

Politechnika Śląska
Wydział Automatyki, Elektroniki i Informatyki

Programowanie Komputerów

Operacje morfologiczne na obrazie

autor	Aleksander Augustyniak
prowadzący	mgr inż. Marek Kokot
rok akademicki	2019/2020
kierunek	informatyka
rodzaj studiów	SSI
semestr	2
termin laboratorium	środa, 11:30 – 13:00
sekcja	druga
termin oddania sprawozdania	2020-09-18

1 Treść zadania

Napisać program umożliwiający przeprowadzenie operacji morfologicznych obrazu zawartego w pliku BMP. Element strukturalny powinien być zdefiniowany przez użytkownika poprzez dostarczenie odpowiedniego pliku tekstowego. Program powinien działać dla obrazów czarno-białych. Program powinien umożliwić wykonanie następujących operacji: dylatacji, erozji, otwarcia oraz domknięcia. [?]

2 Analiza zadania

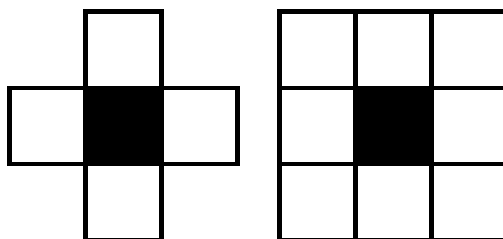
Zadanie polega na cyfrowej obróbce czarno-białego obrazu, używając operacji morfologicznych, które tworzą podstawę zaawansowanych systemów rozpoznających kształty, odległości pomiędzy figurami, szkielet danego kształtu, etc.

Biorąc pod uwagę, że w obrazie występują wyłącznie dwa kolory (biały i czarny), dla ułatwienia i skrócenia pisowni można danym pikselom przypisać binarny stan, w którym się znajdują. Piksel czarny można oznaczyć binarnym 0 - ze względu na zerowe składowe kolorów, z których się składa; oraz - analogicznie – piksel biały, jako binarne 1 ze względu na maksymalną, posiadaną przezeń wartość poszczególnych składowych koloru.

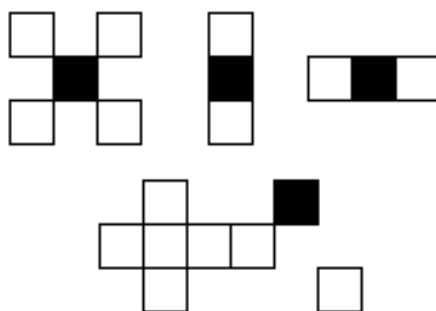
Pracując nad projektem, w którym występują dwa kolory, można wprowadzić założenie, że kolor czarny reprezentuje pierwszy plan, a biały – tło.

2.1 Element strukturalny

Warunkowanie wykonania odpowiedniej modyfikacji na pikselach obrazu następuje przy pomocy elementu strukturalnego – maski, która jest *przykładana* do każdego z nich w sposób zgodny z punktem głównym elementu strukturalnego. Element strukturalny posiada tylko jeden punkt główny. Ponadto posiada również informacje o tym, które punkty względem punktu głównego podlegają sprawdzeniu na obrazie.



Rysunek 1: Najpopularniejsze elementy strukturalne. Od lewej – element strukturalny o sąsiedztwie czterospójnym (von Neumanna) i ośmiospójne (Moore’a) – osiem pikseli sąsiaduje z punktem głównym elementu strukturalnego.



Rysunek 2: Inne, niestandardowe elementy strukturalne.

2.2 Struktury danych

Plik BMP jest plikiem binarnym, którego bajty są usytuowane w odpowiedniej kolejności. Taką strukturę można nazwać stałą, bądź statyczną – rozmiary poszczególnych zawartości są z góry ustalone i nie podlegają jakiegokolwiek zmianie. Jakakolwiek zmiana kolejności spowodowałoby trwałe uszkodzenie pliku.

