Implementacja zorientowanego filtra Gaussa

Akceleracja Algorytmów Wizyjnych

Jan Rosa 26.03.2025

Wyniki konwolucji

Figure 1: Obraz wejściwoy 256x256

Figure 2: Wynik konwolucji nieseparowanej 256x256

Figure 3: Wynik konwolucji separowanej 256x256

Figure 4: Obraz wyjściowy 1024x1024

Figure 5: Obraz wyjściowy 4096x4096

Wyznaczone współczynniki

```
float Hg[HV_DIM * HH_DIM] =
      {
            0, 0, 0, 0, 0,
            0.003429189170273, 0.049352602812964, 0.120046527317579,
0.049352602812964, 0.003429189170273,
            0.008341247025338, 0.120046527317579, 0.292004228746064,
0.120046527317579, 0.008341247025338,
            0.003429189170273, 0.049352602812964, 0.120046527317579,
0.049352602812964, 0.003429189170273,
            0, 0, 0, 0, 0};

float Hv[HV_DIM] = {
      0,
      -0.222222165955144,
      -0.540538852976128,
      -0.222222165955144,
      0};

float Hh[HH_DIM] = {
      -0.015431355173476, -0.222086768891117, -0.540209509711154,
-0.222086768891117, -0.015431355173476};
```

Tabela czasów wykonania

Rozmiar obrazu	Nieseparowalny GPU (s)	Separowalny GPU (s)	Nieseparowalny CPU (s)	Separowalny CPU (s)	MATLAB (s)
256x256	0.000138184	0.0000319824	0.00191797	0.00179004	0.018975
1024x1024	0.000691162	0.00011499	0.023926	0.0260598	0.002625
4096x4096	0.00760205	0.00138403	0.341359	0.368167	0.011424

Zastosowania

Filtr Gaussa o nierównych wymiarach (np. 3x5) jest używany w zadaniach wymagających kierunkowego rozmycia lub wykrywania specyficznych struktur w obrazie:

- **Redukcja szumu** – asymetryczne filtry skuteczniej usuwają szumy w jednym kierunku, np. w obrazach medycznych lub satelitarnych.
- **Wykrywanie krawędzi** – filtry szerokie (np. 1x5) podkreślają poziome krawędzie, a wysokie (np. 5x1) pionowe.
- **Skalowanie obrazów** – wygładzają detale podczas zmniejszania obrazu, zapobiegając aliasingowi.
- **Efekty wizualne** – stosowane w grafice komputerowej do kierunkowego rozmycia, np. efektu ruchu.

Omówienie uzyskanych wyników

Analizując wyniki konwolucji z użyciem zorientowanego filtra Gaussa, można zauważyć znaczące różnice w czasach wykonania w zależności od metody oraz platformy obliczeniowej (GPU vs. CPU).

Porównanie metod separowalnej i nieseperowalnej

- Filtry separowalne (rozkładane na operacje 1D) wykazują **znacznie krótszy czas wykonania** w porównaniu do filtrów nieseperowalnych.
- Różnica jest szczególnie widoczna na GPU, gdzie **dla obrazu 4096x4096 separowalna konwolucja trwa ~5,5x krócej niż nieseperowalna** (1.38403 ms vs. 7.60205 ms).
- Na CPU różnice są mniej wyraźne, ale nadal separowalność daje niewielką poprawę wydajności.

Porównanie GPU vs. CPU

- **GPU jest zdecydowanie szybsze niż CPU**, co wynika z równoległego przetwarzania operacji konwolucji.
- Dla największego obrazu (4096x4096):
    - **Nieseperowalna konwolucja na CPU trwa aż 341 ms, a na GPU tylko 7.6 ms**, co oznacza **~45x przyspieszenie**.

Separowalna konwolucja na CPU trwa 368 ms, a na GPU tylko 1.38 ms, co oznacza ~267x przyspieszenie.

- Pokazuje to, że dla dużych obrazów **GPU jest najlepszym wyborem** do obliczeń związanych z konwolucją.

Porównanie GPU, CPU i MATLAB

- MATLAB, pomimo wysokiej optymalizacji, nie osiąga wydajności GPU – np. dla **4096x4096 konwolucja trwa 11.4 ms**, czyli **ok. 8x dłużej niż separowalna wersja na GPU**.
- MATLAB może być jednak bardziej wydajny niż CPU w niektórych przypadkach (np. dla 1024x1024 separowalna wersja MATLAB trwa krócej niż na CPU).

Wnioski

1. **Separowalne filtry są znacznie szybsze** – warto je stosować, jeśli filtr można rozłożyć na operacje 1D.
2. **GPU zapewnia ogromne przyspieszenie** w stosunku do CPU, szczególnie przy dużych obrazach.
3. **MATLAB jest wydajniejszy od CPU, ale nie dorównuje GPU**, co sugeruje, że używa dobrze zoptymalizowanych algorytmów, ale nie wykorzystuje pełnego potencjału akceleracji sprzętowej.