. Створення такого зображення не є тривіальною проблемою і перебуває в стадії інтенсивного вивчення і вирішення, про що свідчить досить широка гама публікацій на цю тему, у т. ч. і оглядових [14-16].

Більшість існуючих на сьогодні рішень задачі створення 3D-зображень обличчя ґрунтуються на використанні одного або декількох ключових компонентів, а саме [17]:

- орієнтирів (особливих точок) обличчя – набору автоматично визначених ключових точок на обличчі (кінчик носа, положення рота, підборіддя, центрів і кутів очей, тощо), які можуть керувати процесом конструювання 3D-зображення;

- опорної моделі лица – усередненої моделі обличчя, яка використовується для ініціалізації оптичного потоку або форми в процедурі затінення;
- тривимірної морфованої моделі обличчя – попередньо створеного маломірного лінійного підпростору правдоподібних геометричних форм обличчя, що дозволяє ефективно (але загалом досить грубо) відновити структуру обличчя.

Хоча окреме використання кожної з відзначених компонент загалом може спростити проблему створення зображення обличчя, їх роздільне застосування вводить деякі властиві кожній з них обмеження [17]. Так, методи, які покладаються тільки на орієнтири обличчя, обмежуються кінцевим набором використаних ключових точок. Методи, що використовують опорну модель обличчя, завдяки тому, що оптичні потоки обмежують деформацію площини зображення, можуть не відновлювати крайні його вирази та нефронтальні пози. Морфована ж модель, забезпечуючи певну надійність, певним чином обмежує можливості створеного зображення, оскільки вона може відображати лише досить грубі геометричні конструкції. Інтеграція деяких з цих компонентів разом могла б пом'якшити проблеми, але основні обмеження все ще проявляються в отриманому кінцевому зображенні.

Таким чином, створення 3D-зображення обличчя для розпізнавання є досить складним завданням [13]. Вирішення цієї проблеми передбачає використання для створення 3D-моделей обличчя різних методів та алгоритмів з метою отримати його таке 3D-зображення, яке б максимально відповідало реальному лицу та було зручним у роботі.

У даній роботі представлені результати розробки програмного продукту, створеного з використанням технології Kinect і призначеного для побудови тривимірного зображення обличчя, виходячи з двовимірного аналога, та дослідження деяких особливостей його роботи.

### **Підходи, використані при створенні редактора 3D-зображення обличчя**

Розроблений додаток представляє собою програму, яка дозволяє реалізувати функції редактора зображення обличчя, у т. ч. створити (підібрати з наявних, представлених Kinect 2.0) 3D-макетів обличчя такий, який найкраще відповідає 2D-зображенню лица, що використовується як вихідне, а також надає необхідні інструменти для обробки цього зображення, потрібної для забезпечення коректного його накладання на вибрану тривимірну модель.

Дані, що необхідні для створення тривимірної моделі обличчя, можуть бути отримані з використанням спеціального давача Kinect 2.0 консолі Microsoft Xbox One [18] – високотехнологічного пристрою, оснащеного камерою з роздільною здатністю Full HD (1920?1080 пікселів), яка фіксує 2D-зображення достатньо хорошої якості. В складі пристрою є також інфрачервоний сенсор глибини та об'ємності, використання якого дозволяє виводити об'ємне зображення з роздільною здатністю 512?424 пікселів. Наявність такого сенсора дозволяє створити “об'ємну карту обличчя” незалежно від умов його освітлення. Зображення, яке надається даним сенсором, є, мабуть, одним з найкращих на сьогодні з широко доступних. Також є можливість використовувати для роботи і зображень кадрів з відео, яке фіксується цим пристроєм з частотою 30 Гц.

Для формування 3D-моделі обличчя у Kinect 2.0 використовується набір стандартних еталонних трикутників, які утворюють специфічний тип полігональної сітки, яка використовується у комп'ютерній графіці і включає в себе набір трикутників (у тривимірному просторі), що мають спільні ребра або вершини. Визначення координат трикутників для кожної конкретної моделі забезпечуються пакетом Software Development Kit

(SDK) Kinect [19], який надає доступ до позицій 1347 вершин, що загалом формують форму обличчя [20]. Повернені дані високої роздільної здатності, які характеризують конкретне обличчя, спочатку інтерпретуються на теоретичні, а не на фактичні риси обличчя, що стоїть перед датчиком. Тобто, повернені дані підходять до загальної, т. з. андрогінної форми обличчя.