Numer bajtu	Zawartość	Rozmiar [B]
0	Pierwsza litera identyfikatora pliku.	1
1	Druga litera identyfikatora pliku.	1
2	Rozmiar całego pliku.	4
6	Zarezerwowany dla innych aplikacji.	4
10	Numer bajtu, który zaczyna macierz.	2
14	Rozmiar struktury nagłówka DIB.	4
18	Szerokość obrazu.	4
22	Wysokość obrazu.	4
26	Liczba płaszczyzn kolorów.	2
28	Ilość bitów na piksel.	2
30	Metoda kompresji.	4
34	Powierzchnia obrazu.	4
38	Rozdzielczość pozioma.	4
42	Rozdzielczość pionowa.	4
46	Ilość kolorów w palecie.	4
50	Ilość ważnych kolorów w palecie.	1
51	Flaga wyznaczająca, czy ma nastąpić rotacja palet.	1
52	—	2

Rysunek 3: Ułożenie struktury nagłówka pliku i nagłówka DIB w pliku o rozszerzeniu BMP.

Oprócz tego, program przechowuje dane posługując się m.in. dynamiczną listą jednokierunkową (w której przechowywany jest element strukturalny).

2.3 Algorytmy

Program składa się z nietrudnych algorytmów. Ich złożoność zwiększa się proporcjonalnie co do wielkości elementu strukturalnego i wielkości obrazu, który poddajemy obróbce. Sczytując piksele obrazu wejściowego, program *sprawdza*, czy wszystkie składowe koloru piksela, w danej chwili sprawdzanego przez program, są równe zeru, jeżeli tak - żadna modyfikacja nie jest

podejmowana - w przeciwnym wypadku wszystkie składowe są zwiększane do maksymalnej wartości.

Istotnym - w zasadzie najważniejszym - algorytmem jest algorytm *sprawdzający* sąsiedztwo piksela modyfikowanego i warunkujący, czy ów piksel ma zostać zmodyfikowany. Algorytm ten działa na zasadzie *przykładania* elementu strukturalnego do każdego, pojedynczego piksela obrazu i porównywania z otaczającymi piksel modyfikowany pikselami.

3 Specyfikacja zewnętrzna

Aby uruchomić program należy podać dokładne lokalizacje odpowiednich plików w linii poleceń w odpowiedniej kolejności, oddzielając poszczególne argumenty znakiem spacji:

`input output sieve char`

Gdzie:

`input` - obraz wejściowy o rozszerzeniu BMP.

`output` - obraz wyjściowy o rozszerzeniu BMP.

`sieve` - plik tekstowy, zawierający element strukturalny (o rozszerzeniu TXT).

`char` - znak operacji morfologicznej, która ma zostać wykonana.

Plik tekstowy, zawierający element strukturalny musi spełniać pewne założenia:

- Zawierać punkt główny, oznaczany małą literą *o*. W przypadku, gdy plik tekstowy zawiera więcej, niż jeden punkt główny, następne wystąpienia będą traktowane jak punkt sąsiedni punktu głównego.
- Zawierać przynajmniej jeden punkt sąsiedni, oznaczany - znakiem minus.

Niedostosowanie się do przynajmniej jednego z warunków skutkuje zakończeniem pracy programu i wyjściem.

Przykładowa zawartość pliku tekstowego zawierającego element strukturalny.

-	- - -	-	-
- O -	- O -	O	
-	- - -	-	-

Program, czytując poszczególne, podane przez użytkownika argumenty, sprawdza, czy każdy z plików rzeczywiście istnieje, i czy dostęp do niego nie jest niemożliwy do uzyskania. Jeżeli przynajmniej jeden z plików nie zostanie wczytany do pamięci programu, program kończy swoją pracę wypisując odpowiedni kod błędu.

