

Focal plane merging of photos with moving object

Błażej Mrzygłód ^{1*}, Mirosław Kordos ²

¹ *University of Bielsko-Biala, Poland, s58012@student.ubb.edu.pl*

² *University of Bielsko-Biala, Poland, mkordos@ubb.edu.pl*

**corresponding author*

Abstract: The paper presents our method of focal plane merging of photos containing moving objects. To achieve this goal, we present a solution that combines static focus bracketing with object segmentation performed using artificial intelligence methods, especially convolutional neural networks. These models were used to detect objects that can move and as a result introduce artefacts in the final photo. In the proposed solution, these objects are compared between shots and the object with the best sharpness is selected in its entirety for the final photo.

Keywords: focal plane merging; convolutional neural network; object segmentation

Składanie ostrości zdjęć z ruchomymi obiektami

Błażej Mrzygłód ^{1*}, Mirosław Kordos ²

¹ *Uniwersytet Bielsko-Bialski, Polska, s58012@student.ubb.edu.pl*

² *Uniwersytet Bielsko-Bialski, Polska, mkordos@ubb.edu.pl*

**corresponding author*

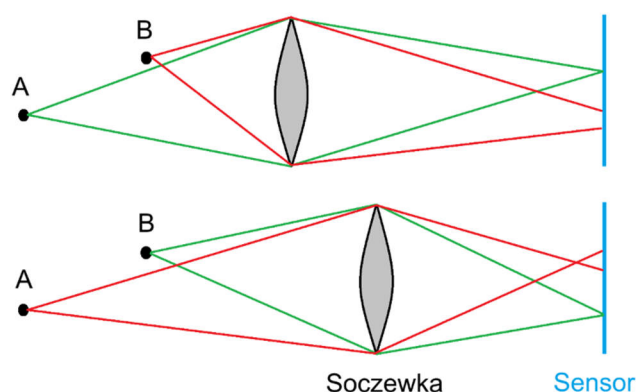
Streszczenie: Praca przedstawia opracowaną przez nas metodę składania ostrości zdjęć na których występują obiekty ruchome. Aby osiągnąć ten cel prezentujemy rozwiązanie łączące statyczne składanie ostrości z segmentacją obiektów wykonywaną przy pomocy metod sztucznej inteligencji, a zwłaszcza konwolucyjnych sieci neuronowych. Modele te wykorzystane zostały do wykrycia obiektów, które mogą się poruszać i skutkiem tego wprowadzać artefakty na zdjęciu końcowym. W proponowanym rozwiązaniu obiekty te są porównywane między ujęciami i obiekt o najlepszej ostrości jest w całości wybierany do zdjęcia końcowego.

Słowa kluczowe: składanie ostrości zdjęć; konwolucyjna sieć neuronowa; segmentacja obiektów

1. Wstęp

Na rys. 1 przedstawiona jest idea. składania ostrości zdjęć. Na górnej części rysunku jest większa odległość soczewki od sensora, dzięki temu punkt A będący dalej od kamery został odwzorowany z pełną ostrością na sensorze (przebiegi promieni światła pokazane na zielono) zaś punkt B, będący bliżej kamery jest rozmyty czyli nieostry (przebiegi promieni światła pokazane na czerwono). Aby uchwycić ostry punkt B należy przybliżyć soczewkę do sensora, jak na dolnej części rysunku, jednakże powoduje to rozmycie na sensorze punktu A. Zmniejszenie przesłony, czyli otworu przez który przechodzą promienie światła padające na soczewkę zmniejsza wprawdzie to rozmycie, gdyż promienie przechodzą tylko przez środkową część soczewki. Jednak zbytne zmniejszanie przesłony także powoduje zmniejszenie ostrości oraz inne zakłócenia, gdyż ze zmniejszaniem przesłony nasila się zjawisko defrakcji, a do sensora dociera coraz mniej światła. Skutecznym sposobem, aby uzyskać pełną ostrość obu punktów A i B jest wykonanie dwóch zdjęć, każdego z innym ustawieniem soczewki, a następnie uzyskanie zdjęcia wynikowego poprzez wybór tej części każdego

ze zdjęć składowych, która jest ostra. W bardziej złożonych zdjęciach trzeba często złożyć zdjęcie wynikowe z więcej, niż dwóch ujęć, a następnie z każdego z nich wybrać te fragmenty, które są na nim ostrzejsze niż na pozostałych ujęciach [1-3].



Rysunek 1. Zasada składania ostrości z kilku ujęć.

Technika ta sprawdza się bardzo dobrze w warunkach kontrolowanych, gdy na fotografii znajdują się tylko nieruchome obiekty, których pozycja nie ulega zmianie między poszczególnymi ujęciami, kamera jest nieruchoma, a oświetlenie stałe. Takich warunków nie da się jednak zagwarantować wykonując zdjęcia osób czy innych ruchomych obiektów, np. liści drzew poruszanych wiatrem. Wynika to głównie z dwóch przyczyn. Po pierwsze, jeśli na jednym ujęciu ustawimy ostrość na jakąś osobę, a na drugim na tło i w międzyczasie ta osoba się przemieści, to obrys tej osoby z pierwszego ujęcia nie będzie już pasował do obrysu tła z drugiego ujęcia. W efekcie uzyskamy nieostre fragmenty zdjęcia. Po drugie jeśli na obu ujęciach ustawimy ostrość na osobę i ona się przemieści między tymi ujęciami, to na składowym zdjęciu otrzymamy nakładające się na siebie dwie postacie nieco przesunięte względem siebie, czyli efekt nie istniejącej rzeczywistości.

Celem tej pracy jest rozwiązanie problemu składania ostrości zdjęć na których występują obiekty ruchome. Aby osiągnąć ten cel prezentujemy rozwiązanie łączące statyczne składanie ostrości z segmentacją obiektów wykonywaną przy pomocy metod sztucznej inteligencji, a zwłaszcza konwolucyjnych sieci neuronowych. Metody te wykorzystane zostały do wykrycia obiektów, które mogą się poruszać i skutkiem tego wprowadzać opisane powyżej błędy na zdjęciu końcowym. W proponowanym rozwiązaniu obiekty te są porównywane między ujęciami i obiekt o najlepszej ostrości jest w całości wybierany do zdjęcia końcowego.

