

wspolne_dla_wszystkich/logo_uczelni.png

Poprawa jakości skanów zdjęć wykonanych techniką analogową

Raport II

projekt realizowany pod opieką prof. dr hab. inż. Artura Przelaskowskiego

wspolne_dla_wszystkich/logo_projektu.png

Streszczenie

Raport 2 projektu poprawy jakości cyfrowych skanów zdjęć wykonanych techniką analogową przez grupę nr 9 (wtorkową z godziny 18) w składzie: Bartosz Wójcik, Katarzyna Szwed, Natalia Szymańska, Patrycja Szałajko, Aleksandra Wójcik, Karol Sęk, Michał Juskiewicz, Filip Sajko.

W tym raporcie zdefiniujemy cel naszego projektu i opiszemy problem z którym się mierzymy. Przedstawimy ponadto wstępną wersję naszego programu i zademonstrujemy jego skuteczność.

Spis treści

| | |
|--|----|
| 1. Cel projektu | 3 |
| 2. Zdjęcia, zdjęcia! | 3 |
| 3. Problemy | 3 |
| 3.1. Niedoświetlenie | 3 |
| 3.2. Zanieczyszczenia | 6 |
| 4. Program i jego działanie | 8 |
| 4.1. Tworzenie maski | 8 |
| 4.1.1. Rozmycie gaussowskie | 8 |
| 4.1.2. Wykonanie różnicy | 9 |
| 4.1.3. Usunięcie ciemnych pikseli | 9 |
| 4.1.4. Korekcja gamma | 10 |
| 4.2. Gotowa maska | 11 |
| 4.3. Działanie właściwe | 12 |
| 5. Uwagi co do działania programu | 13 |
| 6. Dostępność programu | 13 |
| 7. Wykorzystywane narzędzia | 14 |
| 8. Podział obowiązków | 14 |

1. Cel projektu

W związku ze słusznymi uwagami i wskazówkami, podjęliśmy decyzję o ukonkretyzowaniu celu naszego projektu. Skupimy się przede wszystkim na poprawianiu defektów cyfrowych skanów zdjęć analogowych. Staramy się trafić do dwóch (niekoniecznie rozłącznych) grup osób – współczesnych fanów fotografii analogowej (będącą dla amatora niełatwą sztuką) i posiadaczy pękatek archiwów zdjęć rodzinnych chcących je zachować i cyfrowo utrwalić.

2. Zdjęcia, zdjęcia!

Profilowym zdjęciem dla nas jest portret – tak indywidualny jak i grupowy. Jest to typ zdjęć najbardziej popularny w rodzinnych albumach – mnogość w nich zdjęć z ważnych dla danej rodziny wydarzeń: chrztów, wesel czy pogrzebów... Służą one utrwaleniu wspomnień oraz pamięci po krewnych i bliskich, którzy już odeszli... A więc noszących dużą wartość emocjonalną dla ich posiadacza.

Przykładem takiej osoby jest nasza koleżanka Ola – wraz z jej rodzinnym albumem.

