

Poprawa jakości zdjęć we współczesnych smartfonach

Multimedialne Systemy Interaktywne
Jakub Sachajko, 179976

April 5, 2023

Spis treści

1	Wybór tematu	2
2	Analiza wymagań	3
2.1	Cel aplikacji	3
2.2	Docelowa grupa użytkowników	3
2.3	Docelowa platforma	3
2.4	Środowisko działania aplikacji	3
2.5	Określenie wymagań	3
2.5.1	Funkcjonalne	3
2.5.2	Wydajnościowe	3
2.5.3	Jakościowe	3
2.5.4	Sprzętowe	3
2.5.5	Skalowalność	4
2.5.6	Rozszerzalność	4
2.5.7	Otwartość systemu	4
2.5.8	Niezawodność	4
2.6	Definicja interfejsów użytkownika	4
2.7	Opcjonalne rozwiązania funkcjonalne	4
2.8	Inne założenia poczynione z użytkownikiem	4
2.9	Odporność na awarie, bezpieczeństwo systemu	4
2.10	Określenie niezbędnej dokumentacji	4
2.11	Zakładane ograniczenia aplikacji	5
3	Projekt systemu	5
3.1	Architektura systemu oraz interfejsu użytkownika	5
3.2	Środowisko tworzenia aplikacji	6
3.3	Projekt najważniejszych struktur danych	7
3.3.1	Zdjęcia	7
3.3.2	moduły do przetwarzania zdjęć	7
3.4	Wykorzystane zasoby	7
3.5	Realizowane główne funkcje systemu	7

1 Wybór tematu

Tematem niniejszego projektu jest: **"4.Poprawa jakości zdjęć we współczesnych smartfonach"**. Jest to ciekawe zagadnienie, ze względu na wykorzystanie nowoczesnych technologii stosowanych w smartfonach. Dodatkowo składa się na niego również temat obróbki zdjęć, którą można wykonać na wiele różnych sposobów.



2 Analiza wymagań

2.1 Cel aplikacji

Celem aplikacji jest wykorzystanie najnowszych rozwiązań technologicznych zaimplementowanych w aparatach smartfonów typu iPhone oraz wykorzystanie algorytmów do przetwarzania obrazu w celu poprawy jakości obrazu oraz osiągnięcia efektów zmiany stylu zdjęcia tak jak w przypadku filtrów w aplikacji Instagram. Liczba filtrów, która będzie wymagana wynosi 5.

Lista implementowanych filtrów:

- odsumianie zdjęcia na podstawie wykonanej serii zdjęć
- modyfikacja motywu kolorów zdjęcia pierwszego na podstawie innego wybranego zdjęcia
- nakładanie przedmiotów znalezionych dzięki kamerze deep sense na inne zdjęcie
- możliwość zaszumienia drugoplanowych/pierwszoplanowych scen.
- usuwanie przedmiotów znajdujących się na pierwszym planie na podstawie wykonanej serii zdjęć

2.2 Docelowa grupa użytkowników

Biorąc pod uwagę prostotę aplikacji, docelową grupą użytkowników są osoby należące do każdej grupy wiekowej. Aplikacja znalazłaby zastosowanie między innymi u fanów fotografii, ze względu na możliwość poprawy jakości zdjęć, czy u młodzieży chcącej podzielić się ze znajomymi odszumionym zdjęciem dobrej jakości na portalach społecznościowych.

2.3 Docelowa platforma

Docelową platformą jest iPhone posiadający "true depth camera", czyli każdy iPhone w wersji X oraz wszystkie nowsze wersje. Telefon powinien także posiadać appstore w celu zainstalowania aplikacji.

2.4 Środowisko działania aplikacji

Aplikacja będzie działać w środowisku swift posiłkując się biblioteką open cv oraz w przypadku konieczności wykonania trudnych obliczeń serwerem zewnętrznym.

2.5 Określenie wymagań

2.5.1 Funkcjonalne

Stworzenie aplikacji do wgrywania zdjęć i implementacja menu wyboru, ponadto możliwość przetwarzania zdjęć w wybranych wariantach

2.5.2 Wydajnościowe

Aplikacja powinna implementować optymalne algorytmy oraz przerabiać zdjęcia w czasie nie dłuższym niż dwie sekundy

2.5.3 Jakościowe

Aplikacja powinna działać płynnie i nie powinna zamykać się w niewyjaśnionych okolicznościach (zabezpieczenia względem zapytań bez odpowiedzi, klikanie kilku opcji na raz). Serwer powinien sprawnie odpowiadać na zapytania.

2.5.4 Sprzętowe

Aplikacja powinna być dostępna na najnowszych Ipadach oraz na telefonach iPhone w wersji X i wyższej.