Programy oferujące opcję składania ostrości można podzielić na dwie grupy: programy do ogólnej edycji zdjęć posiadające taką funkcję oraz programy stworzone specjalnie do składania ostrości. Do pierwszej grupy należą programy płatne jak Photoshop, ON1 Phoro Raw, Luminar Neo, Affinity Photo oraz darmowe jak ImageJ, Chasys Draw IES. Do grupy programów wyspecjalizowanych do składania ostrości należą programy płatne, jak najbardziej znany i przez wielu uważany jako standard Helicon Focus, ponadto Zerene Stacker oraz programy darmowe jak CombineZP i TuFuse.

Jednakże żadem z tych programów nie daje pożądanego efektów, gdy na zdjęciu pojawiają się obiekty dynamiczne. Przykładowo Helicon Focus zauważa często, że problem ten wystąpił, jednak nie umie go rozwiązać i każe wówczas użytkownikowi ręcznie wybrać odpowiednie fragmenty z poszczególnych zdjęć. Natomiast celem rozwiązania zaprezentowanego w niniejszej pracy jest pełne zautomatyzowanie składania ostrości dla zdjęć z ruchomymi, dynamicznymi obiektami.

2. Proponowana metoda

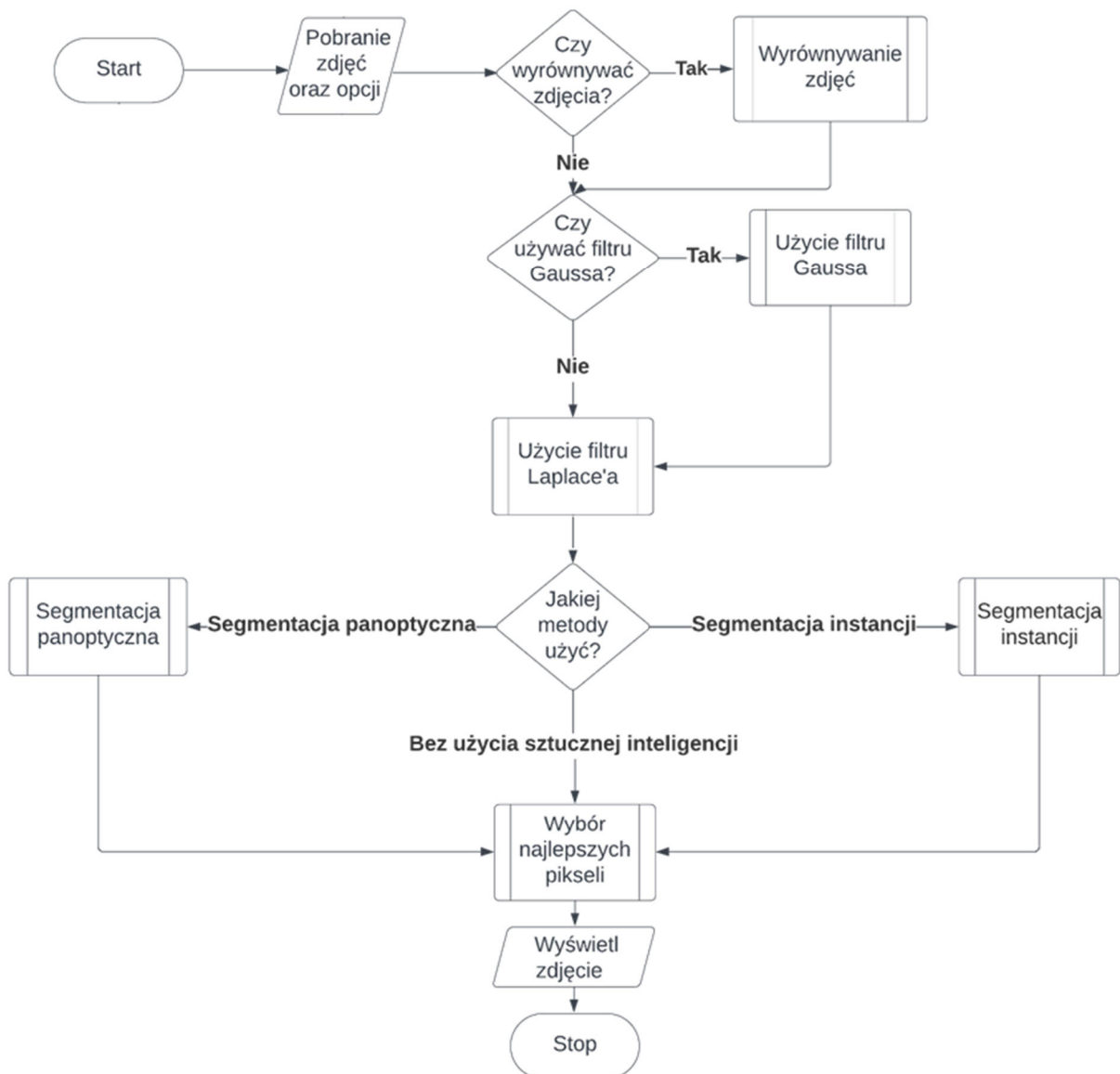
Przedstawiony algorytm rozszerza istniejące algorytmy składania ostrości poprzez odizolowanie konkretnych obiektów, które mogą być problematyczne z powodu ich zmian na kolejnych zdjęciach. Każdy taki obiekt jest następnie dopasowywany do obiektów na pozostałych zdjęciach i wybierana jest spośród nich wersja obiektu o najlepszej jakości, a pozostałe z nich są ignorowane. Pozwala to na usunięcie wielu problemów, które powstają przy używaniu klasycznych algorytmów składania ostrości zdjęć.

Ogólny schemat działania algorytmu przedstawia rys. 2. Algorytm ten można podzielić na dwie części. Pierwsza z nich to część stworzona na podstawie istniejących już algorytmów składania ostrości [3]. Opiera się ona na czterech krokach:

- Wyrównanie zdjęć
- Użycie filtru Gaussa
- Użycie filtru Laplace'a
- Wybór najlepszych pikseli

Druga część algorytmu odpowiada za wykrywanie obiektów z wykorzystaniem sztucznej inteligencji. Używamy tutaj dwa rodzaje segmentacji obrazów: segmentację instancji (ang. instance segmentation) oraz segmentację panoptyczną (ang. panoptic segmentation). Niezależnie od użytej techniki segmentacji efektem jest zmiana zapisanych wyników działania filtru Laplace'a.

Po użyciu filtru Laplace'a algorytmy te zmieniają poszczególne kolory pikseli, znajdujących się w zakresie danego obiektu, na biały lub czarny, zależnie od tego, czy obiekt na tym zdjęciu jest w najlepszej ostrości, czy też nie. W efekcie przy wybieraniu fragmentów o największej ostrości, wybrane zostaną piksele jedynie ze zdjęcia, na którym obiekt ma najlepszą ostrość spośród wszystkich ujęć.