Kody błędów:

- 98 - Do programu nie została dostarczona wystarczająca ilość argumentów do poprawnego działania.
- 99 - Wczytany plik wejściowy nie jest plikiem mapy bitowej.
- 100 - Nie udało się wczytać elementu strukturalnego.
- 101 - Nie udało się otworzyć pliku wejściowego.
- 102, 103 - Nie udało się poprawnie wczytać nagłówka pliku.
- 104 - Nie udało się zarezerwować odpowiedniej ilości pamięci dla struktury elementu strukturalnego.
- 105 - Nie udało się zagospodarować odpowiedniej ilości pamięci dla współrzędnych punktu modyfikowanego (piksela modyfikowanego).
- 106 - Nie udało się zaalokować wymaganej ilości pamięci do przechowania składowych koloru poszczególnych pikseli pliku wejściowego.
- 107 - Nie udało się zaalokować wymaganej ilości pamięci do przechowania składowych koloru poszczególnych pikseli pliku wyjściowego.
- 108 - Nie udało się utworzyć, bądź otworzyć istniejący plik wejściowy.

4 Specyfikacja wewnętrzna oraz szczegółowy opis typów i funkcji

Szczegółowy opis typów i funkcji znajduje się w załączniku.

4.1 Ogólna struktura programu

Program realizuje funkcje w odpowiedniej kolejności:

1. Sprawdzenie, czy została podana odpowiednia ilość argumentów.
2. Otwarcie pliku wejściowego.

3. Załadowanie do struktury nagłówka pliku wejściowego w celu odczytania kluczowych danych, tj. sprawdzenie, czy plik wejściowy jest plikiem mapy bitowej.
4. Otwarcie pliku, zawierającego element strukturalny.
5. Wczytanie elementu strukturalnego do struktury elementu strukturalnego.
6. Zapisanie do pamięci współrzędnych piksela modyfikowanego.
7. Zajęcie odpowiedniej ilości pamięci na obraz wyjściowy.
8. Wykonanie operacji morfologicznej i zapisanie wyniku do struktury wyjściowej.
9. Przepisanie danych z nagłówka pliku wejściowego do nagłówka pliku wyjściowego.
10. Zwolnienie całej pamięci zajętej przez program.

5 Testowanie

Program został przetestowany na obrazach różnorodnych – tych spełniających założenia zadania, jak i tych, które nie do końca pokrywają się z wymaganiami (lecz nieuszkodzonymi). Testy były wykonane na obrazach o dużej powierzchni pikseli.

Program został sprawdzony pod kątem wycieków pamięci.

Program przetestowany na komputerze z systemem Windows 10.

5.1 Wyniki

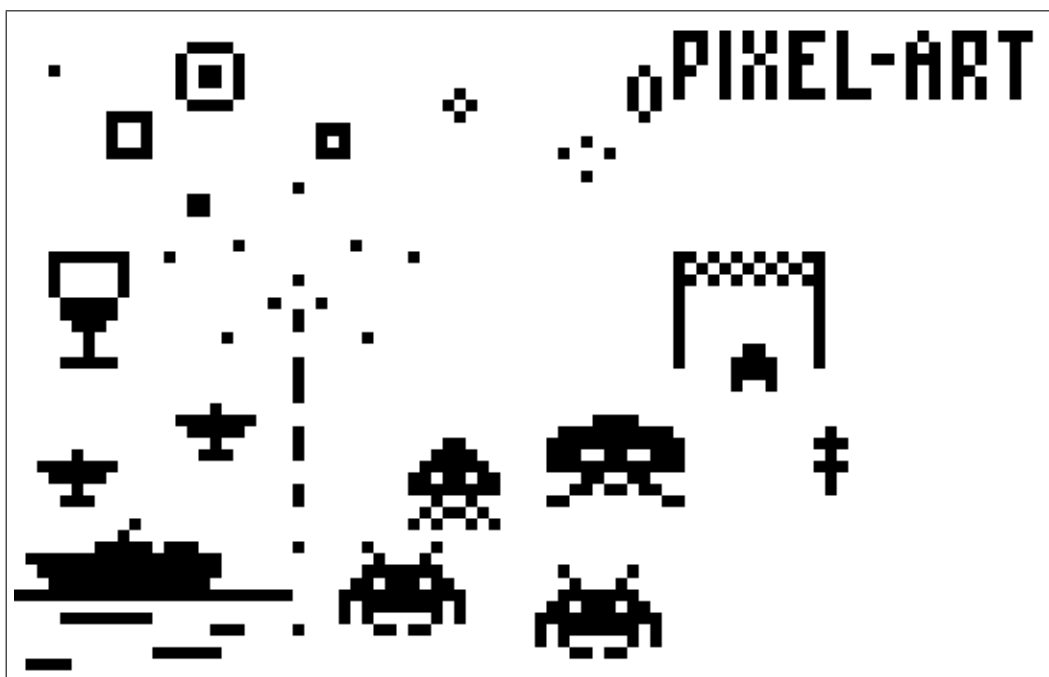
Poniżej przedstawione zostały wyniki programu dla sześciu różnych elementów strukturalnych i trzech różnych obrazów wejściowych.

