

Autor: Szymon Tokarz

Data: 25.10.2024r. godz.8.00

Rezultaty

Część II.1

Wszystkie wartości w poniższej tabeli wskazują na piksel 70x90.

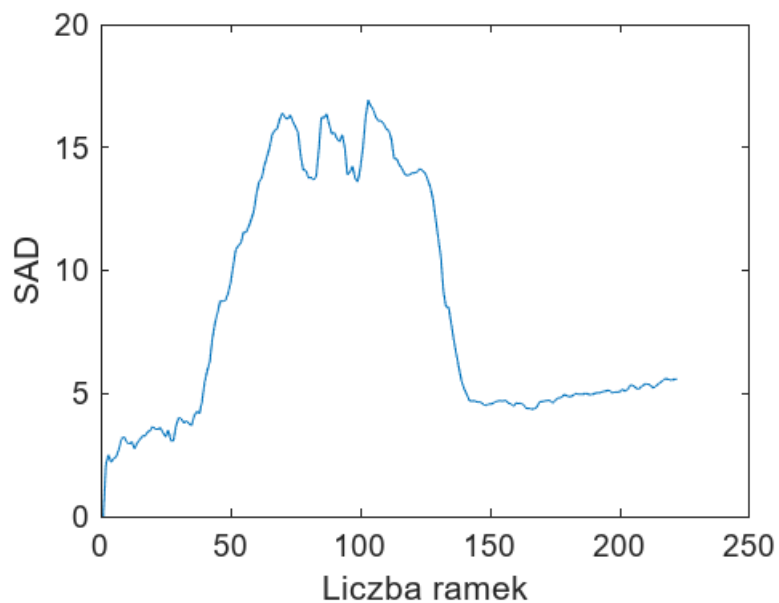
	Wartość piskela dla im0	Wartość piskela dla im1	Różnica
D1	109	208	0
D2	109	208	-99
D3	0.4275	0.8157	-0.3882
D4	109	208	99

Część II.2

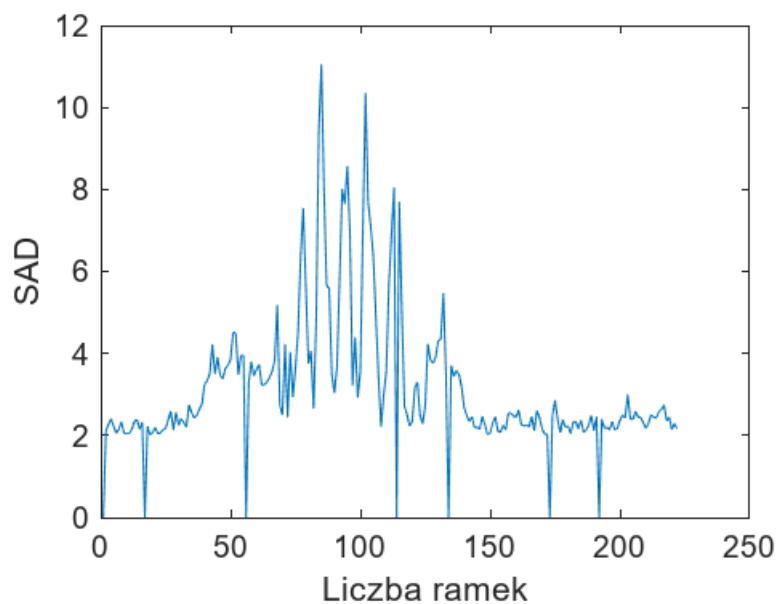
Numer obrazu	Granica dolna	Granica górna	Typ danych
D1	0	126	Uint8
D2	-129	126	Uint8
D3	-129	126	Uint8
D4	0	129	Uint8

Najlepsze obrazy różnicowe to d2 oraz d3. W przypadku d1 proste odjęcie obrazu powodowało przepiętnienie przez co duża część informacji o pikselu mogła zostać utracona. Różnica pomiędzy d2 i d3 jest taka, że w przypadku obrazu d3 użyto funkcji `matlaba im2double`, która służy do konwersji obrazu do formatu `double`.

