# Seminarium magisterskie

## Wykorzystanie cyfrowej analizy obrazu w metodykach pracy z dziećmi w wieku przedszkolnym

Jakub Lebiedziński
269096

## Wstęp

Przenieśmy się do roku 1945, właśnie dokonuje się przełom w dziedzinie cyfryzacji i komputeryzacji - pierwszy komputer ujrzał światło dzienne. Enigmatyczna nazwa ENIAC skłania nas do myślenia nad genezą tego słowa. W swojej istocie, nazwa ta, jest tak naprawdę banalna i pochodzi z języka angielskiego od 'electronic numerical integrator and computer' [1] czyli 'elektroniczny integrator numeryczny i komputer'. Robi się ciekawiej. Warto w tym miejscu nakreślić ogólne pojęcie o zautomatyzowanych procesach w 1945 roku. Mam na myśli to, jak ludzie postrzegali wówczas maszyny wykonujące skomplikowane obliczenia matematyczne w znacznie krótszym czasie niż oni sami. Ludzkość w późniejszych latach '40 ubiegłego wieku była sceptycznie nastawiona do robotów i maszyn. Głównie dlatego, że ich nie rozumieli - taka natura ludzka, że boimy się tego czego nie jesteśmy wyjaśnić w logiczny sposób. Dzisiaj tj. 80 lat później nie wyobrażamy sobie życia bez współczesnych rozwiązań techniki. Cóż, przynajmniej uważamy to za coś nierealnego, nieosiągalnego. Jak zauważa Monika Frania 'Człowiek żyjący na przełomie XX i XXI w. to świadek eksplozji przemian instrumentarium medialnego' [2] Ciężko się nie zgodzić. Nasi dziadkowie pamiętają czasy, w których technologia była im zupełnie obca. Ba, sami doświadczyliśmy okresu, w którym 'Internet' było słowem nieznanym, niezrozumiałym. Czym jest tak naprawdę ta cała 'technologia', o którą się tu rozchodzi? Według Polskiego Słownika PWN słowo 'technologia' [3] opiera się na metodach powiązanych z procesem przetwórczym lub produkcji. Technologie informacyjne jako termin są z nami stosunkowo od niedawna. Wraz ze wzrostem znaczeń maszyn i komputerów, przez gromadzenie i agregację danych, aż do 'targetowania' użytkowników wspomnianego już Internetu, na znaczeniu zyskały również inne dziedziny nauki, ale i nie tylko. Edukacja jest z nami od zarania dziejów. Już w starożytnej Grecji powstawały pierwsze 'szkoły', gdzie słuchacze mogli śledzić wykłady najznamienitszych filozofów ówczesnego świata. Platon, Arystoteles czy Sokrates byli uważani za mistrzów retoryki, etyki czy filozofii. Warto wspomnieć, że nie każdy obywatel antycznego państwa miał obowiązek pobierania nauki. Edukacja była bowiem przywilejem, na który stać było tylko najbogatszych. Dzisiaj jest to nie do pomyślenia. Z czasem jednak edukacja stawała się coraz bardziej dostępna dla niższych warstw społecznych.

Dawni nauczyciele wspomagali się w edukowaniu swoich słuchaczy wskaźnikami, elementami natury, mapami, malunkami i tak dalej. Podobnie jest i teraz. Różnicą jest stopień zaawansowania 'pomagaczy' i procent ich wykorzystania w szkolnictwie. Celem niniejszej pracy magisterskiej będzie wskazanie na rosnącą rolę technologii w kontekście edukacji w XXI wieku. Skupimy się na dzieciach w wieku przedszkolnym, ponieważ od najmłodszych lat można dostrzec schematy i utrwalanie pewnych zachowań w nauce podstawowych czynności takich jaki mówienie, pisanie czy postrzeganie świata. Józef Bednarek w swojej książce pod tytułem 'Multimedia w kształceniu' [4] , iż każdy uczeń chce się uczyć, ale to podejście pedagoga jest niezbędnym elementem, aby ta nauka 'nie poszła w las'. Porównanie ww. Autora tablicy i kredy do telewizji i gier komputerowych wydaje się trafnym porównaniem podkreślającym rolę jaką odgrywa współczesny nauczyciel. Z pomocą przychodzi już wspomniana technologia. Dzieci w wieku przedszkolnym tj. 3-6 lat są niezwykle wrażliwe na bodźce ze świata zewnętrznego. Przy oglądaniu zdjęć, obrazów, malunków ich mózg skupia się na rozpoznawaniu kształtów, kolorów, osób, a z wiekiem potrafi zauważyć analogię pewnych zestawień liter, kolorów, a nierzadko jest w stanie je nazywać. W swojej pracy wykorzystuje autorskie rozwiązanie interaktywnej kolorowanki w czasie rzeczywistym. Ów narzędzie opiera się na rozpoznawaniu kolorów, rysowaniu kształtów oraz stymuluje prawidłowy rozwój dziecka w myśl zasady - nauka poprzez zabawę. Zakresem badań będą przeprowadzenie serii eksperymentów i zapis obserwacji oraz wnioski z zachowań dzieci bawiących się wyżej opisanym narzędziem.