W lewym górnym rogu: erozja.

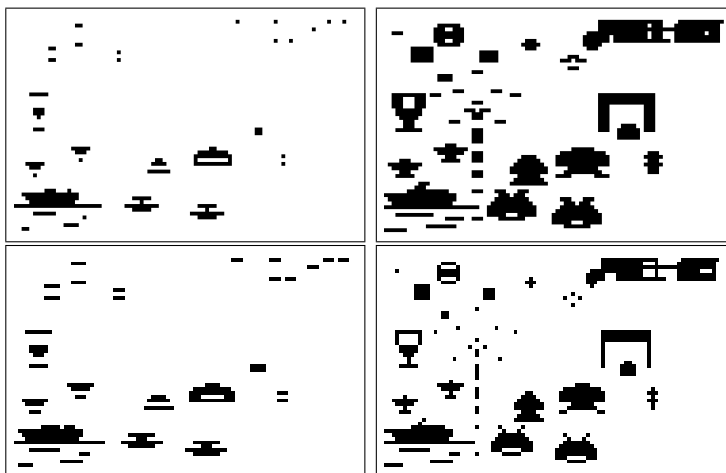
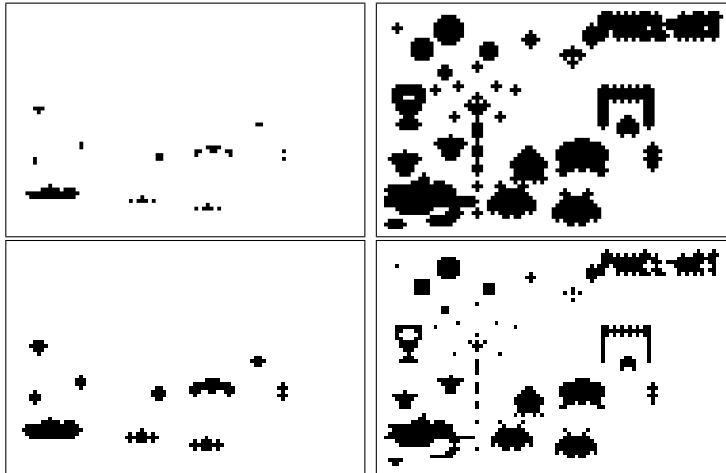
W prawym górnym rogu: dylatacja.