За відсутності можливості використання реального давача Kinect 2.0 можна також використати шаблонні моделі, що надаються пакетом SDK Kinect і дають можливість зреалізувати вибір з шести найпоширеніших його форм, які характеризуються як generic (узагальнена), round (кругла), broad (широка), long (видовжена), oval-broad (овально-широка), oval (овальна). Також визначаються 5 основних ключових точок для кожного обличчя: положення правого і лівого ока, кінчика носа, рота і підборіддя. Моделі відрізняються між собою об'ємністю, положенням ключових точок та відстанями між ними.

Для всіх двовимірних зображень обличчя, яке накладається на 3D-модель, значення відстаней між ключовими точками на зображенні мають максимально співпадати зі значеннями відповідних відстаней на цій моделі. Якщо вони співпадають, то дана область на зображенні добре накладається на модель обличчя (тобто “обтікає” 3D-модель), оскільки значення відстаней між ключовими точками на зображенні обличчя та 3D-моделі, на яку воно буде накладатись, у такому випадку будуть однакові. Для оцінки якості обтікання використовується метод на основі “розтяжних коефіцієнтів” – з їх допомогою розтягують або стискають 2D-зображення на величину, рівну відповідному коефіцієнту.

При використанні вказаних вище п'яти ключових точок лица SDK Kinect використовуються чотири розтяжні коефіцієнти, які враховують відстані:

- $k_1$  – від середини правого ока до середини лівого ока;
- $k_2$  – від кінчика носа до точки середини ока;
- $k_3$  – від кінчика носа до верхньої губи;
- $k_4$  – від верхньої губи до підборіддя.

Область, яка пов'язується з коефіцієнтами  $k_1$  та  $k_2$ , відповідає на зображенні за область від ока до носа і називається областю “око-ніс”. Коефіцієнт  $k_3$  відповідає на зображенні за область від носа до рота; ця область називається “ніс-рот”. Коефіцієнт  $k_4$  відповідає на зображенні за область від рота до підборіддя; це область “рот-підборіддя”.

Хорошим “обтіканням” буде вважатись таке, для якого значення коефіцієнтів  $k_1, \dots, k_4$  будуть рівні одиниці або ж максимально наближені до неї. Якщо коефіцієнт більший за 1, то зображення обличчя, яке обтікатиме обрану 3D-модель обличчя, потрібно стиснути на величину коефіцієнта; якщо ж менший – розтягнути.

Процедуру визначення значень відповідних коефіцієнтів та припасування вихідного зображення до накладання його на обрану 3D-модель можна описати за допомогою алгоритму з такими кроками:

1. знаходять відстань між очима у 2D-зображенні обличчя, яке накладається на 3D-модель;
2. знаходять відстань між очима на 3D-моделі зображення обличчя;
3. знаходять відстань від кінчика носа до середини очної лінії для 2D-зображення обличчя, яке накладається на 3D-модель;
4. знаходять відстань від кінчика носа до середини прямокутної лінії для 3D-моделі зображення обличчя;
5. горизонтально розтягують 2D-зображення обличчя на основі коефіцієнта, отриманого шляхом ділення відстані, визначеного в п.1, на відстань, визначену в п.2;
6. вертикально розтягують 2D-зображення обличчя на основі коефіцієнта, отрима-

- ного шляхом ділення відстані, визначеної в п.3, на відстань, визначену в п.4;
7. для розтягнутого зображення, отриманого за допомогою застосування п.6, знаходять відстань від кінчика носа до верхньої губи по вертикалі;
  8. знаходять відстань від кінчика носа до верхньої губи по вертикалі для 3D-моделі обличчя;
  9. розтягують (або стискають) зображення обличчя, яке буде накладене на 3D-модель, по вертикалі за коефіцієнтом, отриманим шляхом ділення відстані, визначеного в п.8, на відстань, визначену в пункті п.7;
  10. для отриманого розтягнутого зображення знаходять вертикальну відстань від верхньої губи до підборіддя;
  11. знаходять вертикальну відстань від верхньої губи до підборіддя для 3D-моделі зображення обличчя;
  12. розтягують (або стискають) зображення вертикально за коефіцієнтом, отриманим шляхом ділення відстані, визначеного в п.10, на відстань, визначену в п.11.

Таким чином, суть припасування зображення до моделі полягає у знаходженні відношень відстаней між ключовими точками на зображенні обличчя, яке потрібно накласти, до відповідних їм відстаней на 3D-моделі обличчя, на яку це зображення потрібно накласти, та зміни вихідного зображення відповідно зі знайденими коефіцієнтами. Величина його розтягу або стиску залежить від того, наскільки форма обличчя на зображенні, яке накладається, є сумісною з 3D-моделлю, на яке це зображення накладається.