## Multimedia w szkolnictwie

Z biegiem lat zauważamy coraz to nowsze rozwiązania technologiczne wprowadzane począwszy od szkół podstawowych, a skończywszy do szkolnictwie wyższym. Kiedyś hitem były tablice multimedialne, które stawały się niemalże wizytówką placówki oświatowej. Tak było kilka lat temu. Rok 2020 okazał się jeszcze bardziej wymagający i zmusił szkoły, organy oświatowe a także samorządy i placówki administracyjne do zakupu wszelkiego rodzaju tabletów, laptopów, notebooków, smartfonów etc. Nagle okazało się, że podstawowy ekwipunek ucznia o wadze 11 kilogramów został zastąpiony jednym 800g tabletem, w którym znajdują się elektroniczne wersje niezbędnych do nauki podręczników i ćwiczeń. W przedszkolach również zauważa się wzrost technologii wykorzystywanej do nauki dzieci. Co prawda komputery już wcześniej były obecne w salach zabaw, a dzieci potrafiły w mniej lub bardziej logiczny sposób wyjaśnić obecność elektroniki, ale elektroniczna kolorowana to coś całkiem nowego dla całego półświatka przedszkolnego.

## Język C++

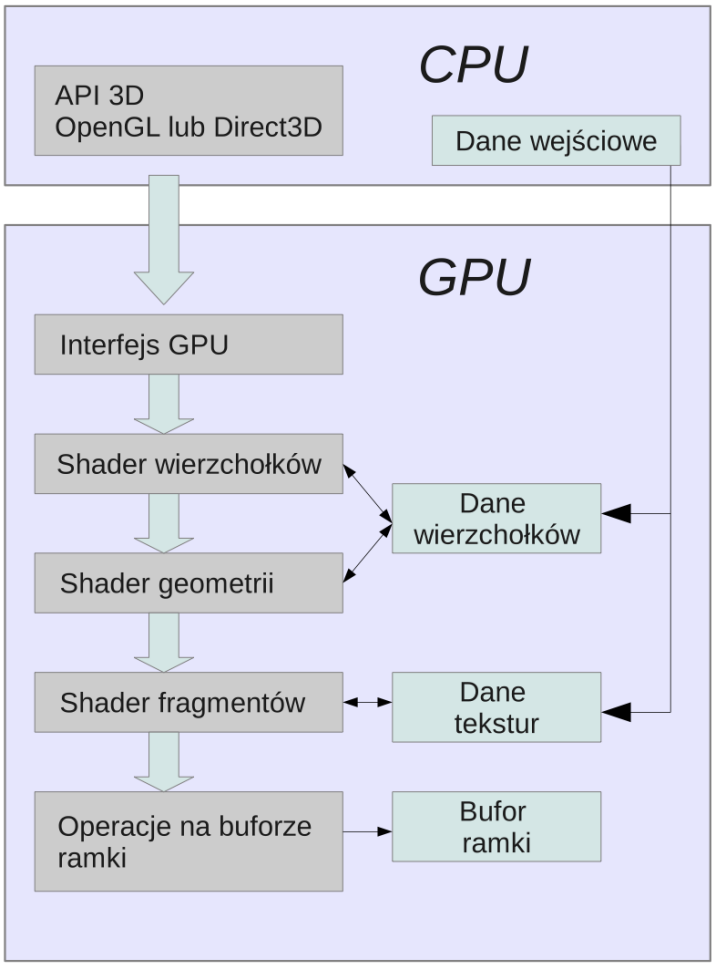
Zapewne każdy aspirujący programista musiał miał styczność z tym językiem. Prosty, niewymagający, wybaczający błędy - idealny do rozpoczęcia przygody z programowaniem. Język C++ to nie jest "związek" tylko na chwile. Jego zastosowanie można zaobserwować w takich produktach jak Windows, Mac OS, pakiet Adobe czy Office. Dlaczego? Odpowiedź jest prosta - język C++ jest bardzo wydajny i zużywa o wiele mniej zasobów przy kompilacji. Jak słusznie zauważa Karol Kuczmarski w swojej książce "Kurs C++. Od zera do gier kodera" [5] mocnym atutem języka C++ jest jego popularność i dostępność rozwiązań. Cytując jego słowa można stwierdzić, że "[...] C++ zdaje sie byc bardziej uniwersalny (od języka Delphi, przyp. tłum.). Dobrze rozumie sie z waznymi dla nas bibliotekami graficznymi, jest takze bardzo szybki i posiada duze mozliwosci" Jak to się stało, że język C++ osiągnął taką przewagę nad innymi językami? Żeby odpowiedzieć na to pytanie należy cofnąć się do początków lat siedemdziesiątych kiedy to niejaki Dennis Ritchie finalnie zaimplementował z pomocą języka C jądro systemu operacyjnego Unix. Uznaje się to wydarzenie za początek dominacji języka C (i jego późniejszych ewolucji, o których wspomnę w dalszej części). Po roku 1980 język C (a także jego późniejsza wersja z dwoma plusami) wiódł prym w programowaniu systemów i aplikacji czego dowodem był produkt Microsoftu pod nazwą Windows, przy którego budowanie w znacznej mierze opierano się na wyżej wymienionym języku programowania. Warto zaznaczyć, że język C/C++ posiadał bardzo ważną cechę jaką niewątpliwie jest możliwość przenoszenia go na inne urządzenia. Dodatkowym atutem, o którym trzeba powiedzieć jest to, że większość dzisiejszych sterowników do kart graficznych, dźwiękowych i tak dalej są pisanie niskopoziomowo z wykorzystaniem właśnie języka C/C++.