2.5.5 Skalowalność

Aplikacja powinna być w stanie obsługiwać praktycznie nieskończoną liczbę osób. Jedynym problemem, czy też wąskim gardłem może być połączenie z serwerem (w przypadku implementacji), które będzie kolejowało zapytania. W razie większej potrzeby przepustowości serwera występuje możliwość wrzucenia serwera na platformę AWS.

2.5.6 Rozszerzalność

Implementacja nowych funkcjonalności nie powinna stanowić żadnego problemu, a nawet jest oczekiwana. Dodatkowo zmiany powinny zostać wprowadzone w przypadku nowych technologii, czy też potrzebie implementacji kreatora filtrów.

2.5.7 Otwartość systemu

Aplikacja będzie ogólnodostępna w sklepie AppStore i będzie bezpłatna. W przypadku chęci modyfikacji kodu, czy też wykorzystania jego części, kod aplikacji będzie dostępny w serwisie GitHub.

2.5.8 Niezawodność

Aplikacja powinna być dostępna praktycznie cały czas. Serwer do obliczeń powinien sprawnie odpowiadać na zapytania. Na szczęście system nie przechowuje trwale danych użytkownika więc nie trzeba robić kopii zapasowych danych, czy też używać baz danych.

2.6 Definicja interfejsów użytkownika

Użytkownik będzie posiadał możliwość zrobienia zdjęcia z podglądem czterech poprzednich edycji. Scroll niżej będzie dawał możliwość zmiany algorytmu, czy też sposobu na odszumianie. Po lewej stronie będzie znajdował się przycisk, który pozwoli użytkownikowi wybrać zdjęcie z galerii. Ponadto, z prawej strony użytkownik będzie posiadał opcję zmiany robienia zdjęcia na nagrywanie filmu. Ostatnią częścią interfejsu będzie przycisk do wykonania zdjęcia.

2.7 Opcjonalne rozwiązania funkcjonalne

Możliwość wgrywania zdjęć oraz filmów poprzez podanie linku do zasobu. Dostawienie serwera do obliczania trudnych operacji (niektórych przekształceń potrzebujących większej mocy obliczeniowej w celu zaoszczędzenia czasu).

2.8 Inne założenia poczynione z użytkownikiem

Chęć wyboru czy dane zdjęcie powinno zostać zapisane w bibliotece czy nie. Aby korzystać z aparatu, czy też zdjęć z biblioteki, użytkownik musi zaakceptować dostęp do podanych zasobów.

2.9 Odporność na awarie, bezpieczeństwo systemu

Za odporność na awarie aplikacji będzie odpowiadał sklep appstore, jedynym zmartwieniem będzie utrzymywanie serwera. W przypadku elementów losowych jak i ataków ddos, serwer będzie musiał zostać zrestartowany aby przywrócić działanie. System nie przechowuje danych, dlatego bezpieczeństwo użytkownika można uznać za bardzo wysokie.

2.10 Określenie niezbędnej dokumentacji

Dokumentacja całego kodu jak i funkcjonalności będzie udostępniona w publicznym repozytorium na platformie GitHub. Confluence nie jest zalecanym rozwiązaniem ze względu na prostotę zastosowanych rozwiązań.

2.11 Zakładane ograniczenia aplikacji

Aplikacja będzie przeznaczona tylko na systemy mobilne z systemem IOS - Android oraz Windows nie będą wspierane. Rozszerzenie aplikacji na inne platformy nie będzie możliwe ze względu na zastosowane technologie.

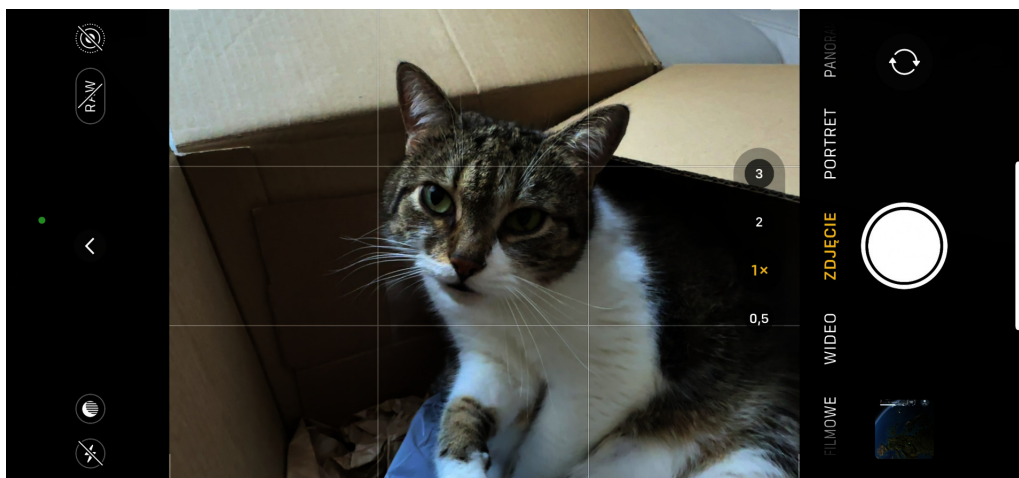
3 Projekt systemu

3.1 Architektura systemu oraz interfejs użytkownika

Interfejs użytkownika powinien być stosunkowo prosty. Będą dostępne dwa ekrany. Pierwszy będzie aparatem, który umożliwi robienie zdjęć. Będzie on pozwalał na:

- zrobienie zdjęcia
- wybranie zdjęcia z biblioteki
- zmianę typu zdjęcia

Są to proste operacje, jednak są one bardzo przydatne.