Для врахування при розрахунку помилки припасування різного впливу дії вказаних коефіцієнтів застосовано відповідні вагові коефіцієнти, значення яких враховує таке: оскільки розтягування “очі-ніс” впливає на всю область обличчя, то значення вагового коефіцієнта для нього присвоюється найвище – 4; відповідно, менше значення вагового коефіцієнта присвоюється розтягуванню ділянки “рот-ніс”, яке включає в себе лише розтягнення носа вниз – 2; значення ж вагового коефіцієнта розтягуванню “рот-підборіддя”, яке включає тільки розтягнення рота вниз, рівне 1.

Якщо всі етапи вказаного вище алгоритму будуть здійснені коректно, то положення ключових точок на отриманому в результаті цих дій 2D-зображенні будуть змінені в порівнянні з вихідним у відповідності зі ступенем вдалості їх припасування до вибраної 3D-моделі. Однак, в результаті такого припасування, отримане нове 2D-зображення може бути значно деформоване порівняно з вихідним. Ідеальним же буде вважатися те “обтікання”, для якого значення усіх коефіцієнтів будуть рівними одиниці.

3D-модель може бути виготовлена з застосуванням класів середовища .NET (напр., “System.Windows.Media.Media3D”) з використанням “MeshGeometry 3D” для моделювання, “Viewport3D” для перегляду у вікні WPF (Windows Presentation Foundation – інтерфейс програмування додатків (API) [21], “PerspectiveCamera”, “AmbientLight” та “DirectionalLight” для візуалізації моделі під різне освітлення та кути перегляду.

Класи “AForge.NET” забезпечують фільтри для обробки вихідного зображення, що змінює яскравість та контрастність. “OpenCV” надає інструмент “HaarClassifier” для виділення контурів обличчя та очей із вхідних зображень [22].

Класи “.NET.Drawing” використовуються для обробки зображень GDI+ (Graphics Device Interface) – однієї з трьох основних компонент, які разом з ядром і Windows API є базовими для інтерфейсу Windows. Класи “System.Windows.Media” використовуються для презентації у вікні WPF, що відповідає за графічний інтерфейс користувача і є частиною Microsoft .NET Framework).

### Опис програми

Проект розроблений на платформі Microsoft .NET та підтримується ОС Windows. Для створення проекту використано середовище розробки Microsoft Visual Studio 2017. Логіка програми реалізована мовою програмування. Інтерфейс програми реалізований за допомогою WPF та класів System.Windows.Media. Для коректної роботи програми потрібно встановити .NET Framework 4.5 або новішу її версію.

Після створення додатку за шаблоном WPF Application, системою створюються ряд файлів, головним є MainWindow.xaml, який реалізує функціонал головного вікна програми. Файли Window 1-7.xaml реалізують інтерфейси допоміжних вікон програми для роботи зі зображеннями та макетами обличчя. Реалізація інтерфейсу програми здійснюється за допомогою мови розмітки XAML (Xtensible Application Markup Language). На початку XAML-файлу визначений елемент верхнього рівня Window – вікно програми; в ньому визначено елемент Grid – контейнер верхнього рівня, в який можна додавати інші елементи. Кожен елемент може мати певні атрибути, яким притаманні властивості, які дозволяють змінювати розміри, фон, позицію елементів вікна. <ContextMenu> та <MenuItem> описують елементи меню та їх властивості. Властивість Click задає ім'я обробників подій, які реалізують функціонал елементів інтерфейсу.

Вигляд інтерфейсу розробленого додатку показаний на рис.1.

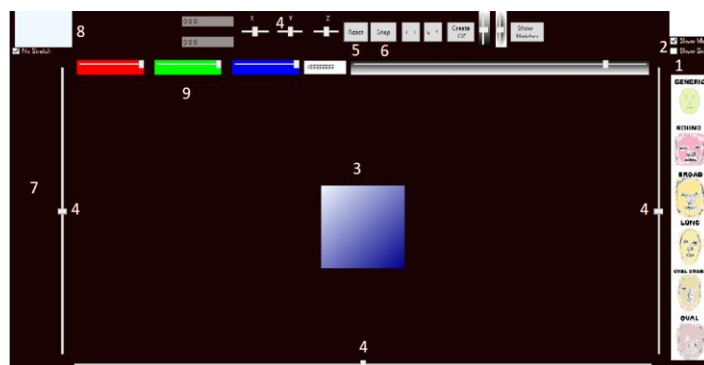


Рис. 1. Загальний вигляд інтерфейсу програми:

1 – панель 3D-моделей обличчя; 2 – перемикачі показу сітки обличчя; 3 – куб – основа для 3D-моделі обличчя; 4 – повзункі налаштування перегляду; 5 – кнопка “Reset”; 6 – кнопка “Snap”; 7 – панель відображення знімків моделей обличчя; 8 – елемент відкривання та завантаження зображень; 9 – палітра кольорів RGB.

