

МЕТОДИ РОБОТИ З ТЕКСТУРАМИ ЗА ДОПОМОГОЮ ЗАСОБІВ ОБРОБЛЕННЯ ЗОБРАЖЕНЬ НА МОВІ PYTHON

У статті проаналізовано наявні підходи й алгоритми оброблення зображень та подальшого їх використання для оброблення фото одягу, перетворення їх на фото текстури для 3D-моделей. Програмна частина, описана в цій статті, являє собою бекенд для онлайн-сервісу примірювання одягу на 3D-моделях у браузері, глобальним результатом якої буде тестова версія цього сервісу.

Ключові слова: оброблення зображень, видалення фону, примірювання одягу в браузері, 3D у браузері.

Вступ

Мета фінального продукту, частиною якого стане практична частина цієї статті, – зробити інтернет-шопінг якомога зручнішим і комфортнішим, звести потребу в безпосередньому примірюванні одягу та інших видах шопінгу до мінімуму.

Із кожним роком усе більше і більше людей купують найрізноманітніші речі онлайн, і, звісно, одна з базових потреб людини – це одяг. Однак через те, що люди відрізняються один від одного як габаритами (зростом, вагою, довжиною талії тощо), так і вподобаннями, то процес купівлі саме одягу через інтернет стає таким важким і інколи неприємним. Вирішивши основні проблеми, які виникають у людей під час купівлі одягу онлайн, можна зробити його ще більш популярним.

Отже, є декілька основних проблем у цьому процесі:

1. Людина не знає, як саме на ній виглядатиме певна річ.
2. Людина не впевнена, чи підходить їй розмір.
3. Людина хоче візуально бачити, чи пасуватиме певна річ до іншого одягу з її гардеробу (цю проблему не завжди вирішує навіть безпосереднє примірювання одягу).

На початковому етапі саме останню проблему розв'язуватиме фінальний продукт. Потрібно дати змогу кінцевим користувачам «одягати» 3D-модель у браузері в одяг, що продається на сайті (теоретичний сайт одягу, до якого буде під'єднано фінальну систему), а також завантажувати свої фотографії одягу, які переробляться на текстуру для моделі, що дасть змогу візуалізувати те, як одяг користувач виглядатиме разом із речами, які продаються на сайті.

Вибір технологій, які найкраще підходять для обраної задачі

Проаналізувавши три основні технології (OpenCV, Google Cloud Vision API, scikit-image), які використовують для комп'ютерного зору та оброблення зображень, було зроблено такі висновки.

Бібліотека OpenCV є повністю безплатною, підтримується багатьма мовами програмування, дуже гарно продокументована і має великий список можливостей, які охоплюють саме те, що потрібно для поставленої задачі.

Усю бекенд-частину фінального продукту було вирішено писати на мові Python, тому відповідно буде краще, якщо основна задача бекенд-частини, як і базові, буде теж на мові Python.

Реалізація алгоритму для розв'язання основної задачі

На вхід програмі подається зображення однієї сторони одягу (рис. 1).

На виході програма має перетворити фотографію одягу на частину текстури для 3D-моделі, а саме: виокремити потрібну інформацію з фотографії (самого одягу), видалити фон (відповідно фотографія має бути у форматі png, який підтримує прозорий фон) і зайві частини фотографії, довести фотографію до певних розмірів, які потрібні для текстури.

Тобто найважливішою частиною є знаходження потрібного контенту на зображенні, а саме одягу, і видалення зайвого фону.



Рис. 1. Приклад зображення одягу

Опис логіки алгоритму

Перше і найголовніше, що потрібно зробити, це – «Edge detection», тобто знайти контури основного елемента на зображенні, а саме одягу. Використаємо для цього стандартний алгоритм комп'ютерного зору Кенні.

Алгоритм Кенні (детектор границь Кенні, оператор Кенні) у дисципліні комп'ютерного зору – оператор виділення границь зображення. Був розроблений Джоном Кенні [2].

Алгоритм виявлення границь Кенні можна розділити на п'ять різних етапів:

- Застосуйте фільтр Гауса, щоб згладити зображення та зняти шум.
- Знайдіть градієнти інтенсивності зображення.
- Застосуйте немаксимальне придушення, щоб позбутися хибної реакції на виявлення ребер.
- Застосуйте подвійний поріг для визначення потенційних ребер.
- Відстежте край за допомогою гістерезису, завершіть виявлення країв, придушивши всі інші ребра, які слабкі та не з'єднані із сильними краями.