## Ewolucji ciąg dalszy czyli język C Sharp

W końcówce lat dziewięćdziesiątych zaczęto zastanawiać się na stworzeniem od podstaw języka programowania opartego na obiektowości, który mógłby rywalizować z zyskującą popularność Javą. Narodził się pomysł utworzenia zespołu projektowego, którego zadaniem byłoby stworzenie mocnego konkurenta na "rynku obiektowości" w myśl zasady głoszonej przez model PME (Properties - Methods - Events). Na czele projektu stanął charyzmatyczny Duńczyk Anders Hejlsberg - znany i ceniony programista w środowisku informatycznym. W swoim dorobku miał doświadczenie w pracy nad takimi językami jak wspomniany już wcześniej Delphi, Pascal, a później również TypeScript. W swojej książce pod tytułem "The C# Programming Language" [6] wspominał, że praca przy nowym języku była dla niego nie lada wyzwaniem, ale zaznaczył, że traktował to jako ciekawą rozrywkę, żeby nie powiedzieć zabawę. Dodał, że miarą rozwiązywania problemów jest dodanie wartości tworzonemu produktowi przy poszukiwaniu solucji.

## Programowanie GPU

Jak wspomniałem jedną z zalet języka C/C++ jest szeroki zasób i dostępność do bibliotek o bardzo różnym charakterze. Od napisania mikrokontrolera przez fotokomórkę aż po grę komputerową. W dzisiejszym świecie coraz więcej elementów otaczającego nas świata zależy właśnie od technologii. Częściej musimy zasięgnąć opinii przysłowiowych "informatyków", aby Ci wytłumaczyli nam chociażby jak kupić bilet na autobus z poziomu telefonu komórkowego. Wraz z rozwojem sektora oprogramowania graficznego na znaczeniu, w kwestii mocy obliczenio- wej, zyskały układy graficzne oparte głównie na dwóch architekturach - NVIDIA CUDA oraz OpenCL. Zauważono, że obecnie układy te pozwalają na wykonanie serii bardziej skomplikowanych operacji niż dotychczas sądzono. Nowoczesne procesory graficzne są w stanie generować żądany obraz w czasie rzeczywistym i co więcej są w stanie zmodyfikować go według określonych zasad ustanowionych przez programistę. Dzieje się tak, gdyż układ CPU komunikuje się z układem GPU z pomocą specjalnego API (Application Programming Interface) w charakterze komunikatora pomiędzy dwoma układami. Poniższy schemat powinien nieco rozjaśnić powyższe rozważania.




Dzięki takiemu rozwiązaniu powszechne stało się wykorzystanie mocy obliczeniowej pro-

cesorów graficznych w tworzeniu oprogramowania wykorzystującego funkcje przetwarzania obrazu przechwyconego z kamery. Przykładem takiego rozwiązania jest na przykład rozpoznawanie twarzy w nowoczesnych telefonach, skanowanie numerów rejestracyjnych samochodów wyjeżdżających z parkingów, algorytmy rozpoznawania elementów na obrazie w celu identyfikacji i opisania rzeczy znajdujących się przed obiektywem kamery. W nowych samochodach można znaleźć system odpowiadające za śledzenie toru, po którym porusza się pojazd, albo oprogramowanie odpowiadające za detekcję znaków drogowych i stosowanie odpowiednich ograniczeń wbudowanych w oprogramowanie komputera pokładowego. Można by wymieniać i wymieniać, ale żeby "namacalnie" (a przynajmniej w teorii) móc poczuć wcześniej wspomniany skok technologiczny warto chociażby zdobyć i porównać zdjęcia dwóch gier wideo, których daty premier oddalone są na osi czasu o raptem 20 lat. W ciągu tak krótkiego czasu zrobiono więcej niż przez ostatnie 50 lat w kwestii motoryzacji (chociaż Elon Musk każe sądzić inaczej). Jestem bardzo ciekaw jak będzie wyglądać rozgrywka za drugie tyle lat. Prawdopodobnie niemożliwym zadaniem będzie odróżnienie gry od rzeczywistości. Sekretem realistycznej grafiki w grach komputerowych jest skomplikowanie algorytmów odpowiedzialnych za renderowanie w czasie rzeczywistym tekstur w bardzo wysokiej rozdzielczości. Swoim artykule Christoph Liedtke z redakcji GameStar (gamestar.de) [7] zwraca szczególną uwagę na nowe efekty graficzne, które pojawiły się wraz z rozwojem mocy obliczeniowej kart graficznych "Również efekty graficzne takie jak np. mapowanie wypukłości (bump mapping), multiteksturowanie czy też oparta na rzeczywistych obrazach technika fotogrametrii, a także pakiety tekstur o wysokiej jakości tworzone przez fanów, sprawiają że pod względem graficznym gry stają się coraz bardziej imponujące."