В проєкті реалізовано можливість вибору шістьох форм лиць, наданих SDK Kinect, а саме: generic (узагальнена), round (кругла), broad (широка), long (видовжена), oval-broad (овально-широка), oval (овальна). Шаблони відповідних макетів 3D-моделей обличчя розміщені на правій панелі розробленого інтерфейсу (1 на рис. 1). Вище розташовані два перемикачі (2 на рис. 1), які дозволяють показати або згорнути сітки обличчя.

В центрі робочої області (3 на рис. 1) розміщений куб, який є основою для 3D-моделі обличчя і дозволяє виконувати її переміщення та повертання в 3D-просторі. Зміна положення куба (перегляд моделі під різними кутами) здійснюється за допомогою повзунків налаштування перегляду (4 на рис. 1).

При виборі будь-якої з шести моделей лица, зображених на панелі 1, завантажуються відповідна 3D-модель обличчя. Щоб повернути всі позиції за замовчуванням та параметри обертання камери, потрібно натиснути на кнопку “Reset” (5 на рис. 1) або двічі клацнути мишкою на візуалізованій моделі.

Вихідним елементом моделі може бути в знімок моделі обличчя, який можна зробити за допомогою кнопки “Snap” (6 на рис. 1). Він буде являти собою 3D-обличчя, зберігатися в пам'яті та відображатись у вертикальній панелі ліворуч (7 на рис. 1). Тут відображаються останні 6 знімків. Прокрутити решту збережених знімків можна за допомогою колеса миші. Натиснувши на вибране зображення, можна переглянути його або зберегти в файл у папці проекту Snap.

У верхньому куті зліва (8 на рис. 1) розташований елемент, який дозволяє відкривати та завантажувати зображення обличчя. Зображення можна завантажити, помістити в центр робочої області та здійснити його мінімальну обробку за допомогою зміни палітри кольорів RGB (9 на рис. 1), яка також показує код кольору в шістнадцятковому форматі та значення яскравості і контрасту.

Коли зображення обличчя вибирається та завантажуються, програма знаходить на ньому ключові точки (жовті кільця на зображенні лица, рис. 2). Виявлення очей здійснюється за допомогою OpenCV HaarClassifier. Кольорові кола на зображенні – знайдені ключові точки, за якими відбувається “обтікання” обличчя. Також з’являється лупа (1 на рис. 2), що відображає збільшене зображення вибраної області на зображенні та дозволяє коректувати положення відповідної особливої точки на зображенні обличчя і визначити таким чином значення відповідних коефіцієнтів.

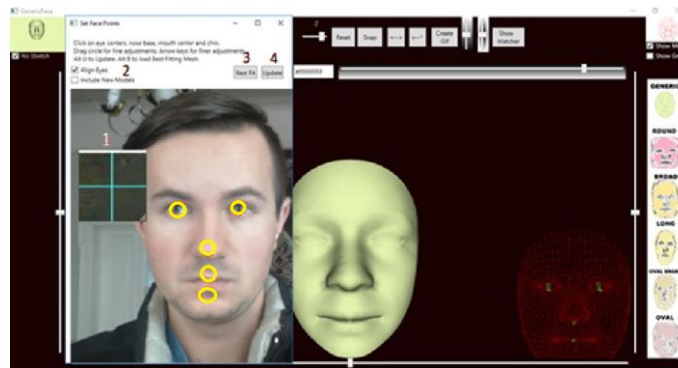


Рис. 2. Ключові точки моделі на зображенні обличчя:

1 – лупа; 2 – прапорець для забезпечення симетричності очей; 3 – кнопка “Best fit”; 4 – кнопка “Update”.

Щоб накласти зображення на обличчя, потрібно прив'язати його до ключових точок на макеті обличчя. Щоб точно припасувати ключові точки, потрібно вибрати їх, а потім за допомогою лупи переглянути збільшене зображення вибраної області та точно перемістити ключові точки в потрібну частину зображення. Можна використати прапорець “Align eyes” (2 на рис. 2) для забезпечення симетричності очей. Кнопка “Best fit” (3 на рис. 2) потрібна, щоб вибрати модель, яка найкраще підходить для нового обличчя. Щоб використати поточну вибрану модель обличчя для наступного накладання на неї двови-

мірного зображення, використовується кнопка "Update" (4 на рис. 2).

В процесі накладання зображення на модель обличчя інтерфейс програми має вигляд (рис. 3).