Na drugim Ekranie będzie całe centrum aplikacji. Będzie tam można:

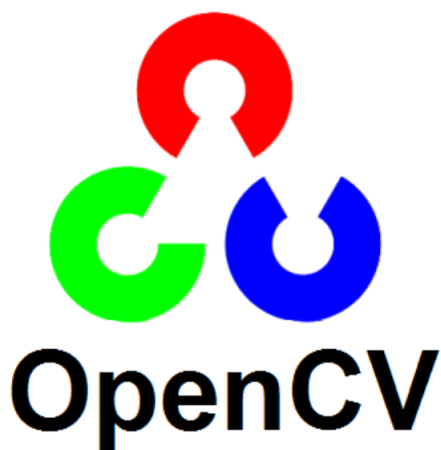
- wybrać sliderem sposób przerabiania zdjęcia
- wybrać zdjęcia zapisane w pamięci
- przejść do ekranu robienia zdjęcia
- przycisk pozwalający na rozpoczęcie przetwarzania zdjęcia

Są to główne zadania. Wszystko będzie odbywało się na telefonie więc architektura będzie stosunkowo prosta. Wszystko przekazywane będzie do poszczególnych funkcji i wywoływane od razu po naciśnięciu przycisku czy też wybraniu opcji. Więc tak naprawdę architektura będzie maksymalnie uproszczona, dzięki czemu dodanie nowej funkcjonalności powinno być stosunkowo proste.

Dodatkowo w pamięci aplikacji trzymane będą ostatnio zrobione czy też wybrane z biblioteki zdjęcia. Ich liczebność nie powinna przekraczać 4 ostatnio używanych + 1 aktualnie edytowanego. Zdjęcia te będą w oryginalnych rozdzielczościach i wszystkie parametry dla struktury pliku zdjęcia mogą zostać wykorzystane w dalszym przetwarzaniu zdjęcia.

3.2 Środowisko tworzenia aplikacji

- Aplikacja będzie tworzona w aplikacji xcode na systemy IOS dla wersji X oraz nowszych modeli, iPad nie będzie brany pod uwagę.
- Główny język programowania w którym będzie wykonywana aplikacja to SWIFT, który dzięki natywnemu przeznaczeniu na urządzenia IOS pozwoli na szybkie operacje w przeciwieństwie do zdobywającego popularność języka dart.
- Wykorzystana zostanie również biblioteka openCV, która zostanie zintegrowana z aplikacją w celu osiągnięcia możliwości szybkiego i dokładnego przetwarzania zdjęć w czasie podchodzącym pod rzeczywisty
- Aplikacja ta będzie testowana przy użyciu emulatora. Dodatkowo wersją alpha będzie udostępniona za pomocą TestFlight dla wszystkich ochotników.



3.3 Projekt najważniejszych struktur danych

W tym projekcie występują dwie ważne struktury danych i są to zdjęcia oraz moduły do przetwarzania zdjęć. Ich pola będą prezentowały się następująco

3.3.1 Zdjęcia

- nazwa
- wysokość
- szerokość
- rozszerzenie
- film czy zdjęcie

3.3.2 moduły do przetwarzania zdjęć

- nazwa
- funkcja robiąca operację na zdjęciu
- dodatkowe zmienne w zależności od potrzeby

3.4 Wykorzystane zasoby

- Biblioteka openCV
- Wbudowane funkcje aparatu
- Format zapisu pliku w języku swift
- Algorytmy potrzebne do operacji na zdjęciach:
 - Synthetic Boken
 - HDR
 - Shallow Boken

3.5 Realizowane główne funkcje systemu

Pierwsze dwie opcje zapewnione są dzięki bibliotekom zawartym w języku swift. Następne opcje natomiast muszą zostać zaimplementowane i wywoływane będą poprzez widgety zawarte w aplikacji. W przypadku wykorzystania modeli do przetwarzania zdjęć integracja na drodze C++ oraz swift może stanowić wyzwanie.

- robienie zdjęć
- ładowanie zdjęć
- przetwarzanie zdjęć
- zapisywanie ostatnich zdjęć
- wykorzystanie modelu do przetwarzania zdjęć