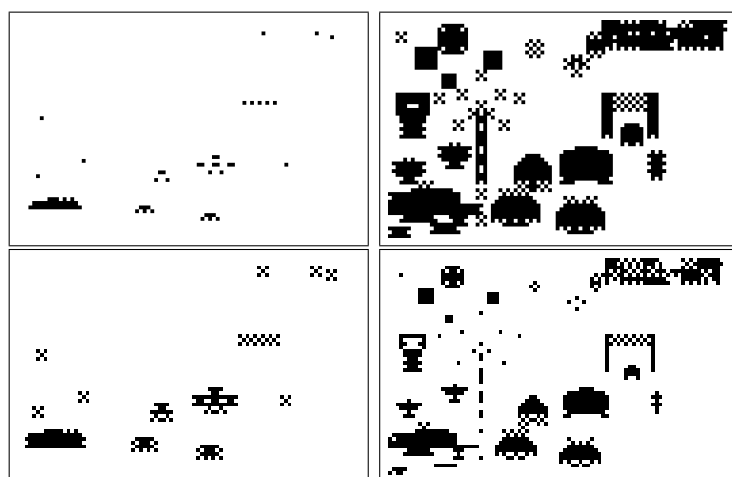
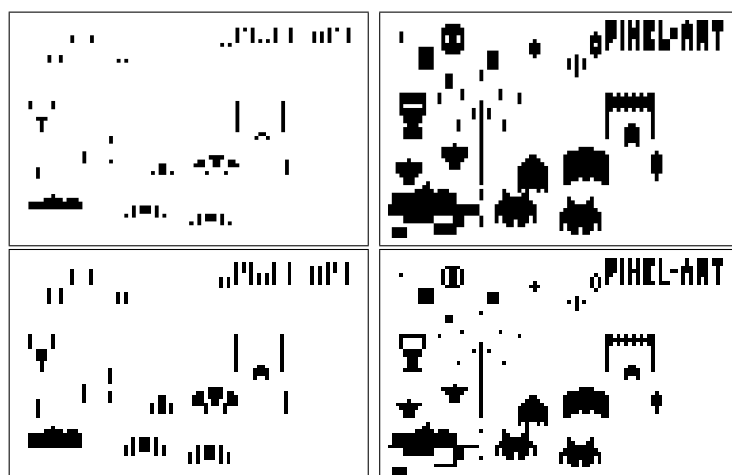
W lewym dolnym rogu: otwarcie.

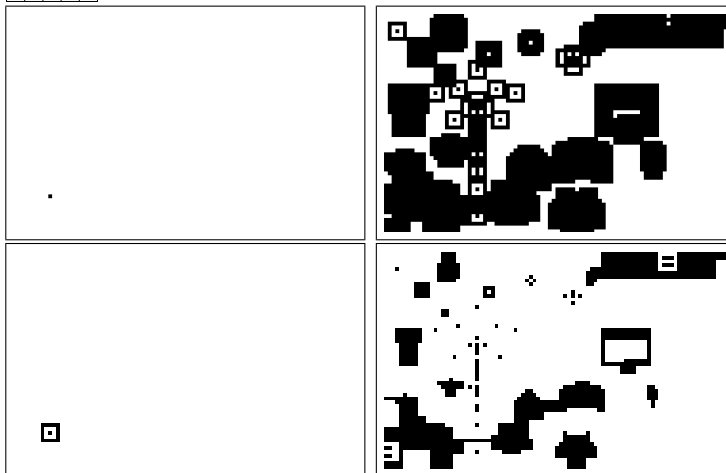
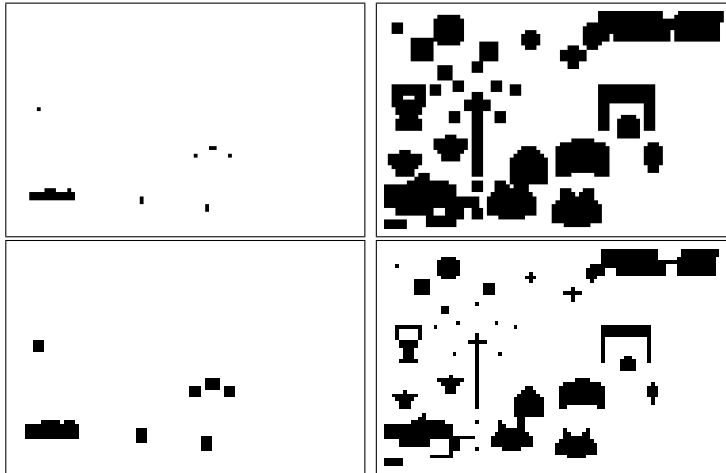
W prawym dolnym rogu: domknięcie.