Для розмиття зображення з метою видалення шуму оператор Кенні використовує розмивання Гауса з $\alpha = 1.4$:

$$\mathbf{B} = \frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} * \mathbf{A}.$$

Для пошуку градієнтів границі позначають там, де градієнт набуває найбільшого значення. Вони можуть мати різні напрямки, тому алгоритм Кенні використовує чотири фільтри для визначення горизонтальних, вертикальних і діагональних ребер у розмитому зображенні.

$$\mathbf{G} = \sqrt{\mathbf{G}_x^2 + \mathbf{G}_y^2}$$

$$\Theta = \arctg\left(\frac{\mathbf{G}_y}{\mathbf{G}_x}\right).$$

Кут нахилу градієнта округлюється і може набувати значень 0, 45, 90, 135.

Алгоритм повністю описано в роботі Джона Кенні «A Computational Approach to Edge Detection» [1].

Наступним кроком буде виділення із результатів зображення, після оброблення методом Кенні, самого контуру. З цим також допоможе стандартний метод оброблення зображень, що приймає на вхід зображення, оброблене методом Кенні, та виокремлює основну інформацію (див. приклади нижче).

Після того як ми знайшли фінальний контур контенту, який хочемо залишити на зображенні (в нашому прикладі – елемента одягу), маємо створити так звану маску цього контуру і видалити все із зображення, крім цієї маски.

Так ми отримуємо зображення з видаленим фоном, останній крок – обрізати зайве і звести зображення до одного розміру.

Алгоритм обрізання фотографії також використовує контур основного зображення і обрізає його до границь цього контуру.

Тестування фінального алгоритму на різних типах одягу

Протестуємо алгоритм на різних типах одягу. В початковій версії бібліотеки було вирішено додати лише два елементи одягу, а саме: футболки і штани, але алгоритм працюватиме для будь-якого одягу.

Протестуємо на футболці, вхідна фотографія та результат – на рис. 2.



Рис. 2. Початкове зображення одягу і результати після оброблення програмою

Результат такий, як і очікувалось, алгоритм працює коректно.

Додавання алгоритму до фінальної програми, тестування цілісної системи

Кінцевий вигляд програми

Фінальна програма являє собою тестування написаної javascript-бібліотеки, яка реалізує примірювання одягу на 3D-моделях.

На практиці це мало би працювати таким чином: користувач додає бібліотеку до себе на сайт, на певну сторінку додає головний Canvas із бібліотеки, який і надає всі основні можливості бібліотеки.

Логіка роботи

Що потрібно буде зробити працівникам інтернет-магазину?

Щоб програма коректно працювала при додаванні нового товару на сайт, потрібно завантажити



Рис. 3. Приклад правильного розміщення одягу

дві додаткові фотографії елемента одягу, спереду і ззаду відповідно. Одяг має лежати на рівній поверхні однотонного яскравого кольору (рис. 3).

Після додавання фотографій на серверній частині з кожної фотографії буде видалятися фон і змінюватися розмір до стандартизованої певної

величини, яка є оптимальною для текстури. Після всіх маніпуляцій зберігається вже оновлене зображення.

Як відбуватиметься відображення певного елемента одягу на 3D-манекені?

Користувач обрає певний елемент одягу, щоб побачити, як він виглядатиме на манекені. За елементом одягу визначають, який це тип одягу (наприклад футболка), з бази беруть відповідну 3D-модель та «одягають» на манекен.

Фотографії текстур накладають на вже одягнену на манекен 3D-модель.

Кожна 3D-модель поділена на дві частини (передню і задню, відповідно до текстур), що мають свою певну UV-розгортку, яка і описує те, як має накладатися текстура на об'єкт. Приклад накладання обробленої фотографії на 3D-модель у редакторі blender – на рис.4.



Рис. 4. Демонстрація накладання одягу на 3D-модель

Демонстрація можливостей фінального проекту

На сервісі є три сторінки:

- сторінка вибору одягу;
- сторінка візуалізації манекена, який одягнений у вибраний одяг;
- сторінка адміністратора, де можна додавати та редагувати елементи одягу.

На сторінці вибору одягу відображаються елементи одягу, які наявні на сайті (тобто які попередньо додав адміністратор). Будь-який наявний елемент одягу можна приміряти на манекені. Також є можливість комбінувати різні типи одягу.

На основній сторінці можна подивитися на манекен, одягнений у вибраний на попередній сторінці одяг. Манекен, за бажанням, можна обернути та наблизити/віддалити.



Рис. 5. Манекен, одягнений у широкі штани і звичайну футболку

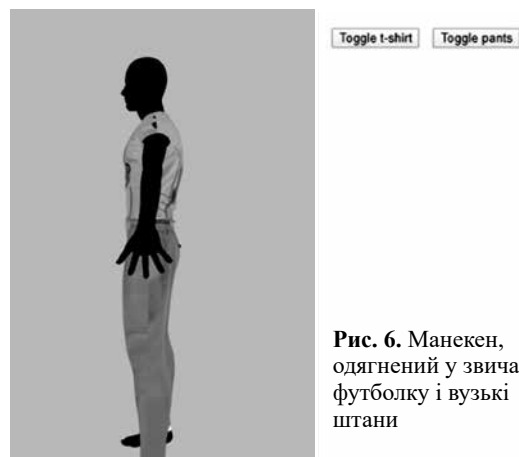


Рис. 6. Манекен, одягнений у звичайну футболку і вузькі штани

Сторінка адміністратора дає змогу працювати з елементами одягу. Є можливість додати новий елемент, для якого можна задати розмір, або видалити раніше додані елементи.