## Cyfrowa analiza obrazu

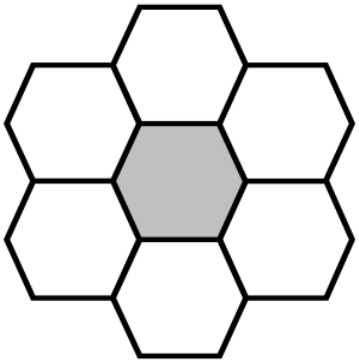
Aby wyjaśnić sposób działania algorytmów odpowiedzialnych za analizę fotografii czy obrazu z aparatu/kamery trzeba nieco cofnąć się w czasie do klasycznej fotografii. W czasach, gdy ludzie chcieli upamiętnić ważną chwilę trudno szukać jako takiej technologii znanej nam

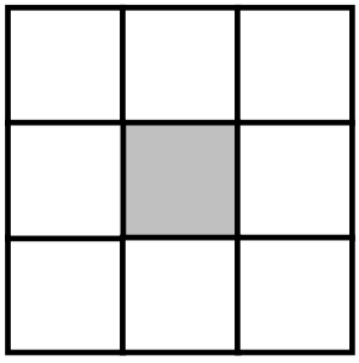
dzisiaj. Czy można powiedzieć, że wraz z rozwojem fotografii cyfrowej, wymyślono aparat na nowo?. O tym za chwilę. Na ten moment zostańmy jeszcze chwilę w tematach, jak to się dziś ładnie mówi - retro. Klasyczne, żeby nie powiedzieć stare, aparaty fotograficzne działają w oparciu o nośniki wykorzystujące materiały światłoczułe (np. klisze). Trzeba zaznaczyć, że klasyczny nie jest równy analogowy. W rozumieniu analogowy mamy na myśli te, które nie wykorzystują sygnału cyfrowego do zapisu obrazu na materiale światłoczułym. Błędne określenie aparatów analogowych klasycznymi pojawiło się wraz z rozwojem fotografii cyfrowej. Trzeba z tego zapamiętać tyle, że w dawnych aparatach klasycznych nie stosowano zapisu obrazu za pomocą sygnału analogowego. Tymczasem świat poszedł naprzód, a to co najważniejsze, ludzie zaczęli "zapamiętywać" swoje życie na cyfrowych matrycach nowoczesnych aparatów. Cyfrowy zapis od analogowego różni się tym, że w tym pierwszym fotografia utrwalana jest binarnie na matrycach cyfrowych, które po błyskawicznym "przetłumaczeniu" oddaje obraz w różnych barwach reprezentowanych przez ciąg zero-jedynkowy. Każdy taki obraz składa się z setek, tysięcy pikseli. Każdemu pikselowi odpowiada konkretny obszar i zapis na ww. cyfrowej matrycy. Innymi słowy, działanie aparatu takie same, ale zapis już niekoniecznie. Mając zapis binarny zdjęcia możemy dzięki algorytmom zastosować szereg operacji mających na celu np. rozpoznanie tekstu, detekcję ruchu, zliczanie elementów czy wspomnianą już wcześniej asystę dla kierowcy przy zjeździe z pasa na jezdni. Maszyna mając zero-jedynkowe przedstawienie fotografii widzi macierz liczb, która zawiera reprezentację barw.

| | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 67 | 67 | 66 | 68 | 66 | 67 | 64 | 65 | 65 | 63 | 63 | 69 | 61 | 64 | 63 | 66 | 61 | 60 |
| 69 | 68 | 63 | 68 | 65 | 62 | 65 | 61 | 50 | 26 | 32 | 65 | 61 | 67 | 64 | 65 | 66 | 63 |
| 72 | 71 | 70 | 87 | 67 | 60 | 28 | 21 | 17 | 18 | 13 | 15 | 20 | 59 | 61 | 65 | 66 | 64 |
| 75 | 73 | 76 | 78 | 67 | 26 | 20 | 19 | 16 | 18 | 16 | 13 | 18 | 21 | 50 | 61 | 69 | 70 |
| 74 | 75 | 78 | 74 | 39 | 31 | 31 | 30 | 46 | 37 | 69 | 66 | 64 | 43 | 18 | 63 | 69 | 60 |
| 73 | 75 | 77 | 64 | 41 | 20 | 18 | 22 | 63 | 92 | 99 | 88 | 78 | 73 | 39 | 40 | 59 | 65 |
| 74 | 75 | 71 | 42 | 19 | 12 | 14 | 28 | 79 | 102 | 107 | 96 | 87 | 79 | 57 | 29 | 68 | 66 |
| 75 | 75 | 66 | 43 | 12 | 11 | 16 | 62 | 87 | 84 | 84 | 108 | 83 | 84 | 59 | 39 | 70 | 66 |
| 76 | 74 | 49 | 42 | 37 | 10 | 34 | 78 | 90 | 99 | 68 | 94 | 97 | 51 | 40 | 69 | 72 | 65 |
| 76 | 63 | 40 | 57 | 123 | 88 | 60 | 83 | 95 | 88 | 80 | 71 | 67 | 69 | 32 | 67 | 73 | 73 |
| 78 | 50 | 32 | 33 | 90 | 121 | 66 | 86 | 100 | 116 | 87 | 85 | 80 | 74 | 71 | 56 | 58 | 48 |
| 80 | 40 | 33 | 16 | 63 | 107 | 57 | 86 | 103 | 113 | 113 | 104 | 94 | 86 | 77 | 48 | 47 | 45 |
| 88 | 41 | 35 | 10 | 15 | 94 | 67 | 96 | 98 | 91 | 86 | 105 | 81 | 77 | 71 | 35 | 45 | 47 |
| 87 | 51 | 35 | 15 | 15 | 17 | 51 | 92 | 104 | 101 | 72 | 74 | 87 | 100 | 27 | 31 | 44 | 46 |
| 86 | 42 | 47 | 11 | 13 | 16 | 71 | 76 | 89 | 95 | 116 | 91 | 67 | 87 | 12 | 25 | 43 | 51 |
| 96 | 67 | 20 | 12 | 17 | 17 | 86 | 89 | 90 | 101 | 96 | 89 | 62 | 13 | 11 | 19 | 40 | 51 |
| 99 | 88 | 19 | 15 | 15 | 18 | 32 | 107 | 99 | 86 | 95 | 92 | 26 | 13 | 13 | 16 | 49 | 52 |
| 99 | 77 | 16 | 14 | 14 | 16 | 35 | 115 | 111 | 109 | 91 | 79 | 17 | 16 | 13 | 46 | 48 | 51 |

Jest to jedna ze struktur obrazów cyfrowych, które "rozumie" maszyna. W głębszym rozu-

mieniu, pozyskiwanie obrazów cyfrowych i translację ich na język komputerowy nazywamy dyskretyzacją obrazu. Ryszard Tadeusiewicz i Przemysław Korohoda w "Komputerowej analizie i przetwarzanie obrazów" [8] zaznaczają, że ów sztuczna reprezentacja i sposób jej kreowania posiada swoje ograniczenia wynikające z wydajności dzisiejszych maszyn. Niestety (a może i stety) nie osiągnęliśmy jeszcze poziomu maszyn, które można byłoby określić szybszymi i mądrzejszymi niż zmysły ludzkie (w tym wypadku mam na myśli zmysł wzroku). Na ten moment uważa się, że przybliżone przetwarzanie danych w czasie rzeczywistym przez ludzkie oko plasuje się na poziomie około stu megabajtów na sekundę co przekracza możliwości komputerów w znanym nam świecie. Być może, a raczej wielce prawdopodobne, jest to, że maszyna oparta o technologię kwantową sprostałaby wyżej wymienionym wymaganiom, ale na tę chwilę musimy te rewelacje odsunąć na bok i skupić się nad dostępnością dzisiejszych rozwiązań. Co możemy zrobić, aby ograniczyć reprezentację obrazu? Ryszard Tadeusiewicz i Przemysław Korohoda wymieniają między innymi możliwość ograniczenia fotografii poprzez zmniejszenie ilości szczegółów, ale też uproszczenie i ujednolicenie stanów elementów (na przykład poprzez wykorzystanie czarno-białej palety barw). Autorzy sugerują również analizowanie obrazu płaskiego zamiast przestrzennego i statycznego zamiast dynamicznego. Ten ostatni przypadek odnosi się do ciągu klatek (obrazów) dlatego w dalszej części nie będę go brał pod uwagę. Warto mieć to jednak na uwadze. Dzisiejsze algorytmy przetwarzające fotografie wykorzystują jeden z dwóch sposobów umieszczenia cyfrowych odpowiedników elementów obrazu (pikseli). Są to: heksagonalne i kwadratowe.





Wspomniana macierz reprezentująca barwy, a dokładniej nasycenie jednej z trzech bar modelu RGB (Red - Green - Blue), służy komputerowi jako źródło kilku istotnych danych. Maszyna rozpoznaje nasycenie, barwę oraz potrafi rozpoznać co znajduje się na pierwszym, drugim planie i tak dalej. Niemniej jednak to wciąż za mało, aby być pewnym otrzymanych wyników. Na dane wejściowe nierzadko stosuje się szereg zabiegów mających na celu zniwelowanie niedoskonałości powstałych przy wykonywaniu fotografii. Zanim jednak zajmiemy się wyżej wymienionymi operacjami należy sobie wyjaśnić z czego biorą się błędny przy pracy ze źle przygotowanymi danymi wejściowymi. Niestety czynnik ludzki ma tutaj ogromne

znaczenie dla wysokiej wartości tych danych. Często nie przywiązujemy należytej uwagi do jakości aparatów i ich podzespołów co prowadzi od początku do niewystarczającej ostrości, naświetlenia i tym podobne. Również pośpiech nie jest dobrym doradcą w tych sprawach. Ważną cechą dobrego zdjęcia jest odpowiednie światło i ostrość - co w przypadku obiektów poruszających się jest niezwykle utrudnione, żeby nie powiedzieć niemożliwe.