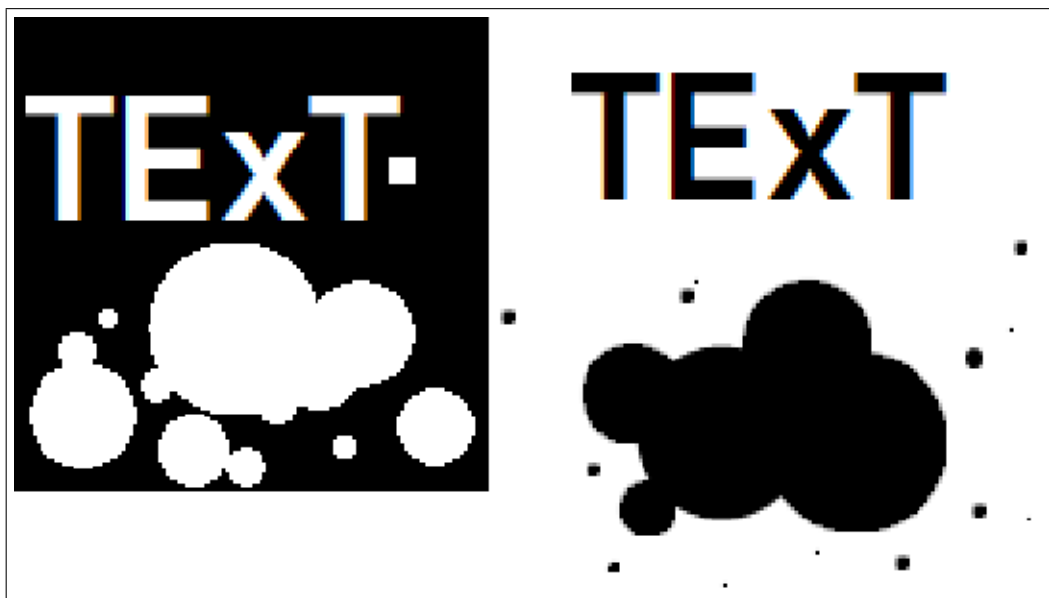


Rysunek 4: Obraz wejściowy numer 1.

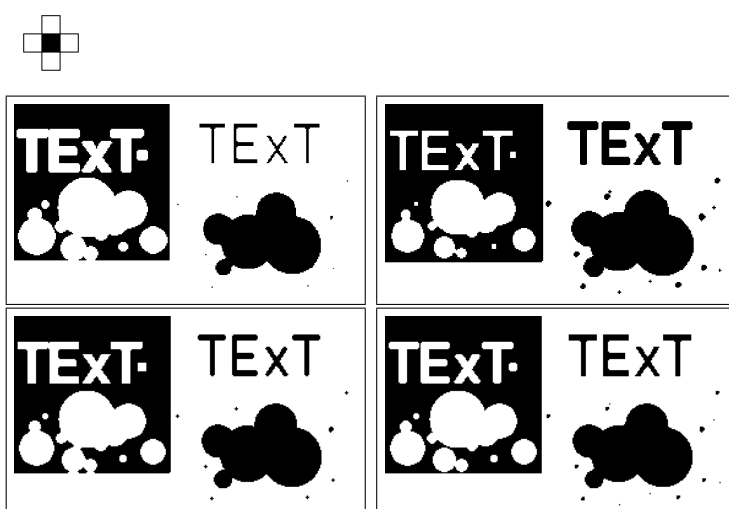


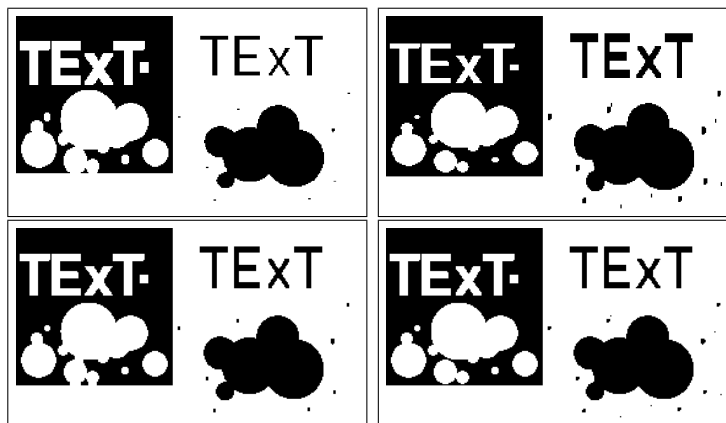
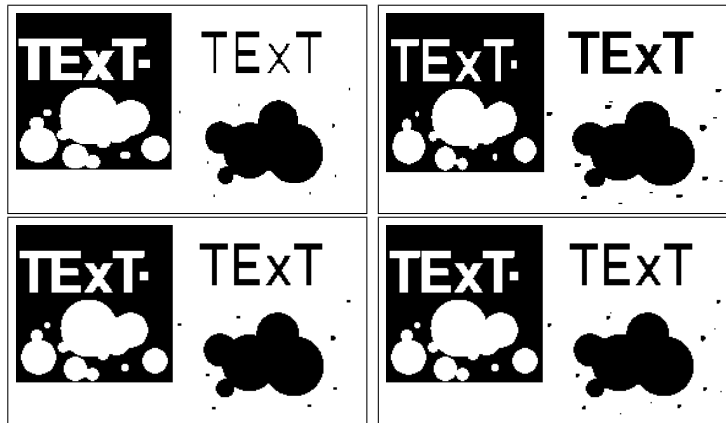