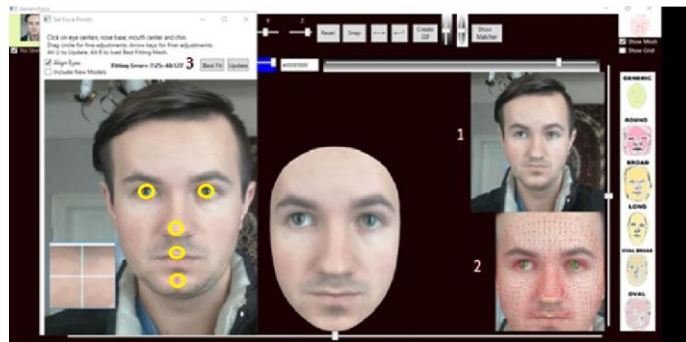


Рис. 3. Процес накладання зображення на обличчя:

1 – розтягнуте зображення обличчя; 2 – фіксація розтягнутого обличчя щодо моделі; 3 – поле значень помилок припасування "Fitting error".

Після оновлення моделі потрібно мати на увазі таке:

- верхнє праве зображення (1 на рис. 3) – розтягнуте обличчя, яке буде використовуватися, як текстура;
- зображення внизу праворуч (2 на рис. 3) – зафіксоване розтягнуте зображення обличчя щодо моделі, на яку воно накладатиметься;
- "Fitting error" (3 на рис. 3) – показує значення помилок припасування параметрів "очі-ніс", "ніс-рот", "рот-підборіддя" і загальної похибки, при розрахунку якого враховані вказані вище вагові коефіцієнти.

Після прикріплення обличчя можна виконувати його переміщення та повороти, щоб отримати бажаний вигляд. В підсумку, створене 3D-зображення обличчя буде виглядати як показано на рис. 4. Натискання кнопки "Snap" зберігає 3D-зображення обличчя в пам'яті – збережене зображення доступне на лівій панелі (7 на рис. 1).

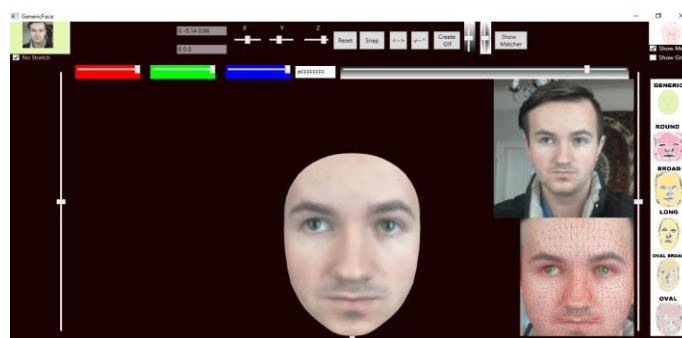


Рис. 4. Накладене зображення на 3D-модель обличчя.



### Приклад використання створеної програми

Однією з головних проблем у вирішенні задачі створення 3D-зображення обличчя, поряд з вибором моделі, є накладання зображення обличчя на 3D-модель. Проблема полягає в тому, що лице кожної людини має свої особливості, які зумовлені індивідуальністю форми лица і практичною неможливістю ідеально змоделювати таку модель обличчя, яке буде підходити для кожного конкретного зображення лица.

За допомогою описаного вище алгоритму обтікання обличчя та введеної загальної похибки обтікання можна обрати таку 3D-модель обличчя, яка найкраще підходить для наданого зображення обличчя. Врахування помилок обтікання допоможе відкоригувати положення ключових точок на зображенні обличчя, яке накладається на модель. Продемонструємо це на прикладі роботи створеного додатку.

На рис. 5 показані особливі точки обличчя, визначені програмою при накладанні обраного 2D-зображення на обрану 3D-модель.

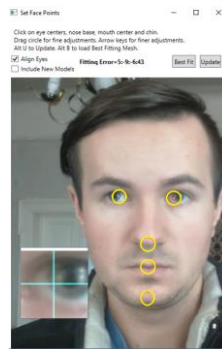


Рис. 5. Ключові точки на зображенні, що накладається на 3D-модель.

