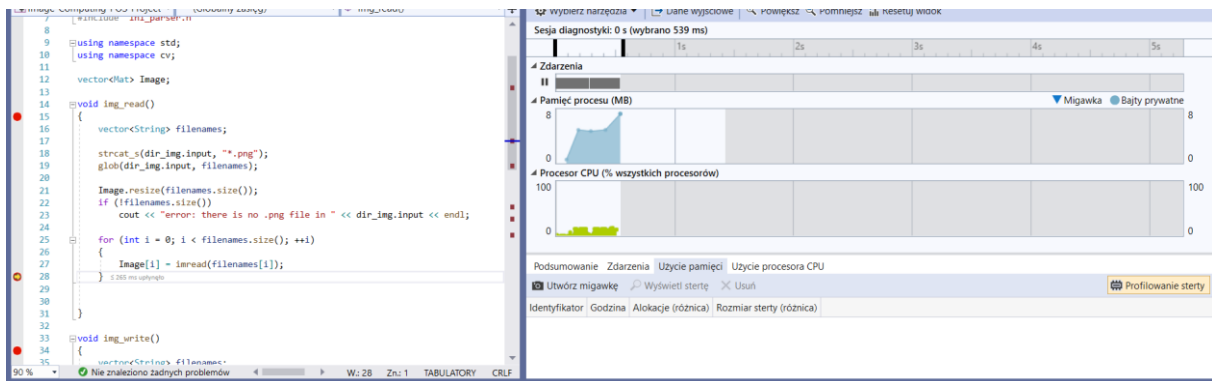


Wykonanie profilowania projektu.

Wykrywanie hotspotów:

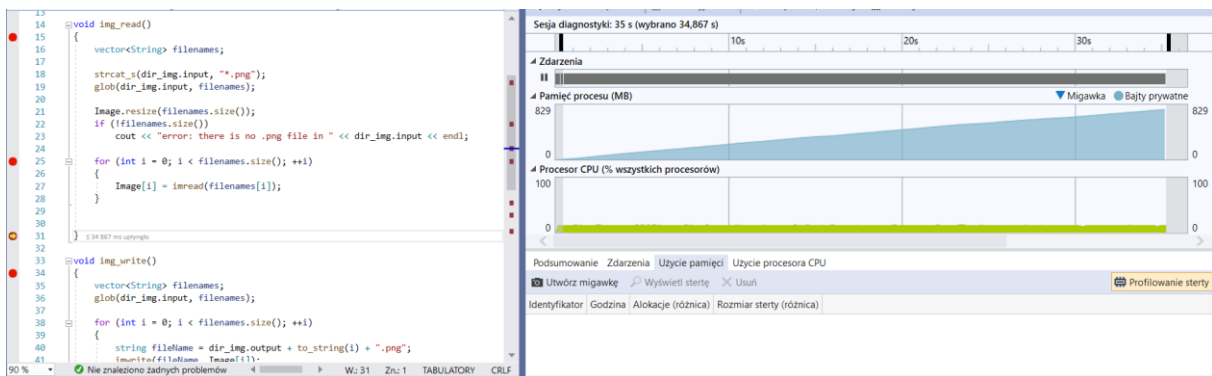
Program przed próbą poprawy hotspotów.

Inicjalizacja programu:



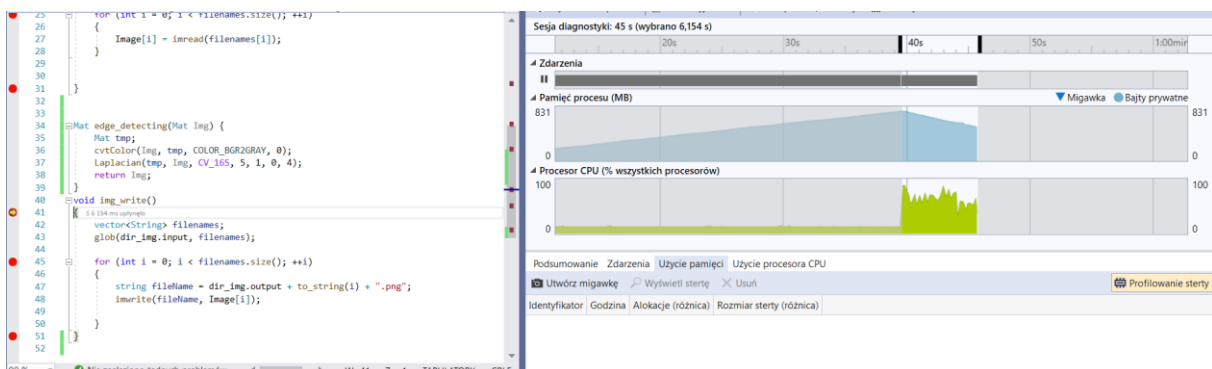
Czas trwania 265ms

Wczytywanie zdjęć:



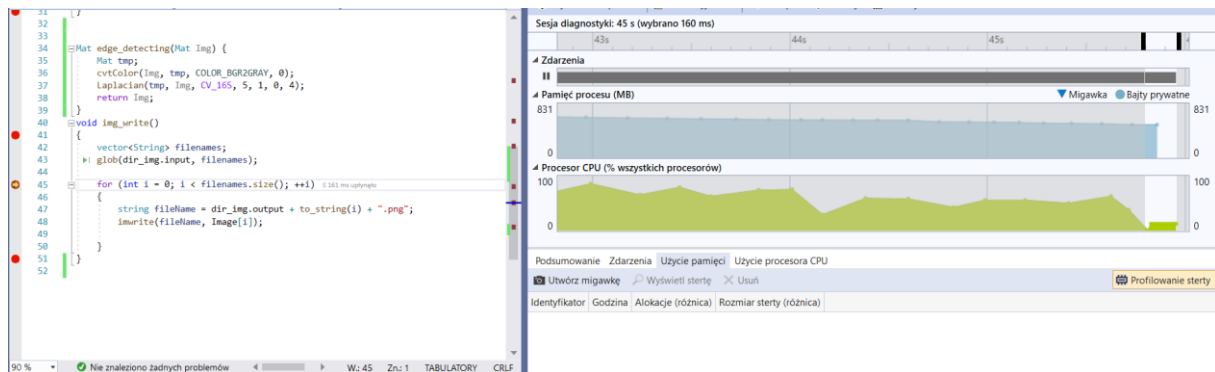
Czas trwania: 34,847s

Wykrywanie krawędzi:



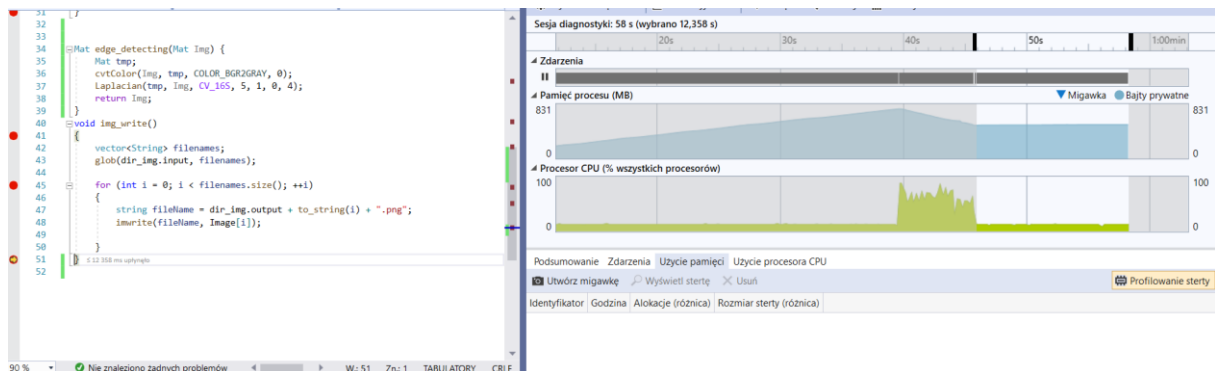
Czas trwania: 6,154s

Inicjalizacja zapisu zdjęć:



Czas trwania: 161ms

Zapis przetworzonych obrazów:



Czas trwania: 12,358s

Łączny czas wykonywania programu, bez generowania macierzy wejściowych i wyjściowych wynosi:

53,785s

Zauważone hotspoty wystąpiły podczas wykonywania się funkcji:

Pętli for w funkcji img_read()

Pętli for w funkcji img_write()

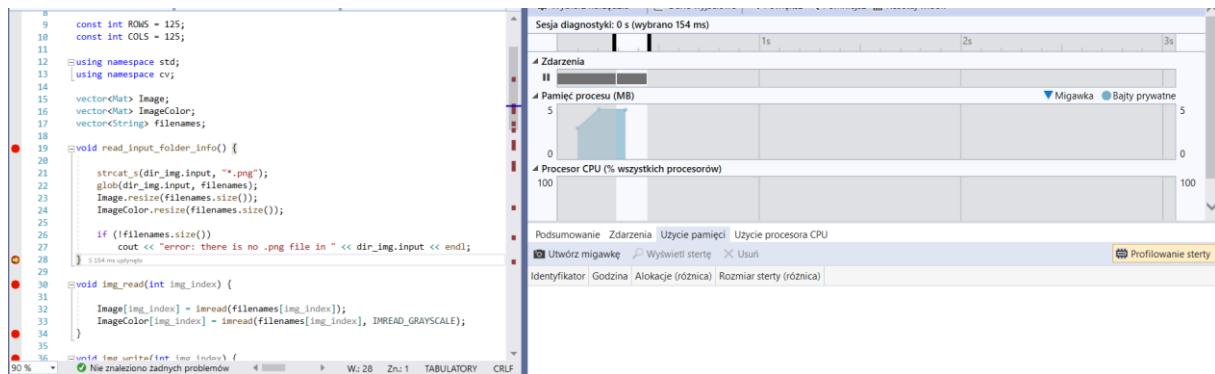
Edge_detecting(Mat Img), która jest funkcją z biblioteki OpenCV

Próba podjęta w celu zredukowania czasu trwania najdłuższych funkcji:

Zastosowano przetwarzanie wielowątkowe, które miało na celu skrócenie czasu wykonywania się funkcji img_read(), img_write() oraz Edge_detecting().

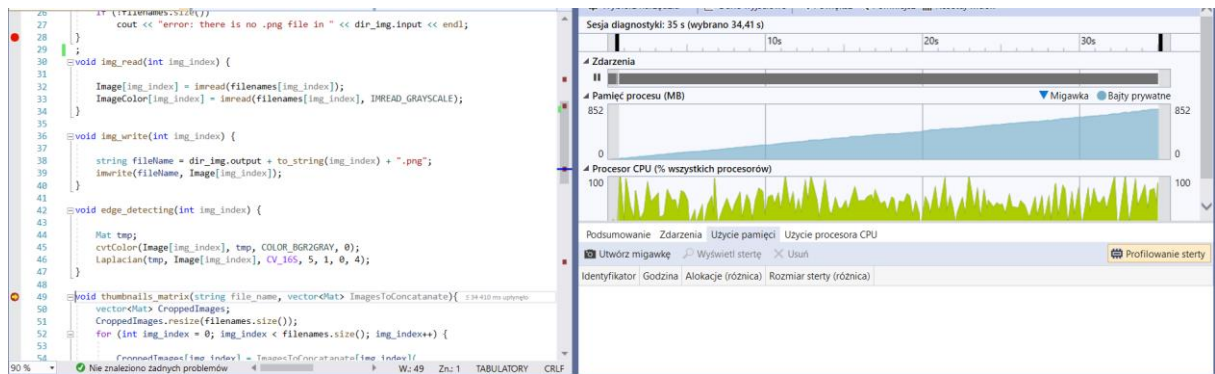
Poniżej przedstawiono wyniki:

Czas inicjalizacji programu:



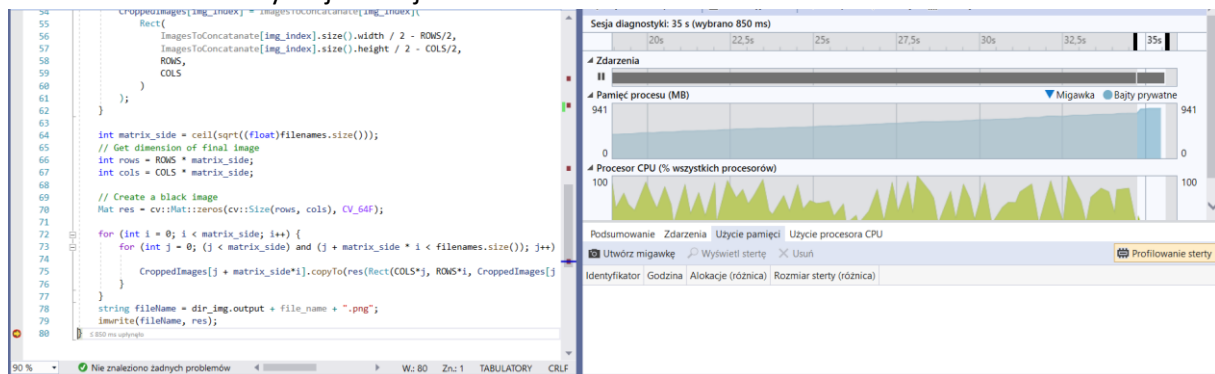
Czas trwania: 154ms

Wczytywanie obrazu, przetwarzanie obrazu i zapis obrazu:



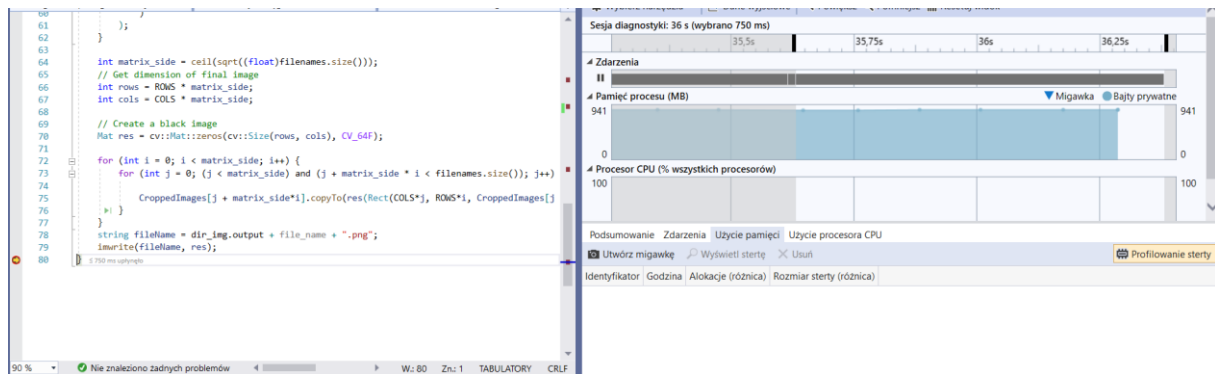
Czas trwania: 34,41ms

Generowanie macierzy wejściowej:



Czas trwania: 850ms

Generowanie macierzy wyjściowej:



Czas trwania: 750ms

Łączny czas wykonywania programu wraz z generowaniem macierzy wejściowej i wyjściowej po optymalizacji wynosi:

36,164s

Podsumowując, stosując multithreading łączny czas wykonywania funkcji `img_read()`, `img_write()` oraz `Edge_detecting()` został zmniejszony z wartości 53,53s do wartości 34,41s co daje 19,11s skrócenia czasu wykonywania się funkcji.