Rysunek 2. Schemat proponowanego systemu składania ostrości

2.1. Wyrównywanie zdjęć

Pierwszym krokiem algorytmu jest ewentualne wyrównanie zdjęć, które jest jednak konieczne tylko wtedy, gdy nastąpiło choćby nieznaczne przesunięcie aparatu fotograficznego względem fotografowanego obszaru między kolejnymi ujęciami, co się mogło wydarzyć na skutek drgań statywu spowodowanych wiatrem, lub wstrząsami podłoża. Wyrównanie przeprowadzamy za pomocą transformacji Euklidesowej z wykorzystaniem algorytmu maksymalizacji zwiększonego współczynnika korelacji (ECC) [3,4]. Proces ten jest bardziej skomplikowany, gdy występują fluktuacje pola widzenia kamery przy zmianie punktu ogniskowania, tzw. focus breathing, ale w ostatnich latach problem udało się w znacznym stopniu rozwiązać i aktualnie w najnowszych dobrych obiektywach zjawisko to zostało ograniczone do minimalnych rozmiarów.

Filtru Gaussa używa się w celu zwiększenia różnicy ostrości między regionami na zdjęciu można użyć filter Gaussa [3]. Użycie filtru Laplace'a skutkuje przypisaniem każdemu pikselowi wartości od 0 do 255. Im większa wartość, tym ostrość danego piksela jest wyższa. Porównując te wartości możliwy jest wybór pikseli o najwyższej ostrości [3,5] Przemnażając sprowadzone do koloru szarości (skala 0-255) fragmenty obrazu o rozmiarze 3x3 piksele przez macierz postaci

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

uzyskuje się liczbę którą jest miarą tego o ile środkowy piksel różni się od otaczających go pikseli. Czym ta różnica jest większa, tym obraz w tym punkcie jest ostrzejszy. Możliwe są też inne postacie filtra Laplace'a.

2.2. Segmentacja instancji

Segmentacja instancji jest rodzajem algorytmu dzielenia zdjęcia na części. Polega na znalezieniu wszystkich obiektów znajdujących się na zdjęciu i stworzeniu masek je obejmujących (rys. 3**Błąd! Nie można odnaleźć źródła odwołania.**). Zdecydowaliśmy się na użycie tego rodzaju algorytmu, ponieważ dokładnie określa on granice każdego wykrytego na zdjęciu obiektu [5].



Rysunek 3. Przykład segmentacji instancji z wykorzystaniem Mask R-CNN.

Użyтым modelem jest implementacja w bibliotece Detectron2 modelu Mask R-CNN [6] o nazwie „Mask R-CNN X152”. Wybraliśmy ten model, ponieważ spośród wszystkich wytrenowanych modeli oferowanych przez Detectron2 charakteryzuje się on najwyższą wartością średniej precyzji maski (ang. mask average precision, AP) [7].

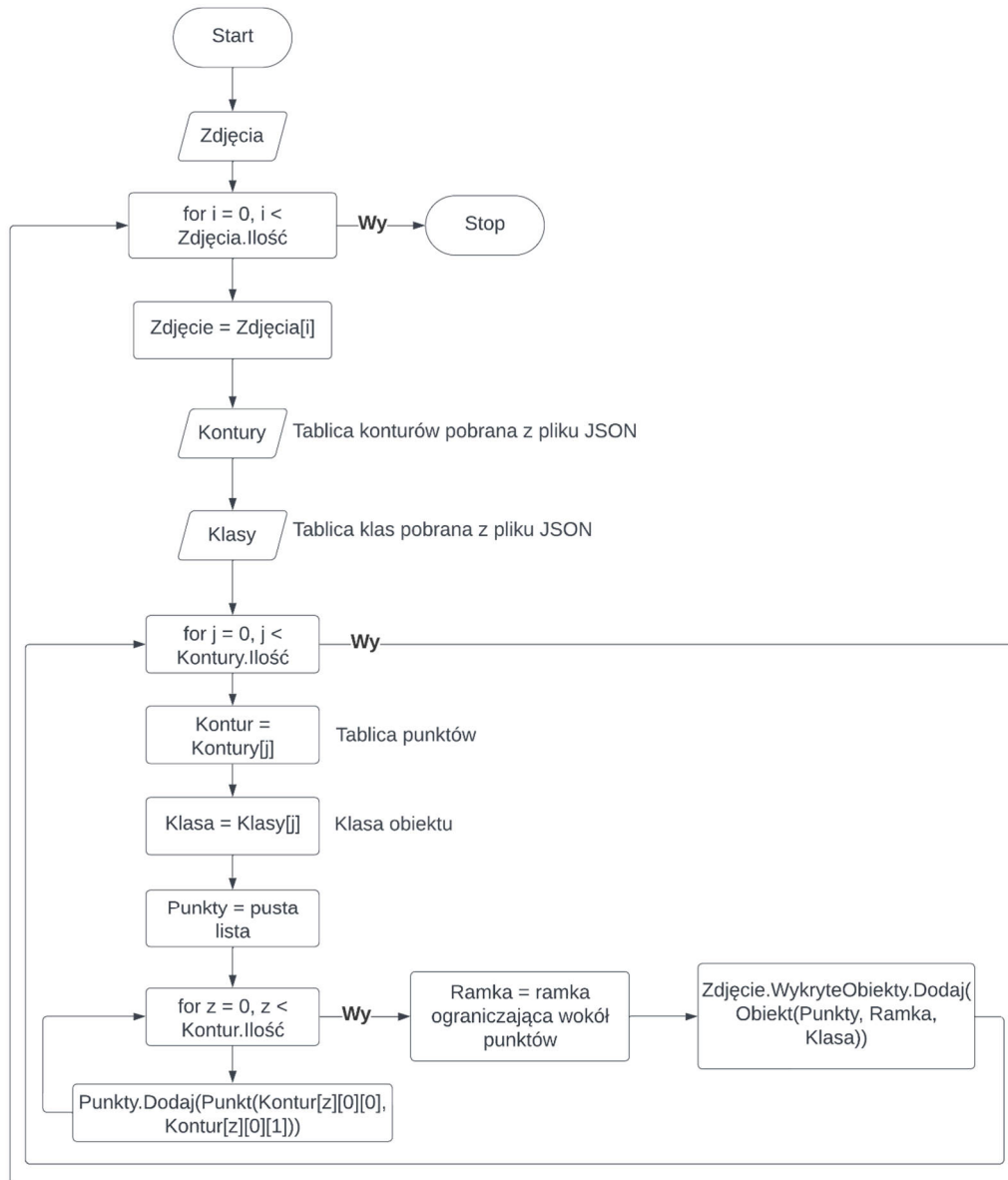
Warto zaznaczyć, że oprócz maski każdego obiektu, zwracana jest również klasa do jakiej obiekt ten został przypisany, co jest pomocne przy dopasowywaniu do siebie obiektów. Z powodu dużej różnicy jakości między zdjęciami pojawia się jednak problem, polegający na tym, że ten sam obiekt na różnych zdjęciach jest przypisywany do różnych klas.