3. Problemy

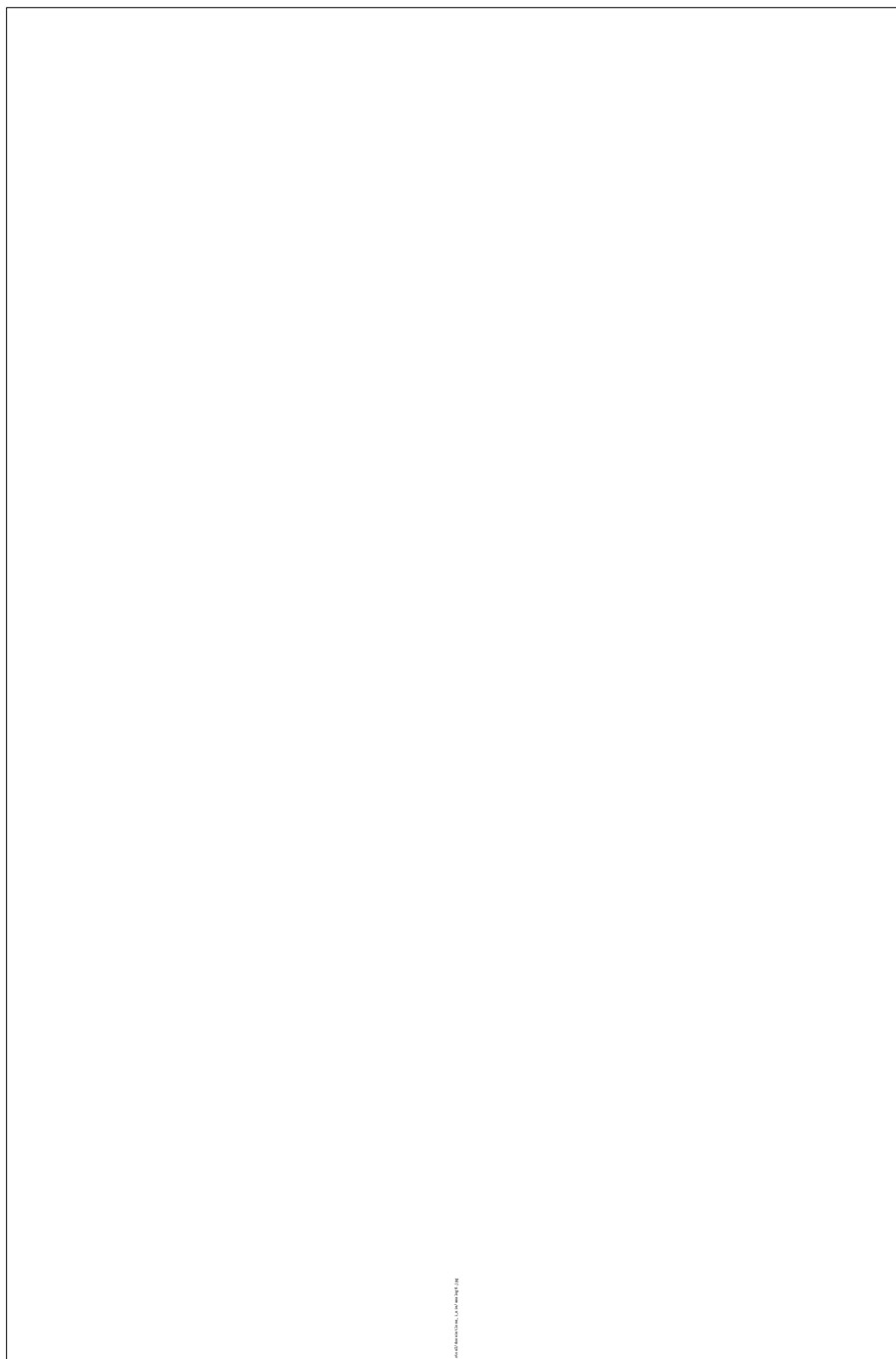
Wykonywanie, a następnie ‘ucyfrowienie’ zdjęcia w technice analogowej wiąże się z różnymi trudnościami, które mogą znacząco obniżyć jakość zdjęcia – a z tym satysfakcję jego posiadacza. Głównymi problemami, którym będziemy przeciwdziałać, będą niedoświetlenie zdjęcia i zanieczyszczenia powietrza osadzające się na oryginalnym zdjęciu i skanerze podczas procesu zmiany informacji z analogowej na cyfrową.