Część II.3



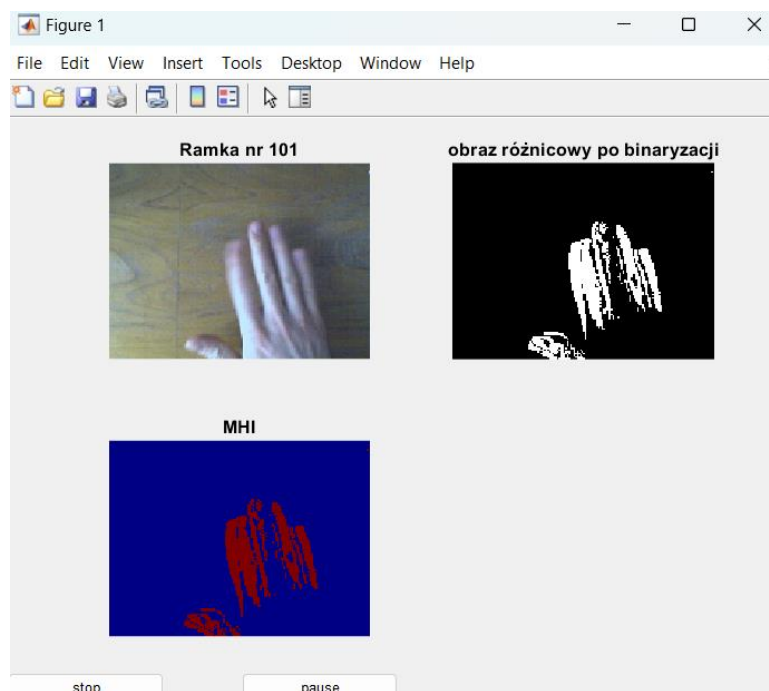
Rys.1 Wykres SAD z obrazem tła



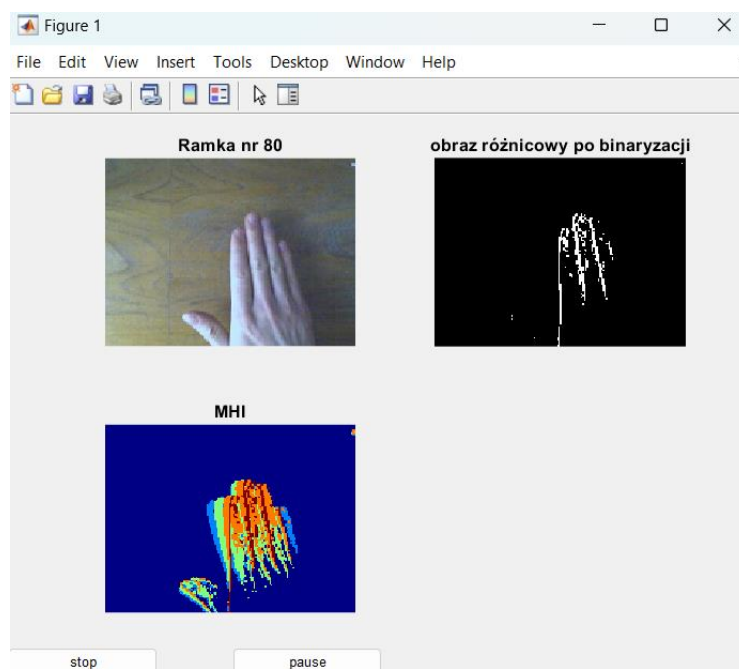
Rys.2 Wykres SAD z obrazem z poprzedniej ramki

Można zauważyć, że algorytm SAD działa dużo poprawniej, gdy poda mu się obraz tła jako jeden z argumentów. Nie występują wtedy takie wahania, jak w przypadku sytuacji z Rys.2, gdzie algorytm miał duży problem stwierdzić, czy obiekt faktycznie się rusza ze względu na brak możliwości ustawienia jednoznacznego progu.