Ogólny schemat działania algorytmu przedstawiono na rys. 2. Składa się on z pięciu kroków. W pierwszym uruchamiany jest model oferowany przez bibliotekę Detectron2 [7,8], który zapisuje kontury oraz klasy wykrytych obiektów do plików JSON. Następnie dane z tych plików są wczytywane do pamięci, a na ich podstawie zmieniane są zdjęcia po użyciu filtru Laplace'a. Poniżej przedstawione jest dokładne wyjaśnienie każdego z kroków.

Proces wyboru obiektów składa się z następujących etapów: pobranie wykrytych obiektów, obliczenie intensywności, dopasowanie obiektów, wybór najlepszych obiektów.

2.3. Pobranie wykrytych obiektów

Po wykryciu obiektów aplikacja pobiera dane zawarte w plikach JSON i zapisuje je w pamięci tymczasowej. Algorytm pobierania danych (rys. 4) iteruje przez wszystkie zdjęcia i pobiera powiązane z nimi tablice danych.



Rysunek 4. Schemat blokowy algorytmu pobierania obiektów (segmentacja instancji)

Pierwsza tablica zawiera współrzędne punktów, które tworzą kontur wykrytego obiektu. Jest to trójwymiarowa tablica o następującej strukturze:

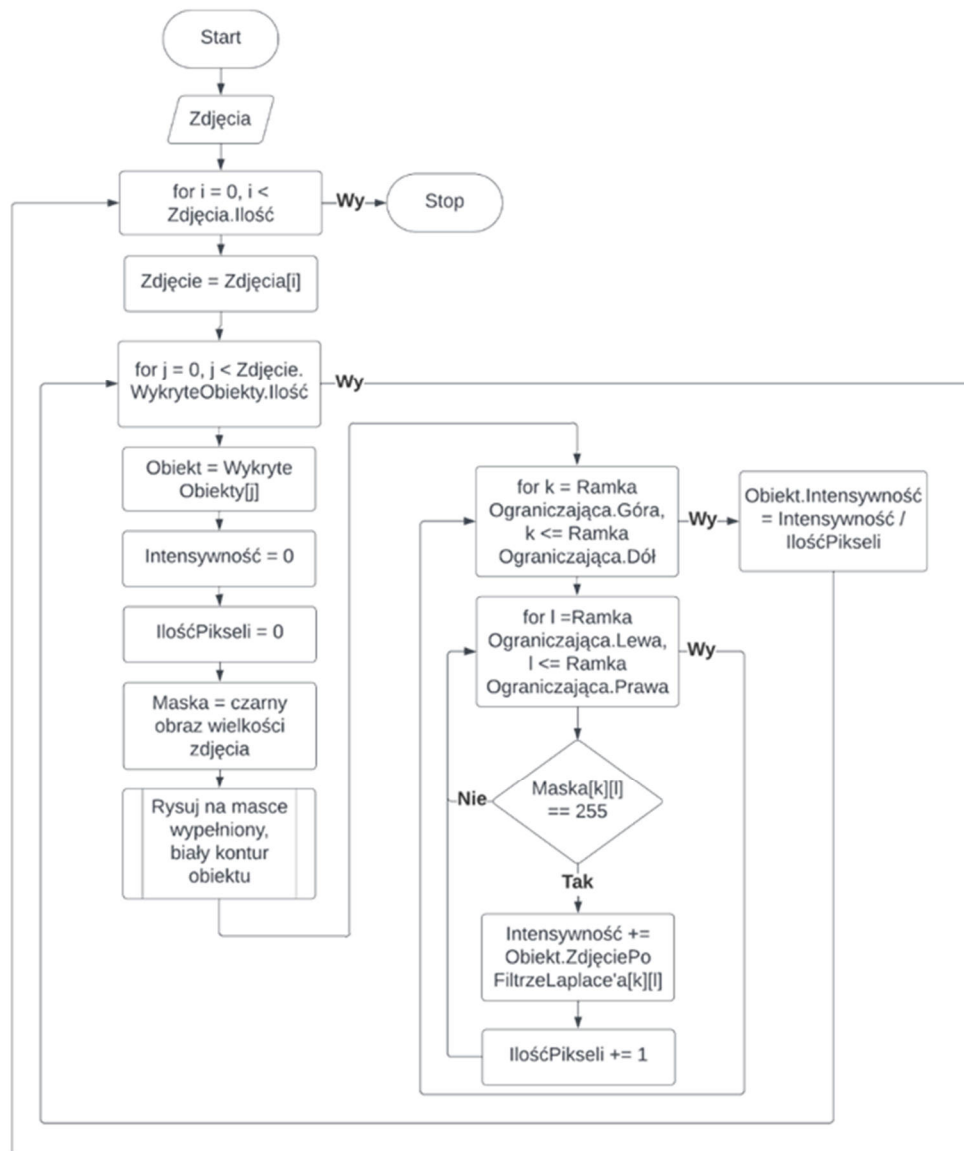
- Wymiar pierwszy - obiekty;
- Wymiar drugi - punkty obiektów;
- Wymiar trzeci - współrzędne punktów;

Druga tablica zawiera klasy obiektów, przedstawione w formie liczb całkowitych. Kolejność rekordów tej tablicy odpowiada kolejności obiektów w pierwszej tablicy.

Po pobraniu tablic i przekształceniu danych do odpowiedniej formy, tworzona jest również ramka ograniczająca wokół obiektu. Obiekt z takimi danymi zapisywany jest w kolekcji obiektów wykrytych na danym zdjęciu.