3.1. Niedoświetlenie

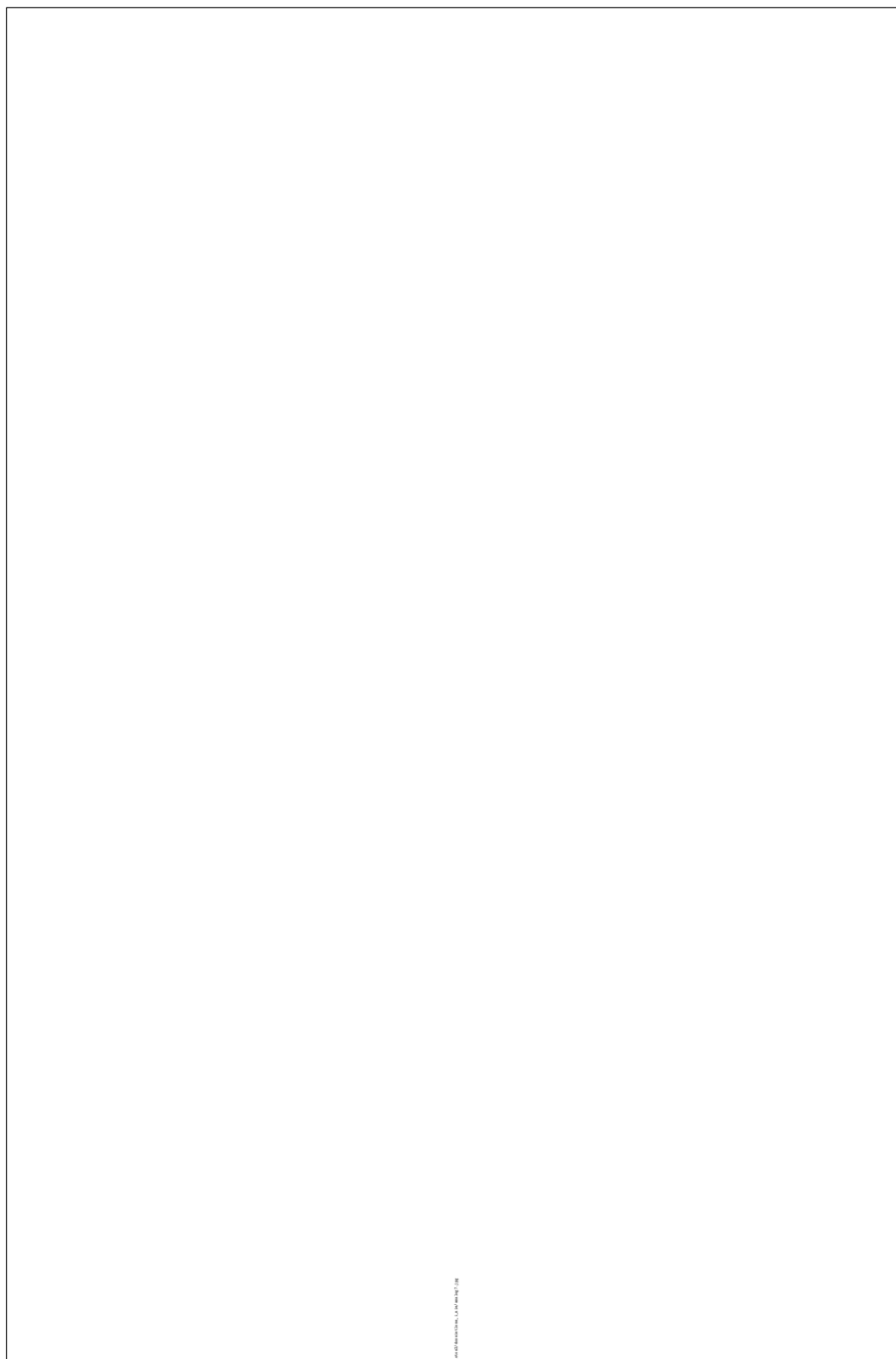
Niedoświetlenie jest problemem trudnym – zwłaszcza dla fanów-amatorów techniki analogowej. Zasadnicza większość klasycznych aparatów nie posiada zaawansowanej mechaniki automatycznie wybierającej odpowiednie ustawienia aparatu, a brak możliwości podglądu tego, jak dane zdjęcie wyszło, często doprowadza do sytuacji, gdzie po wielu dniach okazuje się, że na zdjęciu chwili, którą fotograf chciał uchwycić i utrwalić, niewiele widać, bo przez złe ustawienia większość szczegółów jest niewidoczna...¹

Dla przykładu przypomnijmy:

¹ Jest to problem, który szeroko wraz z przykładami i analizą numeryczną opisywaliśmy w raporcie pierwszym.



Rysunek 1: Zdjęcie niedoświetlone.



Rysunek 2: ...i to w punkt.

3.2. Zanieczyszczenia

Prawie że cała nasza² codzienność dzieje się w niesterylnych warunkach. I jakkolwiek dla większości z nas nie jest to problem, są miejsca i sytuacje, gdy prowadzi to do pewnych problemów. W powietrzu nas otaczającym jest pełno unoszących się zanieczyszczeń: włosów, kurzu, futra etc.