## Operacje na fotografiach

Usuwanie szumu z fotografii nie jest czymś nowym. Zasada działania algorytmów "odszumiających" jest całkiem prosta w wytłumaczeniu. Jak wcześniej wspomniałem, komputer widzi obraz jako macierz z setkami małych punktów (pikselami). Filtr odpowiedzialny za usuwanie szumu sprawdza każdy element fotografii i jej bliskie sąsiedztwo analizując wartości pikseli. Założeniem jest to, że każdy punkt o przybliżonej barwie, naświetleniu powinien mieć podobne wartości, a każde odchylenie od tej reguły traktowane jest jako niepożądane zachowanie i modyfikowane w zależności od parametrów "dobrych pikseli". Metody usuwania szumu są dwie - poprzez średnią arytmetyczną i medianę. W przypadku pierwszej filtr nadpisuje analizowany obszar wartościami średnimi obliczonymi na podstawie barwy i ostrości. W drugiej metodzie piksel otrzymuję medianę z otaczającego go sąsiedztwa co czyni obraz bardziej wyrazistym, ale może powodować skrzywienia i wpływać negatywnie na geometrię obrazu.

```
VideoCapture cap(0);

//sprawdzenie dostepu do kamery urzadzenia
if ( !cap.isOpened() )
{
  cout << "Cannot open the web cam" << endl;
  return -1;
}

//uruchomienie okna z przechwyconym obrazem z kamery
namedWindow("Control", WINDOW_AUTOSIZE);
```

Listing 1: Początek wywołania

Powyższy fragment kodu jest pierwszymi linijkami głównej metody wywołującej o nazwie main. Na początku sprawdzam czy mamy dostęp do kamery urządzenia. Następnie tworzę okno o automatycznych rozmiarach (skalowane w zależności od ustawień komputera, na którym uruchamiany jest kod).

```
if (dArea > 10000) {
  int posX = dM10 / dArea;
  int posY = dM01 / dArea;

  if (iLastX >= 0 && iLastY >= 0 && posX >= 0 && posY >= 0) {
    if (waitKey(60) == 113) {
      line(imgLines, Point(posX, posY), Point(iLastX, iLastY), Scalar(0,0,255), 15);
    }
    if (waitKey(60) == 119) {
      line(imgLines, Point(posX, posY), Point(iLastX, iLastY), Scalar(0,255,0), 15);
    }
    else if (waitKey(60) == 101)
    {
      line(imgLines, Point(posX, posY), Point(iLastX, iLastY), Scalar(0,0,0), 15);
    }
  }
  iLastX = posX;
  iLastY = posY;
}
```

Listing 2: Klawisze funkcyjne

Powyżej zaimplementowane jest reagowanie programu na naciśnięte klawisze klawiatury komputera. Każdy przycisk zakodowany jest w formacie liczbowej i ma swoje odzwierciedlenie w tablicy znaków ASCII