2.4. Obliczenie intensywności

Proces obliczania średniej intensywności pikseli w danym obiekcie przedstawiona na rys. 5. Przeprowadzając iterację przez wszystkie zdjęcia oraz przez wszystkie obiekty na tych zdjęciach tworzy się maskę koloru czarnego oraz rysuje na niej wypełniony kontur, koloru białego, danego obiektu.

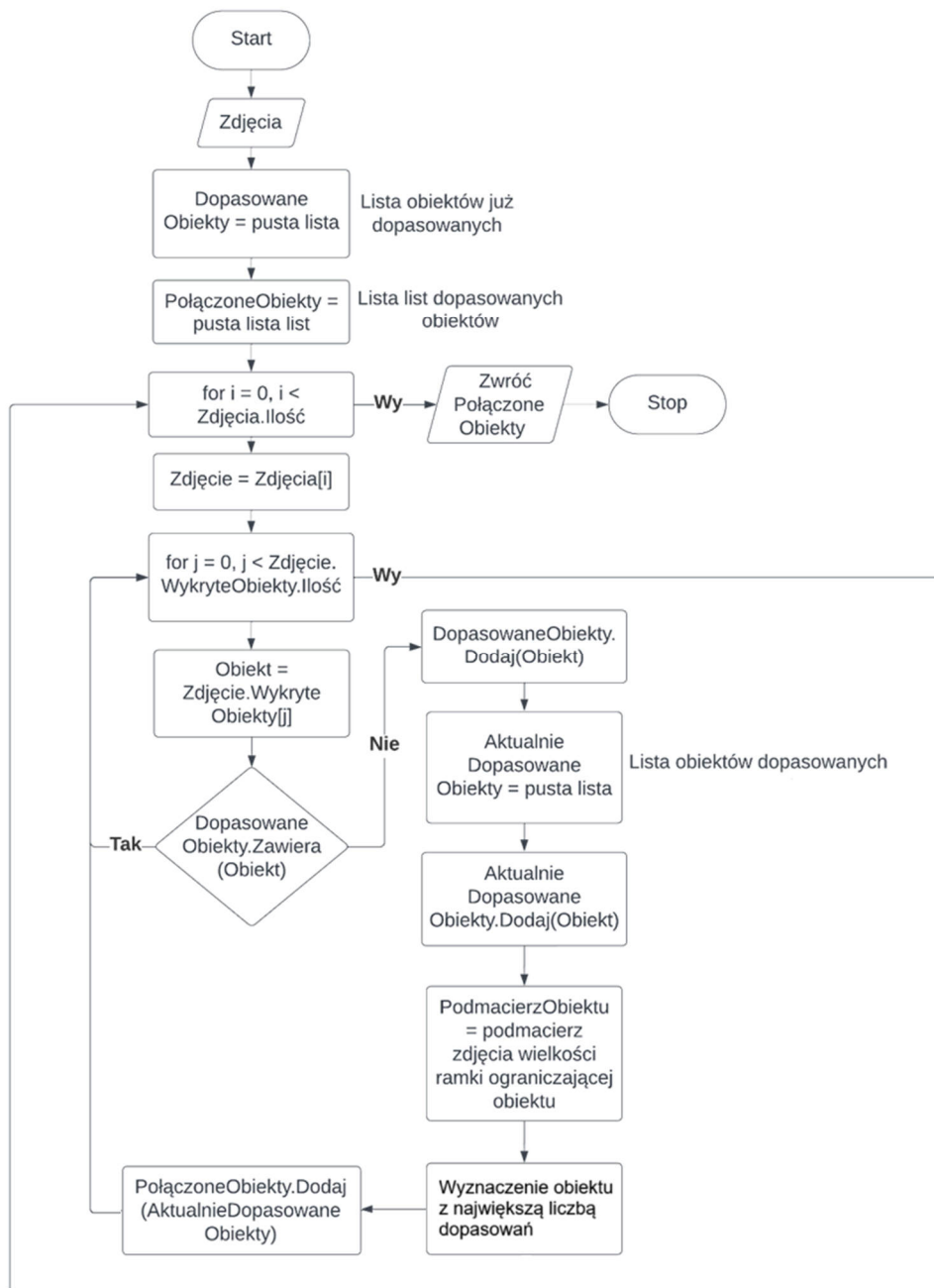


Rysunek 5. Schemat blokowy algorytmu obliczania średniej intensywności (segmentacja instancji)

Następnie przeprowadzając kolejną iterację przez wszystkie piksele znajdujące się w ramce ograniczającej danego obiektu sprawdza się czy piksel na masce jest koloru białego. Jeśli znaleziony zostanie piksel koloru białego, intensywność tego piksela, pobrana ze zdjęcia po użyciu filtru Laplace'a, dodawana jest do ogólnej intensywności obiektu. Po sprawdzeniu wszystkich pikseli, końcowa intensywność dzielona jest przez liczbę znalezionych pikseli. W rezultacie otrzymujemy średnią intensywność pikseli, która następnie zapisywana jest w modelu obiektu.

2.5. Dopasowanie obiektów

W celu porównania ostrości obiektów pomiędzy zdjęciami konieczna jest wiedza czy obiekty, które są ze sobą porównywane to te same obiekty. Ponieważ obiekty mogą zmieniać swoje położenie nie jest możliwe określenie tego na podstawie kolejności zapisu informacji o obiekcie w pliku JSON. Dodatkowo przy dużych zmianach ostrości obiekty mogą być wykrywane jedynie na niektórych ze zdjęć, na których się pojawiają.



Rysunek 6. Schemat blokowy algorytmu dopasowywania obiektów

Aby temu zaradzić wykorzystane zostały możliwości biblioteki OpenCV [8,9] w zakresie dopasowywania cech (ang. feature matching). Program korzysta z oferowanego przez OpenCV detektora BRISK (Binary Robust Invariant Scalable Key-points detector) oraz metody dopasowywania BF (Brute Force), ponieważ to połączenie daje najlepsze rezultaty.

Całość algorytmu odpowiedzialnego za dopasowywanie obiektów przedstawiono na rys. 6. Tworzone są dwie tymczasowe kolekcje pomocnicze nazwane DopasowaneObiekty i PołączoneObiekty. Pierwsza z tych kolekcji zawiera obiekty, które już zostały dopasowane z innymi obiektami, tak aby nie doszło do sytuacji, w której jakiś obiekt łączony jest z kilkoma, całkowicie różnymi, obiektami. Druga z kolekcji zawiera listy tych samych, dopasowanych obiektów.