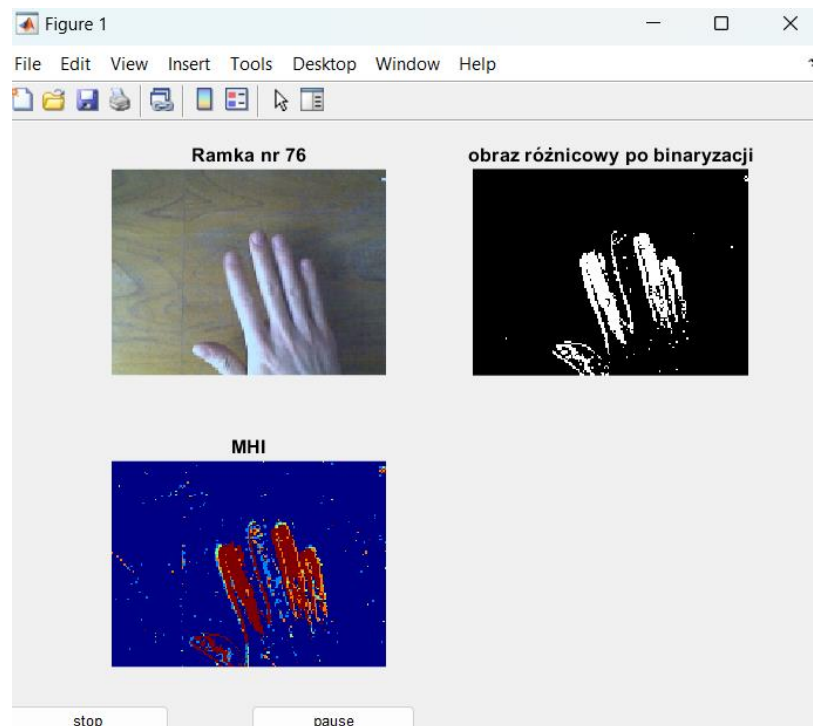
Część III.1



Rys.3 MHI dla $\tau=1$, próg=20



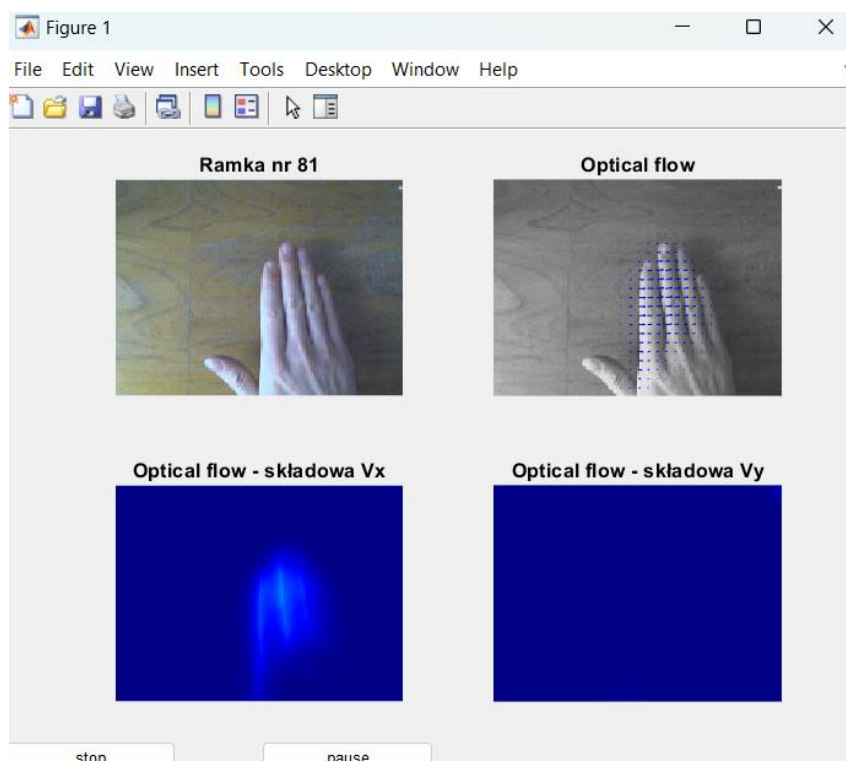
Rys.4 MHI dla $\tau=4$, próg=20