| Decimal | Hexadecimal | Binary | Octal | Char |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | *[NULL]* |
| 1 | 1 | 1 | 1 | *[START OF HEADING]* |
| 2 | 2 | 10 | 2 | *[START OF TEXT]* |
| 3 | 3 | 11 | 3 | *[END OF TEXT]* |
| 4 | 4 | 100 | 4 | *[END OF TRANSMISSION]* |
| 5 | 5 | 101 | 5 | *[ENQUIRY]* |
| 6 | 6 | 110 | 6 | *[ACKNOWLEDGE]* |
| 7 | 7 | 111 | 7 | *[BELL]* |
| 8 | 8 | 1000 | 10 | *[BACKSPACE]* |
| 9 | 9 | 1001 | 11 | *[HORIZONTAL TAB]* |
| 10 | A | 1010 | 12 | *[LINE FEED]* |
| 11 | B | 1011 | 13 | *[VERTICAL TAB]* |
| 12 | C | 1100 | 14 | *[FORM FEED]* |
| 13 | D | 1101 | 15 | *[CARRIAGE RETURN]* |
| 14 | E | 1110 | 16 | *[SHIFT OUT]* |
| 15 | F | 1111 | 17 | *[SHIFT IN]* |
| 16 | 10 | 10000 | 20 | *[DATA LINK ESCAPE]* |
| 17 | 11 | 10001 | 21 | *[DEVICE CONTROL 1]* |
| 18 | 12 | 10010 | 22 | *[DEVICE CONTROL 2]* |
| 19 | 13 | 10011 | 23 | *[DEVICE CONTROL 3]* |
| 20 | 14 | 10100 | 24 | *[DEVICE CONTROL 4]* |
| 21 | 15 | 10101 | 25 | *[NEGATIVE ACKNOWLEDGE]* |
| 22 | 16 | 10110 | 26 | *[SYNCHRONOUS IDLE]* |
| 23 | 17 | 10111 | 27 | *[ENG OF TRANS. BLOCK]* |
| 24 | 18 | 11000 | 30 | *[CANCEL]* |
| 25 | 19 | 11001 | 31 | *[END OF MEDIUM]* |
| 26 | 1A | 11010 | 32 | *[SUBSTITUTE]* |
| 27 | 1B | 11011 | 33 | *[ESCAPE]* |
| 28 | 1C | 11100 | 34 | *[FILE SEPARATOR]* |
| 29 | 1D | 11101 | 35 | *[GROUP SEPARATOR]* |
| 30 | 1E | 11110 | 36 | *[RECORD SEPARATOR]* |
| 31 | 1F | 11111 | 37 | *[UNIT SEPARATOR]* |
| 32 | 20 | 100000 | 40 | *[SPACE]* |
| 33 | 21 | 100001 | 41 | **!** |
| 34 | 22 | 100010 | 42 | **"** |
| 35 | 23 | 100011 | 43 | **#** |
| 36 | 24 | 100100 | 44 | **$** |
| 37 | 25 | 100101 | 45 | **%** |
| 38 | 26 | 100110 | 46 | **&** |
| 39 | 27 | 100111 | 47 | **'** |
| 40 | 28 | 101000 | 50 | **(** |
| 41 | 29 | 101001 | 51 | **)** |
| 42 | 2A | 101010 | 52 | ***** |
| 43 | 2B | 101011 | 53 | **+** |
| 44 | 2C | 101100 | 54 | **,** |
| 45 | 2D | 101101 | 55 | **-** |
| 46 | 2E | 101110 | 56 | **.** |
| 47 | 2F | 101111 | 57 | **/** |
| 48 | 30 | 110000 | 60 | **0** |
| 49 | 31 | 110001 | 61 | **1** |
| 50 | 32 | 110010 | 62 | **2** |
| 51 | 33 | 110011 | 63 | **3** |
| 52 | 34 | 110100 | 64 | **4** |
| 53 | 35 | 110101 | 65 | **5** |
| 54 | 36 | 110110 | 66 | **6** |
| 55 | 37 | 110111 | 67 | **7** |
| 56 | 38 | 111000 | 70 | **8** |
| 57 | 39 | 111001 | 71 | **9** |
| 58 | 3A | 111010 | 72 | **:** |
| 59 | 3B | 111011 | 73 | **;** |
| 60 | 3C | 111100 | 74 | **<** |
| 61 | 3D | 111101 | 75 | **=** |
| 62 | 3E | 111110 | 76 | **>** |
| 63 | 3F | 111111 | 77 | **?** |
| 64 | 40 | 1000000 | 100 | **@** |
| 65 | 41 | 1000001 | 101 | **A** |
| 66 | 42 | 1000010 | 102 | **B** |
| 67 | 43 | 1000011 | 103 | **C** |
| 68 | 44 | 1000100 | 104 | **D** |
| 69 | 45 | 1000101 | 105 | **E** |
| 70 | 46 | 1000110 | 106 | **F** |
| 71 | 47 | 1000111 | 107 | **G** |
| 72 | 48 | 1001000 | 110 | **H** |
| 73 | 49 | 1001001 | 111 | **I** |
| 74 | 4A | 1001010 | 112 | **J** |
| 75 | 4B | 1001011 | 113 | **K** |
| 76 | 4C | 1001100 | 114 | **L** |
| 77 | 4D | 1001101 | 115 | **M** |
| 78 | 4E | 1001110 | 116 | **N** |
| 79 | 4F | 1001111 | 117 | **O** |
| 80 | 50 | 1010000 | 120 | **P** |
| 81 | 51 | 1010001 | 121 | **Q** |
| 82 | 52 | 1010010 | 122 | **R** |
| 83 | 53 | 1010011 | 123 | **S** |
| 84 | 54 | 1010100 | 124 | **T** |
| 85 | 55 | 1010101 | 125 | **U** |
| 86 | 56 | 1010110 | 126 | **V** |
| 87 | 57 | 1010111 | 127 | **W** |
| 88 | 58 | 1011000 | 130 | **X** |
| 89 | 59 | 1011001 | 131 | **Y** |
| 90 | 5A | 1011010 | 132 | **Z** |
| 91 | 5B | 1011011 | 133 | **[** |
| 92 | 5C | 1011100 | 134 | **** |
| 93 | 5D | 1011101 | 135 | **]** |
| 94 | 5E | 1011110 | 136 | **^** |
| 95 | 5F | 1011111 | 137 | **_** |
| 96 | 60 | 1100000 | 140 | **`** |
| 97 | 61 | 1100001 | 141 | **a** |
| 98 | 62 | 1100010 | 142 | **b** |
| 99 | 63 | 1100011 | 143 | **c** |
| 100 | 64 | 1100100 | 144 | **d** |
| 101 | 65 | 1100101 | 145 | **e** |
| 102 | 66 | 1100110 | 146 | **f** |
| 103 | 67 | 1100111 | 147 | **g** |
| 104 | 68 | 1101000 | 150 | **h** |
| 105 | 69 | 1101001 | 151 | **i** |
| 106 | 6A | 1101010 | 152 | **j** |
| 107 | 6B | 1101011 | 153 | **k** |
| 108 | 6C | 1101100 | 154 | **l** |
| 109 | 6D | 1101101 | 155 | **m** |
| 110 | 6E | 1101110 | 156 | **n** |
| 111 | 6F | 1101111 | 157 | **o** |
| 112 | 70 | 1110000 | 160 | **p** |
| 113 | 71 | 1110001 | 161 | **q** |
| 114 | 72 | 1110010 | 162 | **r** |
| 115 | 73 | 1110011 | 163 | **s** |
| 116 | 74 | 1110100 | 164 | **t** |
| 117 | 75 | 1110101 | 165 | **u** |
| 118 | 76 | 1110110 | 166 | **v** |
| 119 | 77 | 1110111 | 167 | **w** |
| 120 | 78 | 1111000 | 170 | **x** |
| 121 | 79 | 1111001 | 171 | **y** |
| 122 | 7A | 1111010 | 172 | **z** |
| 123 | 7B | 1111011 | 173 | **{** |
| 124 | 7C | 1111100 | 174 | **\|** |
| 125 | 7D | 1111101 | 175 | **}** |
| 126 | 7E | 1111110 | 176 | **~** |
| 127 | 7F | 1111111 | 177 | *[DEL]* |