Aby zobrazować zawartość tych kolekcji przyjmijmy następującą sytuację. Do programu wprowadzone zostały 4 zdjęcia, na każdym ze zdjęć wykrytych zostało 5 obiektów. W tej sytuacji po przeprowadzeniu algorytmu dopasowywania w kolekcji DopasowaneObiekty znajdować się będzie 20 obiektów (ponieważ to jest łączna liczba wykrytych obiektów). Natomiast kolekcja PołączoneObiekty będzie się składać z pięciu list (liczba różnych obiektów), z której każda z tych list zawierać będzie 4 obiekty (liczba zdjęć, a więc ilość razy, którą dany obiekt się pojawia).

Algorytm dla każdego ze zdjęć przeprowadza iterację po wszystkich wykrytych na nim obiektach. Sprawdzane jest czy aktualny obiekt znajduje się już w kolekcji DopasowaneObiekty, jeśli nie, jest on dodawany do tej kolekcji oraz do tymczasowej kolekcji AktualnieDopasowaneObiekty, kolekcja ta jest zerowana przy każdej iteracji. Następnie tworzona jest podmacierz, wielkości ramki ograniczającej aktualnego obiektu. W następnym kroku przeprowadzane są ponownie iteracje po wszystkich zdjęciach i po wszystkich obiektach wykrytych na tych zdjęciach, ignorowane jest jednak zdjęcie, z którego pochodzi obiekt z poprzedniej iteracji. Tworzone są również dwie pomocnicze zmienne NajwięcejDopasowań i Index. Ponownie sprawdzane jest czy obiekt należy już do kolekcji DopasowaneObiekty. Jeśli tak nie jest, tworzona jest podmacierz wielkości ramki ograniczającej tego obiektu i razem z podmacierzą zawierającą poprzedni obiekt, przekazywane są one do funkcji zwracającej ilość dopasowań między dwoma zdjęciami. Jeśli wynik zwrócony przez tą funkcję jest większy od zawartości zmiennej NajwięcejDopasowań, wynik ten jest zapisywany w tej zmiennej, a do zmiennej Index zapisuje się index aktualnie sprawdzanego obiektu.

Po przeprowadzeniu iteracji przez wszystkie obiekty na zdjęciu, obiekt o indeksie zawartym w zmiennej Index, jeśli nie jest już zawarty w kolekcji DopasowaneObiekty, jest dodawany do niej i do kolekcji AktualnieDopasowaneObiekty. Następnie przeprowadzana jest iteracja po kolejnym zdjęciu. Jeśli obiekt znajduje się już w kolekcji DopasowaneObiekty, oznacza to, że na aktualnym zdjęciu nie ma obiektu, który można dopasować z obiektem wybranym w początkowej iteracji. W takim przypadku do kolekcji wykrytych obiektów, na tym zdjęciu, dodaje się kopię tego obiektu, z intensywnością ustawioną na 0. Tą kopię dodaje się następnie do kolekcji DopasowaneObiekty i AktualnieDopasowaneObiekty. Ma to na celu zachowanie identycznych rozmiarów kolekcji oraz zabezpieczenie przed sytuacją, w której obiekt w rzeczywistości znajduje się w tym miejscu, jednak z powodu niskiej jakości, nie został wykryty.

Po przeprowadzeniu tych operacji kolekcja AktualnieDopasowaneObiekty dodawana jest do listy PołączoneObiekty i cała operacja powtarzana jest dla kolejnego obiektu, a następnie dla kolejnego zdjęcia. W sytuacji idealnej, w której na wszystkich zdjęciach wykryta jest taka sama ilość, tych samych obiektów, konieczna jest iteracja jedynie przez obiekty wykryte na pierwszym ze zdjęć. Po zakończeniu pracy algorytmu zwracana jest kolekcja PołączoneObiekty.

2.6. Wybór najlepszych obiektów

Końcowym etapem algorytmu jest wybór obiektów o najlepszej średniej intensywności. Po otrzymaniu listy kolekcji dopasowanych obiektów algorytm dokonuje iteracji po tej liście. Tworzy się zmienną pomocniczą MaksymalnaIntensywność i ustala jej wartość na 0. Następnie przeprowadza się trzy iteracje o liczbie kroków równej ilości zdjęć, a więc również równej ilości obiektów w każdej z kolekcji.

Pierwsza z iteracji ma za zadanie znaleźć w kolekcji obiekt o najwyższej intensywności. Jeśli intensywność aktualnego obiektu jest wyższa od MaksymalnaIntensywność, zapisuje się w tej zmiennej tą intensywność, a w zmiennej NajlepszyObiekt aktualny obiekt.

Następna iteracja ma za zadanie znaleźć zdjęcie, na którym znajduje się NajlepszyObiekt. Następnie rysuje się, na tym zdjęciu po użyciu filtru Laplace'a, wypełniony, biały kontur tego obiektu. Po znalezieniu zdjęcia iteracja zostaje

przerwana. Iteracja ta jest konieczna, ponieważ kolejność obiektów w kolekcji dopasowanych obiektów może różnić się od kolejności zdjęć.

Ostatnia z iteracji sprawdza czy aktualne zdjęcie jest zdjęciem zawierającym NajlepszyObiekt. Jeśli tak nie jest rozpoczyna się iteracja po obiektach wykrytych na tym zdjęciu. Znajduje się w taki sposób obiekt obecny na tym zdjęciu, który należy jednocześnie do aktualnej kolekcji dopasowanych obiektów. Po znalezieniu tego obiektu rysuje się, na wersji po użyciu filtru Laplace'a tego zdjęcia, dwa wypełnione czarne kontury. Pierwszy to kontur tego obiektu. Drugi to kontur obiektu zapisanego w zmiennej NajlepszyObiekt. Na wersji zdjęcia po użyciu filtru Laplace'a zawierającego NajlepszyObiekt rysuje się również wypełniony biały kontur aktualnego obiektu.

Po przeprowadzeniu tych czynności na zdjęciach po użyciu filtru Laplace'a znajdują się regiony całkowicie białe i całkowicie czarne. Zapewnione zostaje w ten sposób, że przy wybieraniu pikseli do zdjęcia wynikowego, algorytm wybierze piksele ze zdjęcia, na którym dany obiekt ma jakość najlepszą.

2.7. Wybór najlepszych pikseli

Ostatnim krokiem algorytmu jest wybór najlepszych pikseli do zdjęcia wynikowego. Może się to odbywać na kilka sposobów. Pierwszy, najprostszy, to sprawdzanie każdego piksela i porównywanie go z tym samym pikselem na innych zdjęciach. Do zdjęcia końcowego wybierany jest piksel, którego intensywność jest największa. Rozwiązanie to sprawdza się dobrze w prostych, kontrolowanych przypadkach, jednak gdy poszczególne zdjęcia zaczynają się między sobą różnić prowadzi to do powstawania dużej ilości szumu.