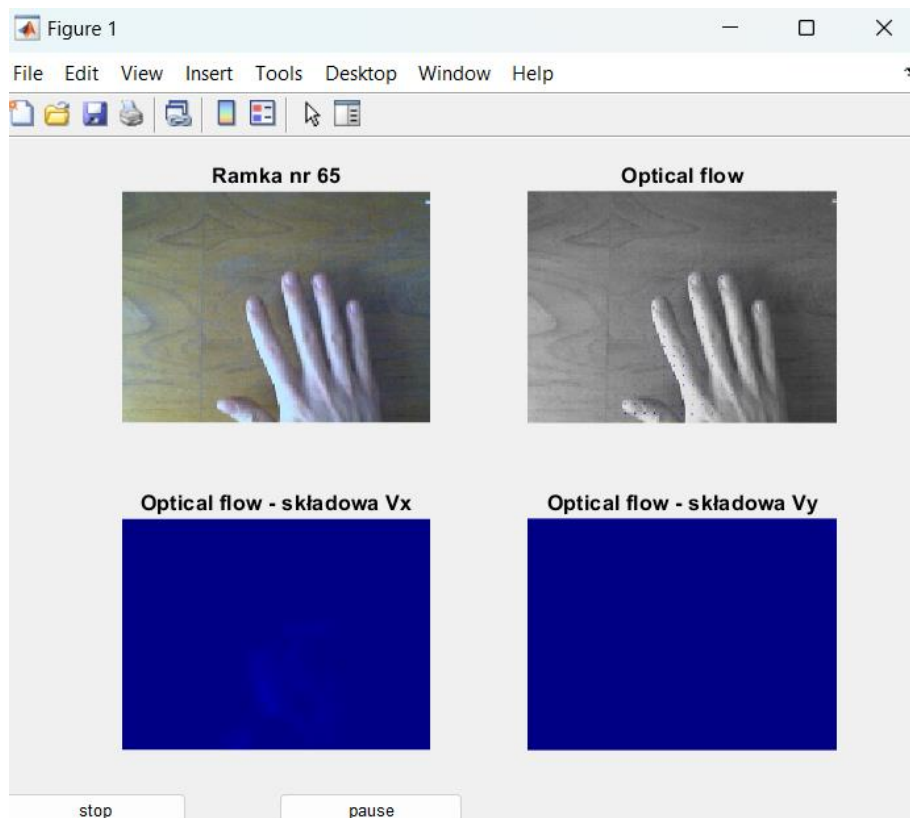
Rys.5 MHI dla $\tau=4$, próg=10

Dla zbyt małego progu zaczyna pojawiać się szum. Z kolei gdy dobierze się zbyt małe τ to dłoń jest słabo wykrywalna.