Figure 1: Tablica znaków ASCII

## Fragmenty kodu oprogramowania z wyjaśnieniami

```
1  //Funkcja odpowiedzialna za nalozenie na plotno zadanego obrazu
2  void overlayImage(const cv::Mat &background, const cv::Mat &foreground,
3  cv::Mat &output, cv::Point2i location)
4  {
5    background.copyTo(output);
6    for(int y = std::max(location.y , 0); y < background.rows; ++y)
7    {
8      int fY = y - location.y;
9      if(fY >= foreground.rows)
10     break;
11     for(int x = std::max(location.x, 0); x < background.cols; ++x)
12     {
13       int fX = x - location.x;
14       if(fX >= foreground.cols)
15       break;
16       double opacity =
17       ((double)foreground.data[fY * foreground.step + fX * foreground.channels() + 3])
18
19       / 255.;
20       for(int c = 0; opacity > 0 && c < output.channels(); ++c)
21       {
22         unsigned char foregroundPx =
23         foreground.data[fY * foreground.step + fX * foreground.channels() + c];
24         unsigned char backgroundPx =
25         background.data[y * background.step + x * background.channels() + c];
26         output.data[y*output.step + output.channels()*x + c] =
27         backgroundPx * (1.-opacity) + foregroundPx * opacity;
28       }
29     }
30   }
31 }
```

Listing 3: Nakładanie obrazu do kolorowania

Funkcja overlayImage z argumentami: background, foreground, output i location jest odpowiedzialna za nałożenie na przechwycony obraz z kamery durgiej warstwy - w tym wypadku jest to fragment obrazka do pokolorowania (same kontury). Funkcja w linijce szóstej rozpoczyna pętle, której zadaniem jest w osiach x i y przychodzących macierzy background i foreground obliczenie i ustawienie w środku generowanego okna obrazu. W linijce 11 zaczyna się sprawdzenie dla osi x po uprzedmim sprawdzeniu osi y. Następnie po ustawniu punktów lokalizacyjnych, w linijce 20 nanoszone są zmiany na ekran poprzez przypisanie mapie nazwie data, która jest elementem zarowno macierzy background jak i foreground.

## Fragmenty kodu

1 Początek wywołania . . . . . 9
2 Klawisze funkcyjne . . . . . 9
3 Nakładanie obrazu do kolorowania . . . . . 11

# Bibliografia


[1] Wikipedia.org

[2] Wybrane dylematy współczesnej edukacji w kontekście "zmediatyzowanej rzeczywistości" - Frania M.

[3] Polski Słownik PWN

[4] Multimedia w kształceniu - Bednarek J.

[5] Karol Kuczmarski - Kurs C++. Od zera do gier kodera

[6] Anders Hejlsberg - The C# Programming Language

[7] Christoph Liedtke - Ewolucja grafiki 3D w grach komputerowych

[8] Ryszard Tadeusiewicz i Przemysław Korohoda - Komputerowa analiza i przetwarzanie obrazów

[9] Nowe media, technologie i trendy w edukacji - Frania M.

[10] Nowe technologie informacyjne w edukacji - Adamkiewicz J.

[11] Multimedia w kształceniu - Bednarek J.

[12] Komputer jako środek dydaktyczny w edukacji wczesnoszkolnej - Hassa A.

[13] Media w edukacji - Gajda J.

[14] Słownik terminów i pojęć badań jakościowych nad edukacją - Jagieła J.