Druga metoda polega na sprawdzaniu intensywności, nie pojedynczego piksela, ale obszaru wokół tego piksela. Dopasowując rozmiar tego obszaru możliwe jest wyeliminowanie dużej ilości szumu. Do końcowego zdjęcia brany jest jednak nadal pojedynczy piksel, wokół którego ten obszar był sprawdzany.

Ostatnia metoda polega również na sprawdzaniu obszaru zamiast pojedynczego piksela, jednak do zdjęcia końcowego wybierany jest cały ten obszar, a nie jedynie pojedynczy piksel. Przyspiesza to mocno działanie algorytmu jednak prowadzi do gorszych efektów.

Wybór metody zależy od zdjęć na jakich zamierzamy przeprowadzić składanie ostrości. Poniżej opisujemy dokładnie wszystkie wykorzystywane metody.

2.8. Segmentacja panoptyczna

Segmentacja panoptyczna (ang. panoptic segmentation) [11] jest połączeniem segmentacji semantycznej (ang. semantic segmentation) oraz segmentacji instancji. Segmentacja semantyczna dąży do przypisania każdemu pikselowi na zdjęciu jakiejś klasy. Natomiast segmentacja instancji przypisuje piksele do konkretnych obiektów. Połączenie tych dwóch technik prowadzi do otrzymania zdjęcia, w którym wszystkie obiekty są prawidłowo wykryte a pozostałe elementy mają przypisaną odpowiednią klasę [5,12].

Zauważyliśmy, że do niektórych zdjęć segmentacja panoptyczna się lepiej sprawdzała od segmentacji instancji. Dotyczyło to zwłaszcza zdjęć, gdzie liczba małych lub dużych zmian między kilkoma ujęciami jest bardzo duża, a nie wszystkie elementy, jak na przykład roślinność, są dokładnie wykrywane. Najlepiej pokazuje to przykład zdjęć zbiorników wodnych. Niezależnie od okresu w jakim zdjęcia będą robione, woda na każdym z nich będzie wyglądać inaczej, a segmentacja instancji nie wykrywa wody jako obiektu. W rezultacie zdjęcie końcowe prezentowałoby słabą jakość. Problem ten rozwiązało użycie w takich przypadkach segmentacji panoptycznej.

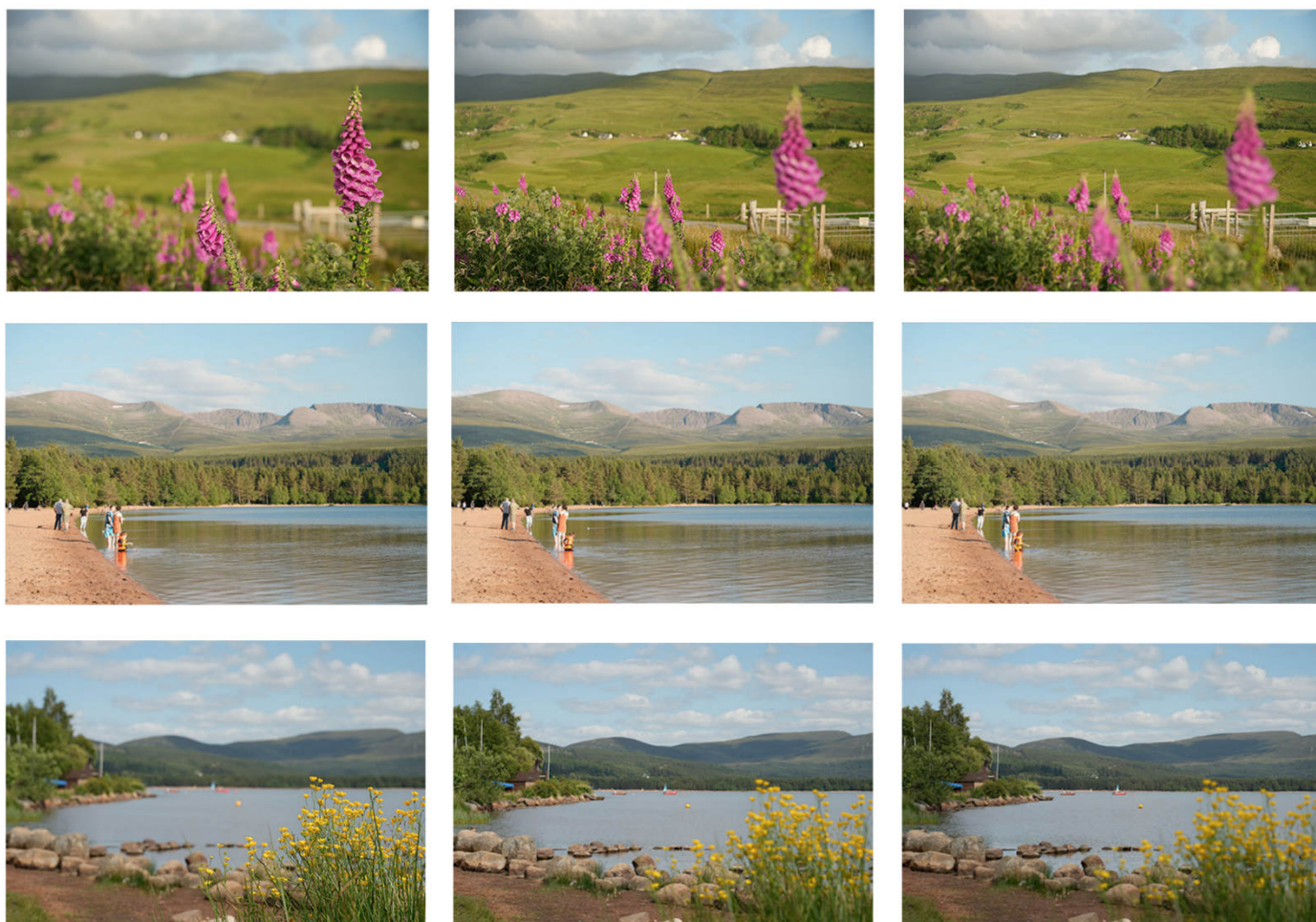
3. Testy i wyniki

Przeprowadzone zostały liczne testy z wykorzystaniem wielu różnych zdjęć, używając różnych metod oraz parametrów. Zdjęcia wynikowe porównywane były ze zdjęciami wytworzonymi przez inne programy.