Problematyczne jest natomiast osadzanie się wspomnianej powyżej materii na zdjęciach i soczewkach – która przenosi się na skan tworząc nieestetyczne artefakty:



Rysunek 3: W skrajnych wypadkach może wyglądać to nawet tak.



Rysunek 4: Choć bardziej częstym jest ten przypadek.

² nie jesteśmy wszak ani naukowcami, ani lekarzami.

wspolne_dla_wszystkich/logo_projektu.png
Poprawa jakości zdjęć

wspolne_dla_wszystkich/logo_uczelni.png

Photos2/przed/new2.jpeg

Rysunek 5: A także taki.

4. Program i jego działanie

Zbrojni w wiedzę co chcemy osiągnąć i zapas zebranych skanów zdjęć do testów wzięliśmy się do pracy nad programem. Stworzyliśmy zaawansowany program, który w przypadku funkcjonalności ‘anty-zanieczyszczeniowej’ inteligentnie przeszukuje cały obszar zdjęcia, zaznacza artefakty, a w następnej fazie działania usuwa je.

Przechodząc do szczegółów, działanie programu można opisać za pomocą kilku kolejnych faz działania³ na przykładzie poniższego zdjęcia:



Rysunek 6: Przykładowe zdjęcie – to za jego pomocą opiszemy działanie programu.

4.1. Tworzenie maski

4.1.1. Rozmycie gaussowskie

Pierwszym krokiem generowania maski jest stworzenie rozmycia gaussowskiego zdjęcia.

Wygładzanie gaussowskie jest efektem rozmywania obrazu za pomocą funkcji Gaussa, która jest szeroko wykorzystywana w grafice komputerowej w celu uzyskania gładkiego wygładzenia obrazu i wyciszenia szumu informacyjnego.

Za pomocą funkcji danej wzorem dla każdego piksela:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (1)$$

gdzie x, y to współrzędne danego piksel’a, a σ oznacza odchylenie standardowe.⁴

³ Algorytm przekształca zdjęcie z formatu RGB do formatu HSV (Hue, Saturation, Value) i działa tylko na Value, która definiuje jasność piksela w skali 0-255.

⁴ Źródło: https://en.wikipedia.org/wiki/Gaussian_blur

Photos2/gauss_blurr/gauss_blurr_gpt1.png

Rysunek 7: Zdjęcie po wykonaniu rozmycia gaussowskiego.

4.1.2. Wykonanie różnicy

Następnie odejmujemy od oryginału zdjęcia otrzymane w poprzednim etapie rozmycia. Pozwala to na wykrycie najbardziej kontrastowych elementów zdjęcia.

Photos2/difference/difference_gpt1.png

Rysunek 8: Zdjęcie po wykonaniu różnicy.

4.1.3. Usunięcie ciemnych pikseli

Zostawiamy tylko jasne piksele (według standardowych ustawień jest to jasność powyżej 60) i nadajemy im maksymalną wartość 255. Pozostałym pikselom ustawiamy jasność na 0.

4.1.4. Korekcja gamma

Korekcja gamma jest techniką stosowaną w grafice komputerowej, której celem jest dostosowanie jasności obrazu do ludzkiego nielinowego postrzegania światła. Korekcja gamma kompensuje ten nieliniowy sposób widzenia, pozwalając na efektywniejsze wykorzystanie dostępnych poziomów jasności.

Korekcja gamma dokonuje transformacji jasności w następujący sposób:


$$L'(x, y) = L(x, y)^\gamma \quad (2)$$

gdzie:

- γ to współczynnik gamma,
- $L(x, y)$ to wartości jasności pikseli.

Wiedząc to, w tym etapie bierzemy ponownie oryginalne zdjęcie, wykonujemy na nim korekcję gamma, a następnie na tym zdjęciu wykonujemy etapy 1-3.

Działanie to pozwala uwzględnić jak najwięcej artefaktów – znajdujących się także na jasnym tle.



Photos2/gamma_corection/gamma_corection_gpt1.png

Rysunek 9: Po wykonaniu korekcji gamma i etapów 1-3.