Як видно з рис. 5, при накладанні 2D-зображення на 3D-модель на ньому з'являються кольорові кола, які показують знайдені програмою ключові точки для цього лица у вибраній 3D-моделі. В полі "Fitting error" (див. п.3, рис. 3) відображені розраховані значення похибок їх розміщення на зображенні (у випадку, показаному на рис. 5, вони складають: похибка "ока-ніс" – 5, похибка "ніс-рот" – -9, похибка "рот-підборіддя" – 6 і загальна похибка (розрахована з вагою 4 для помилки "око-ніс", 2 – для "ніс-рот" та 1 – для "рот-підборіддя" – 43). Для корекції значень помилок використовують т. з. локатори точок обличчя; їх переміщення (суміщення перехрестя ниток лупи з потрібною точкою) похибок приводить до витягування або стиснення зображення. Щоб компенсувати помилку "око-ніс", потрібно, використовуючи лупу, сумістити точки положення очей, визначені для даної моделі, з їх реальним положенням на фотографії або скоректувати точку положення носа. Так, щоб компенсувати додатну помилку "око-ніс", слід переміщувати положення точок очей далі один від одного або точку носа перемістити вище; відповідно, щоб компенсувати від'ємну помилку "ніс-рот", потрібно переміщувати локатори носа та рота подалі один від одного. Щоб компенсувати від'ємну помилку "рот-підборіддя", потрібно перемістити локатори рота та підборіддя подалі один від одного, а для компенсації додатної помилки – перемістити ці локатори ближче один до одного.

Але, якщо здійснено багато виправлень, деякі точки на зображенні обличчя можуть бути не суміщені з відповідними точками обличчя на моделі. Кнопка "Update" дозволяє

переглянути зміни, які відбулись під час переміщення локаторів точок обличчя для зображення, яке припасовується, у правому нижньому куті інтерфейсу програми. Після “обтікання” обличчя можна виконувати його переміщення та повороти, щоб отримати бажаний вигляд та ракурс отриманого 3D-зображення.

Використовуючи обчислене значення похибки “обтікання” обличчя, можна здійснити вибір тієї 3D-моделі обличчя, яка найкраще підходить для даного зображення лиця конкретного типу. Суть такого підбору розглянемо на прикладі з використанням усіх шести шаблонних 3D-моделей, що надаються SDK Kinect.

Результати “обтікання” зображенням обличчя 3D-моделей типу “generic”, “round”, “broad”, “long”, “oval” та “oval broad” і значення відповідних помилок показані на рис. 6.

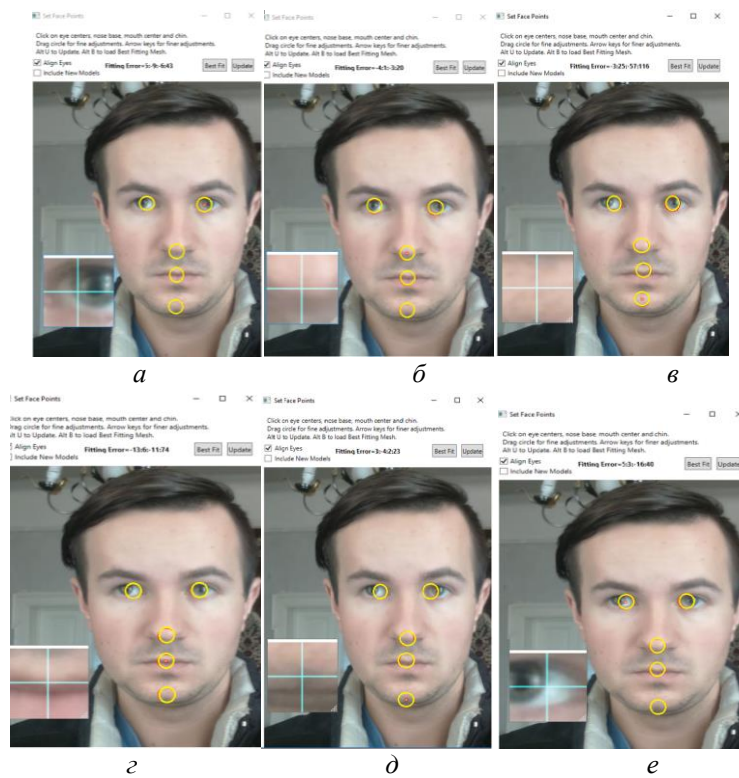


Рис. 6. Результат обтікання зображення та значення помилок для 3D-моделей: а – “generic”; б – “round”; в – “broad”; г – “long”; д – “oval”; е – “oval broad”.

З зображень, представлених на рис. 6, можна бачити, що загальна похибка співпадіння ключових точок на зображення обличчя, яке накладається, для моделі “generic” становить 43, для “round” – 20, для “broad” – 116, для “long” – 74, для “oval” – 23 та “oval broad” – 40. Звідси можна зробити висновок, що для вибраного зображення обличчя найкращою 3D-моделлю буде “round”, оскільки похибка “обтікання” для цієї моделі є найменшою.

Здійснивши припасування 2D-зображення обличчя на вибрану 3D-моделю (для цю-

го необхідно після вибору оптимальної 3D-моделі (у нашому випадку – модель “round”) підібрати потрібні параметри обтікання шляхом переміщення відповідних локаторів ключових точок, тобто добитися суміщення визначених програмою ключових точок моделі з їх зображенням на фотографії) та натиснувши кнопку “Snap” (п. 6 на рис. 1), отримуємо відповідне 3D-зображення обличчя (рис. 7).