На цій сторінці адміністратор може додавати новий комплект одягу в базу даних, який згодом

можна буде переглядати на манекені. Адміністратору пропонують надати назву комплекту, вибрати зображення комплекту, а також чотири зображення для футболки та штанів, спереду і ззаду.

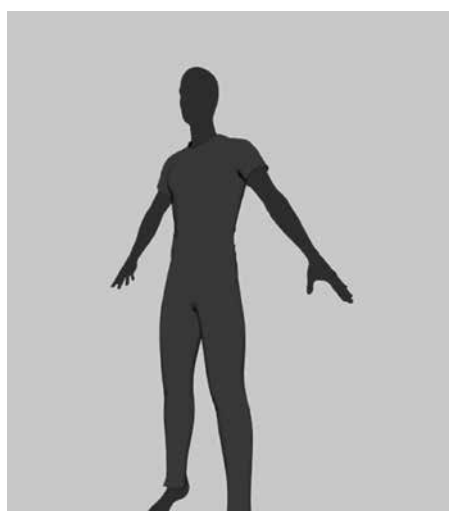


Рис. 7. Користувацький інтерфейс адміністратора

Висновки

У статті було досліджено технології та алгоритми комп'ютерного зору. В результаті було створено алгоритм, який дає змогу перетворювати фотографію будь-якого одягу на частину текстури для 3D-моделі. Фінальним результатом стала javascript-бібліотека, написана з ви-

користанням цього алгоритму. Бібліотека дає змогу користувачам інтернет-магазинів приміряти одяг на 3D-манекенах у браузері. Для тестування бібліотеки було створено сервіс для примірювання свого одягу на віртуальних 3D-моделях. Бібліотека може бути корисною у тих сферах, де потрібна візуалізація одягу, щоб покращити взаємодію з користувачами та клієнтами.

Список літератури

1. A Computational Approach to Edge Detection by John Canny [Electronic resource]. – Mode of access: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.420.3300&rep=rep1&type=pdf>.
2. OpenCV Canny Edge Detection [Electronic resource]. – Mode of access: https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_canny/py_canny.html.

References

- A Computational Approach to Edge Detection by John Canny. Retrieved from <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.420.3300&rep=rep1&type=pdf>.
- OpenCV Canny Edge Detection. Retrieved from https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_canny/py_canny.html.

D. Gamayun, M. Korniiichuk

METHODS OF WORKING WITH TEXTURES USING COMPUTER VISION ON PYTHON

The article describes the analysis of available approaches and algorithms of image processing and their subsequent use for processing photo clothing, transforming them into photo textures for 3D models. The software part described in this article is the back-end for the online service of fitting clothes on 3D models in the browser; the global result of which will be the test version of the service.

The global goal of the final product, part of which will be a practical part of this article, is to make online shopping as convenient and comfortable as possible, to minimize the need for face-to-face clothing and other types of shopping to a minimum.

Every year more and more people buy a variety of things online and, of course, one of the basic human needs is clothing. But due to the fact that each person differs from each other, both in size (height, weight, waist length, etc.) and preferences, the process of buying clothes online, becomes so difficult and sometimes unpleasant. By solving the main problems that people have when buying clothes online, you can make it even more popular.

There are several major problems when buying clothes online: a person does not know exactly what this or that thing will look like on him; a person is not sure whether this or that size suits him; a person wants to visually see if a certain thing will match other clothes from his wardrobe (this problem is not always solved, even by face fitting).

At the initial stage, the final product will solve the last problem.

To solve this problem, end users need to be able to “dress” the 3D model in the browser both in the clothes sold on the site (theoretical clothing site to which the final system will be connected) and upload their photos of clothes that will be processed into texture. for a model that will allow you to visualize how the user’s clothes will look along with the things sold on the site.

After analyzing the three main technologies (OpenCV, Google Cloud Vision API, scikit-image) used for computer vision and image processing, the following conclusions have been drawn.

The OpenCV library is completely free, supported by many programming languages, very well documented and has a long list of features that include exactly what you need for the task.

Keywords: image processing, background removal, fitting clothes in the browser, 3D in the browser.



Creative Commons Attribution 4.0 International License (CC BY 4.0)

Матеріал надійшов 10.06.2020