W wyniku przeprowadzonych testów wyciągnięte zostały następujące wnioski:

- Algorytm składania ostrości stworzony w tej pracy wymaga udoskonalenia, inne programy, zawierające inne i bardziej skomplikowane algorytmy dają o wiele lepsze wyniki jeśli zmiany między ujęciami nie są duże.

- Segmentacja instancji rozwiązuje problem obiektów zmieniających się między ujęciami, jednak w wyniku słabych wyników algorytmu składania ostrości jakość obszarów wokół tych obiektów jest zbyt niska, aby technika ta mogła być stosowana z aktualnie używanym algorytmem.
- Przy dużych zmianach między ujęciami najlepsze efekty dawała segmentacja panoptyczna. Rozwiązywany jest tutaj problem słabych wyników algorytmu składania ostrości, ponieważ przy tej metodzie całe zdjęcie dzielone jest na segmenty, a wszystkim pikselom zdjęcia przypisywane są wartości 0 lub 255, dzięki czemu usuwane są wszelkie zakłócenia. Metoda ta skutkuje jednak, w niektórych przypadkach, maksymalnej możliwej ostrości.



Rysunek 7. Niektóre ze zdjęć wykorzystanych w testach. Na dolnych i górnych zdjęciach wiatr powoduje ruch roślin, na środkowych zdjęciach przemieszczają się postacie oraz fale na wodzie.

Rys. 7. przedstawia trzy przykładowe zdjęcia wybrane spośród zdjęć wykorzystanych do testowania programu. Na rys. 8 u góry przedstawiona jest sytuacja, w której segmentacja panoptyczna nie daje pełnej możliwej ostrości, jednak wyniki są nadal najlepsze w porównaniu z innymi metodami, ponieważ przy ich wykorzystaniu pojawia się wiele nienaturalnych elementów, które wymagają poprawy.

W przypadku obrazu przedstawionego na rys. 8 w środkowej linii nie udało się osiągnąć idealnych wyników, ponieważ małej wielkości elementy tła nie były odpowiednio eliminowane przez wykorzystane metody segmentacji.

Rys. 8 w dolnej linii przedstawia sytuację, w której to segmentacja instancji poradziła sobie najlepiej. Na pozostałych zdjęciach widoczne są zniekształcenia sylwetek osób. Należy jednak zaznaczyć, że z powodu ograniczeń technicznych zostały w tej pracy przedstawione zdjęcia o dość niskiej rozdzielczości. Przy użyciu zdjęć o wyższej rozdzielczości na zdjęciu wynikowym po użyciu segmentacji instancji widoczne są jednak pewne szumy.

Kod źródłowy naszego systemu można pobrać ze strony:

https://drive.google.com/file/d/1oWC_EOvDffPG3qU1rzkP9i38paNaPv50/view



Helicon Focus



segmentacja instancji



segmentacja panoptyczna



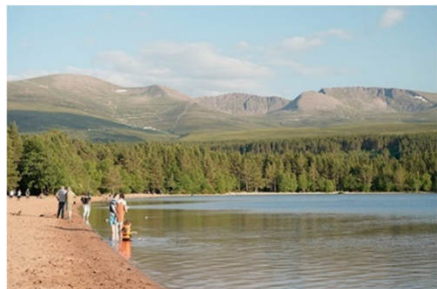
Helicon Focus



segmentacja instancji



segmentacja panoptyczna



Helicon Focus



segmentacja instancji



segmentacja panoptyczna

Rysunek 8. Przykładowe otrzymane wyniki

4. Wnioski

Niniejsza praca łączy proces klasycznego składania ostrości z wykrywaniem obiektów przez sztuczną inteligencję (sieci konwolucyjne), a następnie przez przeprowadzenie szeregu dodatkowych opisanych operacji. Celem tego podejścia jest umożliwienie efektywnego składania ostrości zapewniające prawidłowe odwzorowanie obiektów dynamicznie zmieniających się między poszczególnymi ujęciami.

W wielu przypadkach, choć nie zawsze, byliśmy w stanie uzyskać wyniki charakteryzujące się zadowalającą ostrością w poszczególnych punktach wynikowego zdjęcia oraz niskim poziomem artefaktów. Segmentacja panoptyczna dawała często dobre wyniki, jednak minusem tej metody jest nie osiąganie pełnej możliwej ostrości w niektórych przypadkach. Segmentacja instancji również dawała dobre wyniki, jednak aby obszary wokół wykrytych obiektów prezentowały dobrą jakość odwzorowania rzeczywistości (dobrą ostrość i brak artefaktów), musi być ona połączona z odpowiednim algorytmem składania ostrości.

Literatura

1. R. C. Gonzalez, R. E. Woods. *Digital Image Processing (Third Edition)*, Pearson, 2007.
2. D. Johnson. *How to Do Everything: Digital Camera (Fifth Edition)*, McGraw-Hill/Osborne Media, 2008.
3. J. Wlodek, K. J. Gofron, Y. Q. Cai. Achieving 3D imaging through focus stacking, *AIP Conference Proceedings*, 2019.
4. David Choi; Aliya Pazyzbekova; Wuhan Zhou; Peter van Beek. Improved image selection for focus stacking in digital photography 2017 *IEEE International Conference on Image Processing (ICIP)*
5. R. Szeliski. *Computer Vision: Algorithms and Applications (Second Edition)*, Springer, 2022.
6. K. He, G. Gkioxari, P. Dollar, R. Girshick. Mask R-CNN, *IEEE International Conference on Computer Vision (ICCV)*, Wenecja, Włochy, 2017.
7. Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, R. Girshick. Detectron2, 2019. <https://github.com/facebookresearch/detectron2>
8. G. Bradski i A. Kaehler. *Learning OpenCV*, O'Reilly Media, 2008.
9. G. Bradski. The OpenCV Library, *Dr. Dobb's Journal of Software Tools*, 2000.
10. G. D. Evangelidis, E. Z. Psarakis. Parametric image alignment using enhanced correlation coefficient maximization, *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, pp. 1858-1865, 2008.
11. Kirillov, R. Girshick, K. He, P. Dollár. Panoptic feature pyramid networks, *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019.
12. F. K. Noble. Comparison of OpenCV's feature detectors and feature matchers, *23rd International Conference on Mechatronics and Machine Vision in Practice (M2VIP)*, Nanjing, Chiny, 2016.