Рис. 7. Зображення обличчя, отримане з застосуванням моделі “round”.

На основі даних досліджень можна зробити висновок, що, оскільки кожне людське обличчя має унікальні особливості і форму, для того, щоб ідеально здійснити “обтікання” його зображення, необхідно в кожному конкретному випадку розробити притаманну саме йому 3D-модель, що практично неможливо. Тому на практиці підбирається та модель з наявних в додатку, яка найкраще підходить для конкретного типу лица, і використовується для створення відповідного 3D-зображення обличчя.

### Висновки

Створено додаток для роботи в операційній системі Windows, який виконує функції редактора 3D-зображення обличчя і дозволяє моделювати деякі проблеми, які виникають в процесі створення такого зображення, та надає способи їх вирішення. Для реалізації програми використані технології .NET Framework, WPF, Kinect 2.0 SDK.

За допомогою розробленого додатку продемонстровано процес створення 3D-зображення обличчя, виходячи з його двовимірного зображення. Показано, що головну проблему, яка виникає в процесі створення зображення обличчя – здійснення коректного “обтікання” зображення та накладання маски на модель лица – можна вирішити за допомогою алгоритму з використанням “розтяжних коефіцієнтів”, який мінімізує спотворення, які можуть мати місце при накладанні вихідного 2D-зображення на 3D-модель обличчя. Використовуючи вказаний алгоритм та розраховані на його основі похибки “обтікання”, на реальному прикладі продемонстровано процес підбору 3D-моделі обличчя, яка найкраще підходить для вибраного двовимірного зображення лица, та накладання відкоригованого вихідного двовимірного зображення обличчя на вибрану 3D-модель.

### СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. *Chellappa R. Human and Machine Recognition of Faces / R. Chellappa, C. Wilson, and S. Sirohey // A Survey. Proc. IEEE, – 1995. – V. 83, no. 5. – p. 705-740.*
2. *Tolba A. S. Face Recognition: A Literature Review / A. S. Tolba, A.H. El-Baz, and A.A. El-Harby // International Journal of Signal Processing. – 2006. – V. 2; 2. – p. 88-103.*

3. *Belhumeur, J. P. Hespanha*. Eigenfaces vs. fisherfaces - recognition using class specific linear projection / J. P. Hespanha Belhumeur, , and D. J. Kriegm // IEEE Transactions on Pattern Analysis and Machine Intelligence – 1997. – V. 19(7). – p. 711–720.
4. *AL-Allaf O. N. A*. Review of face detection systems based artificial neural networks algorithms / O. N. A. AL-Allaf // The International Journal of Multimedia & Its Applications. – 2014. – Vol.6, No.1. – p. 1-16.
5. *Tiwari Shradha*. A Review of Advancements in Biometric Systems / Shradha Tiwari , J.N. Chourasia, Vijay S.Chourasia // Int. J. of Innovative Research in Advanced Engineering (IJIRAE). – 2015. – V. 2 Is. 1/– p. 187-204.
6. *Smeets Dirk*. Objective 3D face recognition: Evolution, approaches and challenges / Dirk Smeets, Peter Claes, Dirk Vandermeulen, John Gerald Clemen // Forensic Science International. – 2010. – 201 – p. 125–132
7. *Bledsoe W. W*. The model method in facial recognition. / W.W. Bledsoe – Technical Report PRI 15. – Panoramic Research, Inc. Palo Alto, California. – 1964.
8. Handbook of Face Recognition. / Eds. S.Z. Li, A.K. Jain – Springer-Verlag, New York. – 2005. – p. 398.
9. *Chang K. I*. Face recognition using 2D and 3D facial data / K.I. Chang, K.W. Bowyer, P.J. Flynn // ACM Workshop on Multimodal User Authentication. – 2003. p 25-32.
10. *Chellapa R*. Human and machine recognition of faces: a survey / R. Chellapa, C.L. Wilson, S. Sirohey. // Proc. IEEE. – 1995. – 83. – p. 705–740.
11. *Zhao W*. Face recognition: a literature survey / W. Zhao, R. Chellapa, P.J. Phillips, A. Rosenfeld. // ACM Comput. Surv. – 2003. – 35. – p. 99–458.
12. *Bronstein A. M*. Three-Dimensional Face Recognition / A. M. Bronstein, M. M. Bronstein & R. Kimmel // International Journal of Computer Vision. – 2005. – 64(1). – p. 5–30.
13. *Hu Guosheng*. Efficient 3D morphable face model fitting / Guosheng Hu , Fei Yan, Josef Kittler, William Christmas, Chi Ho Chan, Zhenhua Feng , Patrik Huber // Pattern Recognition. – 2017. – V. 67 – p. 366-379.
14. *Ter Haar F*. 3D face model fitting for recognition / F. ter Haar, R. Veltkamp // European Conference on Computer Vision. – 2008. – p. 652-664.
15. *Brunton A*. Review of statistical shape spaces for 3D data with comparative analysis for human faces / A. Brunton, A. Salazar, T. Bolkart, S. Wuhler // Comput. Vis. Image Underst. – 2014. – 128 – p. 1–17.
16. *Bas A*. Fitting a 3D Morphable Model to Edges: A Comparison Between Hard and Soft Correspondences / A. Bas, W. A. P. Smith, T. Bolkart, S. Wuhler. // Asian Conference on Computer Vision Workshop on Facial Informatics, Nov. 2016, Taipei, Taiwan. Springer, Lecture Notes in Computer Science. – 2017. – vol. 10117. – p. 377-391.
17. *Sela M*. Unrestricted Facial Geometry Reconstruction Using Image-To-Image Translation / Matan Sela, Elad Richardson, Ron Kimmel. // The IEEE International Conference on Computer Vision (ICCV). – 2017. – p. 1576-1585.
18. *Zhang Z*. Microsoft Kinect sensor and its effect / Z. Zhang // IEEE MultiMedia. – 2012. – V. 19, no. 2. – p. 4–10.
19. *Rahman M*. Beginning Microsoft Kinect for Windows SDK 2.0 : Motion and Depth Sensing for Natural Interfaces / M. Rahman. – Apress. – 2017. – 297 p.
20. Kinect for Windows Software Development Kit (SDK) [Electronic source]. – Available from: <https://www.techspot.com/drivers/driver/file/information/18015/>.
21. “WPF overview” [Electronic source]. – Available from:

[https://www.tutorialspoint.com/wpf/wpf\\_overview.htm](https://www.tutorialspoint.com/wpf/wpf_overview.htm)

22. Face Detection using Haar Cascades [Electronic source]. – Available from: [https://docs.opencv.org/3.4.1/d7/d8b/tutorial\\_py\\_face\\_detection.html](https://docs.opencv.org/3.4.1/d7/d8b/tutorial_py_face_detection.html)

## REDACTOR OF 3D FACE IMAGE CREATED USING KINECT 2.0 TECHNOLOGY

**V. Hrabovskyi, O. Martynovych**

*Lviv Ivan Franko National University,  
107 Gen. Tarnavskogo St., UA – 79017, Lviv, Ukraine  
volodymyr.grabovskyi@lnu.edu.ua*

In this paper, we present the results of creating a software product that provides the ability to build 3D image of the face and allows you to solve some of the problems that arise when creating them. To implement the program, the .NET Framework, WPF, Kinect 2.0 SDK, FaceAPI were used.

Some situations and problems that arise when creating such an image are considered, as well as possible means and methods of their solution. The possibility of choosing a 3D model that is best suited for the face type, the image of which will be constructed, as well as the optimal fitting of the original 2D image of this face to the selected 3D model was analyzed.

The role of factors influencing the correctness of the process of fitting the two-dimensional image of a person's face to its three-dimensional model was analyzed. For such fit and evaluation of its quality, a method based on the use of "stretching coefficients" was used. The correct choice of values of these coefficients that allow you to compress or stretch an entire image or its specific areas by the amount necessary for the most adequate fit of it to the selected three-dimensional model, makes possible to realize the optimal attachment of the face image to the selected 3D model.

Using the developed program, the process of creating a 3D image of the face was demonstrated, based on its two-dimensional image. It was shown that the main problems that arise in the process of creating a three-dimensional image of a person's face are the choice of the optimal 3D model of the face, the correct fitting of the two-dimensional image to the selected model of the face and its overlay on this model. These problems can be solved by applying an algorithm using "tensile coefficients" that minimizes distortion of the overlay of a 2D image of a face on a 3D face model and optimizes the overlay process itself. Using this algorithm and the use of calculated fit errors, the selection process of the 3D model of the face, which is best suited for the selected two-dimensional face of the face, and the achievement of minimal distortions when applied to this model, was demonstrated.

The developed program is intended for work in the Windows operating system. To work correctly, you need to use .NET Framework 4.5 or later.

*Keywords:* 3D face image; 3D face model; face recognition; fitting error; Kinect 2.0; Software Development Kit Kinect.

*Стаття: надійшла до редакції 05.07.2020,  
доопрацьована 04.08.2020,  
прийнята до друку 05.08.2020*