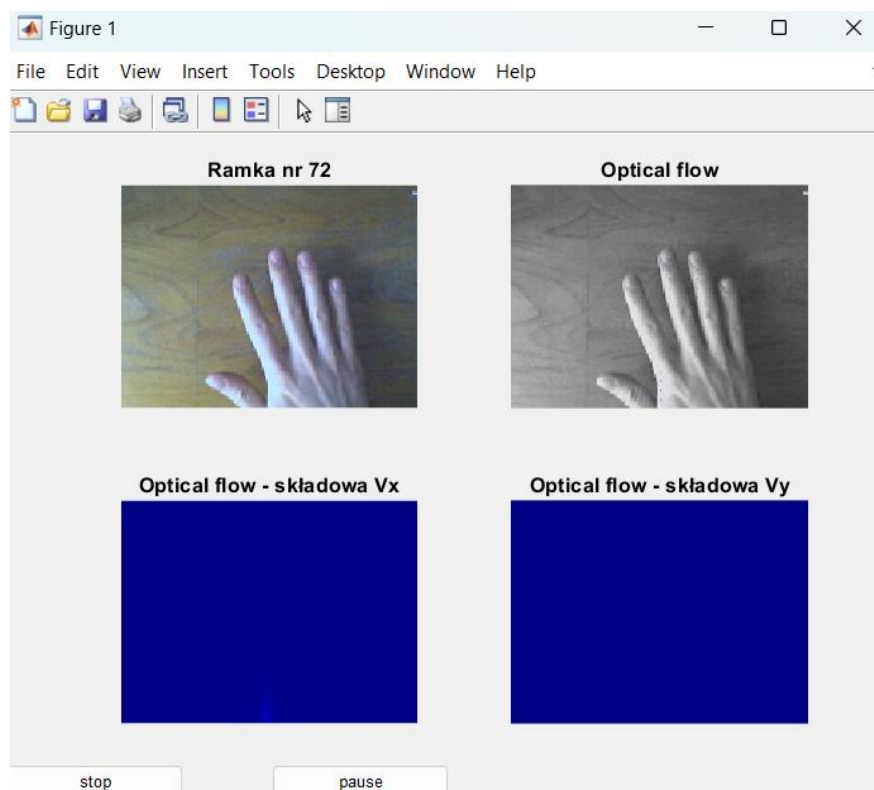
Część IV.1



Rys.6 OF smoothes=5, iteration=120



Rys.7 OF smoothnes=10,iteration =120



Rys.8 OF smoothness=10 iteration =60

Analiza i wnioski

Część II.1

Trzeba przekonwertować obrazy do typu double lub użyć wartości bezwzględnych.

Część II.2

Funkcja `im2double()` konwertuje obrazy do typu double oraz je normalizuje do wartości z przedziału 0 do 1.

Część II.3

Lepszy jest drugi sposób, ponieważ jakość wyników jest podobna, a nie wykorzystuje się bez powodu zbyt dużej liczby bitów.

Część II.4

Wartość SAD to suma wartości wszystkich pikseli podzielona przez ich liczbę, czyli działa jak średnia arytmetyczna.

Część III.1

Obraz wynikowy w algorytmie MHI jest tworzony jako funkcja historii ruchu – wartości pikseli z bieżącej klatki są sumowane z odpowiadającymi im wartościami z wcześniejszych klatek, przy zastosowaniu odpowiednich wag. W rezultacie, jaśniejsze obszary wskazują na niedawny ruch, natomiast ciemniejsze oznaczają ruch, który wystąpił wcześniej.

Część IV.1

Algorytm optycznego przepływu (OF) obejmuje wczytanie obrazu, konwersję do skali szarości, obliczenie estymacji w zadanej liczbie iteracji oraz wygenerowanie obrazu końcowego. Metody OF mają na celu oszacowanie ruchu pikseli między kolejnymi klatkami. Aproksymacja zazwyczaj bazuje na rozwinięciu Taylora. Wyznaczenie ruchu z pojedynczej klatki jest niemożliwe.

Pytania

Pytanie 1

Polega ona na obliczaniu różnicy bezwzględnej między dwoma kolejnymi klatkami (lub klatki obecnej z klatką tła) i na tej podstawie ocenieniu, czy nastąpił ruch. Operacja pozwala więc określić jak bardzo dwie klatki różnią się od siebie. Obliczony wskaźnik przyjmuje wartość tym większą, im więcej pikseli uległo zmianie.

Pytanie 2

Detekcja ruchu za pomocą czujników PIR (ang. Passive Infrared – pasywna podczerwień) opiera się na wykrywaniu zmian w promieniowaniu podczerwonym emitowanym przez obiekty w polu widzenia czujnika.