4.2. Gotowa maska

Po tym wszystkim otrzymujemy dwie podmaski (jedna robiona na oryginalnym zdjęciu, a druga na jaśniejszym – rozświetloną modulacją gamma). Końcową maskę otrzymujemy biorąc wszystkie znalezione piksele z obu podmasek.



Rysunek 10: Maska wykonana z oganianego zdjęcia



Rysunek 11: Maska wykonana z zdjęcia rozświetlonego korekcją gamma

4.3. Działanie właściwe

Mając wygenerowaną maskę właściwą, przechodzimy po wyznaczonych przez nią pikselach na oryginalnym zdjęciu. Dla każdego piksela w zaznaczonego w masce wyliczamy nową wartość jasności – średnią z jasności wszystkich (nie zaliczają się do tego pikseli wyznaczone wcześniej przez maskę.) pikseli na odległości nie więcej 15 od aktualnie analizowanego.

Wynik finalny:



Rysunek 12: Przykładowe zdjęcie – porównanie efektu przed i po.

5. Uwagi co do działania programu

Przez to, że algorytm działa lokalnie (tylko w punktach wyznaczonych przez maskę) nie wpływa on na ogólny wygląd zdjęcia. Z tego też powodu możemy wykonywać program kilkakrotnie na zdjęciu – uzyskując lepsze efekty.

Końcowy algorytm składa się z pięciu iteracji opisanego wyżej algorytmu, znacząco zwiększając szansę na usunięcie zanieczyszczeń.

Ponadto można uzyskać dodatkowe informacje o działaniu programu i jakości zdjęcia – za przykład funkcja: `.countNoise()` zlicza ilość wyznaczonych przez maskę pikseli – które są uznane za zanieczyszczenia.

6. Dostępność programu

Na chwilę obecną nasze rozwiązanie jest programem terminalowym, działającym na systemie nie starszym niż Windows 10 – choć istnieje plan przeniesienia go także na inne popularne systemy operacyjne.

Tak samo pracujemy obecnie nad stworzeniem bardziej przystępnego interfejsu okienkowego.

Program dostępny jest na licencji *open source* i jego kod źródłowy można znaleźć na GitHubie pod adresem:

<https://github.com/ssk12o/PTI-Foto-Projekt>.

7. Wykorzystywane narzędzia

W tej części naszego projektu korzystaliśmy z następujących narzędzi:

- Programu i języka Matlab – do analizy zdjęć;
- Języka C++ – do napisania programu;
- Programu VS Code – do tworzenia, edycji i dokumentacji kodu programu i raportów;
- Programu LibreOffice Calc – do analizy części danych numerycznych;
- \LaTeX 2 ϵ – do przygotowania raportu;
- Strony Github i programu Git – do udostępniania, dystrybucji i pracy nad kodem;
- 7zip – do kompresji zdjęć;
- Google Drive – do udostępniania plików;
- Skanera minilab Noritsu HS-1800 – do wykonywania wysokiej jakości cyfrowych skanów zdjęć wykonanych techniką analogową;
- Aparatów:
 - Canon EOS 300 z obiektywem Tamron 28-105mm 1:4-5.6 i kliszą Fomapan 400
 - Fujifilm FinePix L55 Digital Camera – Black (12MP, 3x Optical Zoom)

8. Podział obowiązków

Na tym etapie projektu podzieliśmy się pracą, obowiązkami i zadaniami w następujący sposób:

- Bartosz Wójcik – wykonywanie, skanowanie i analiza zdjęć; opieka merytoryczna.
- Katarzyna Szwed – tworzenie, analizowanie i pisanie algorytmu; korekta raportu.
- Natalia Szymańska – pisanie raportu.
- Patrycja Szałajko – zarządzanie pracą zespołu, kontakt z mediami.
- Aleksandra Wójcik – skanowanie zdjęć rodzinnych w celu polepszenia ich jakości w końcowych etapach projektu.
- Karol Sęk – tworzenie, analizowanie i pisanie algorytmu.
- Michał Juskiewicz – tworzenie, analizowanie i pisanie algorytmu.
- Filip Sajko – pisanie raportu, implementacja w \LaTeX .