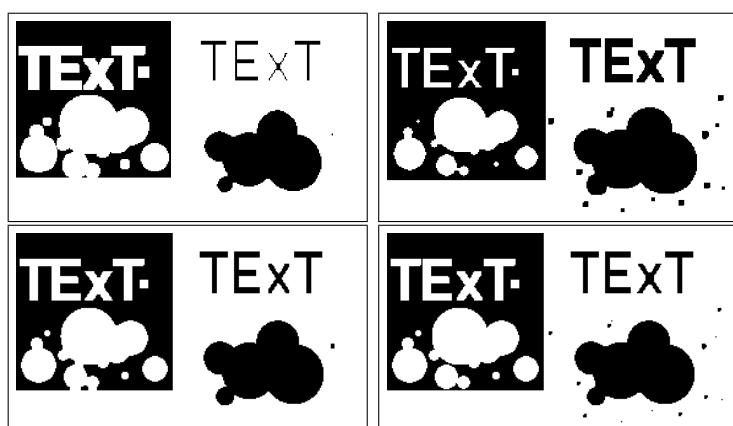
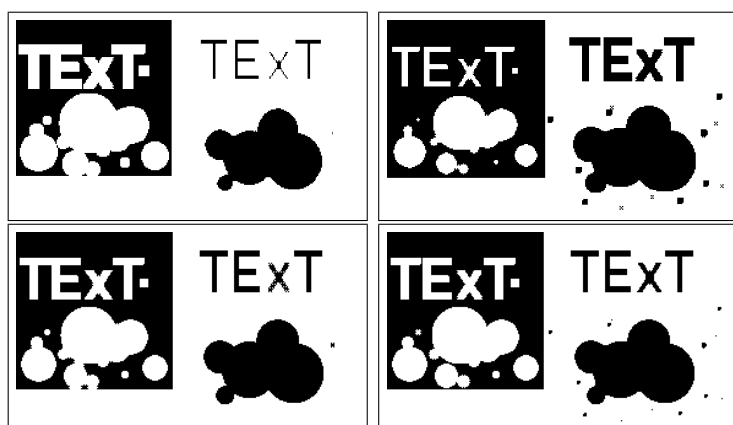


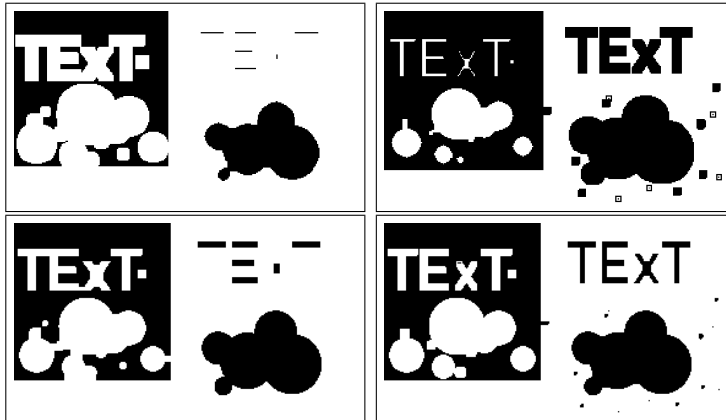
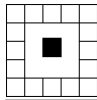


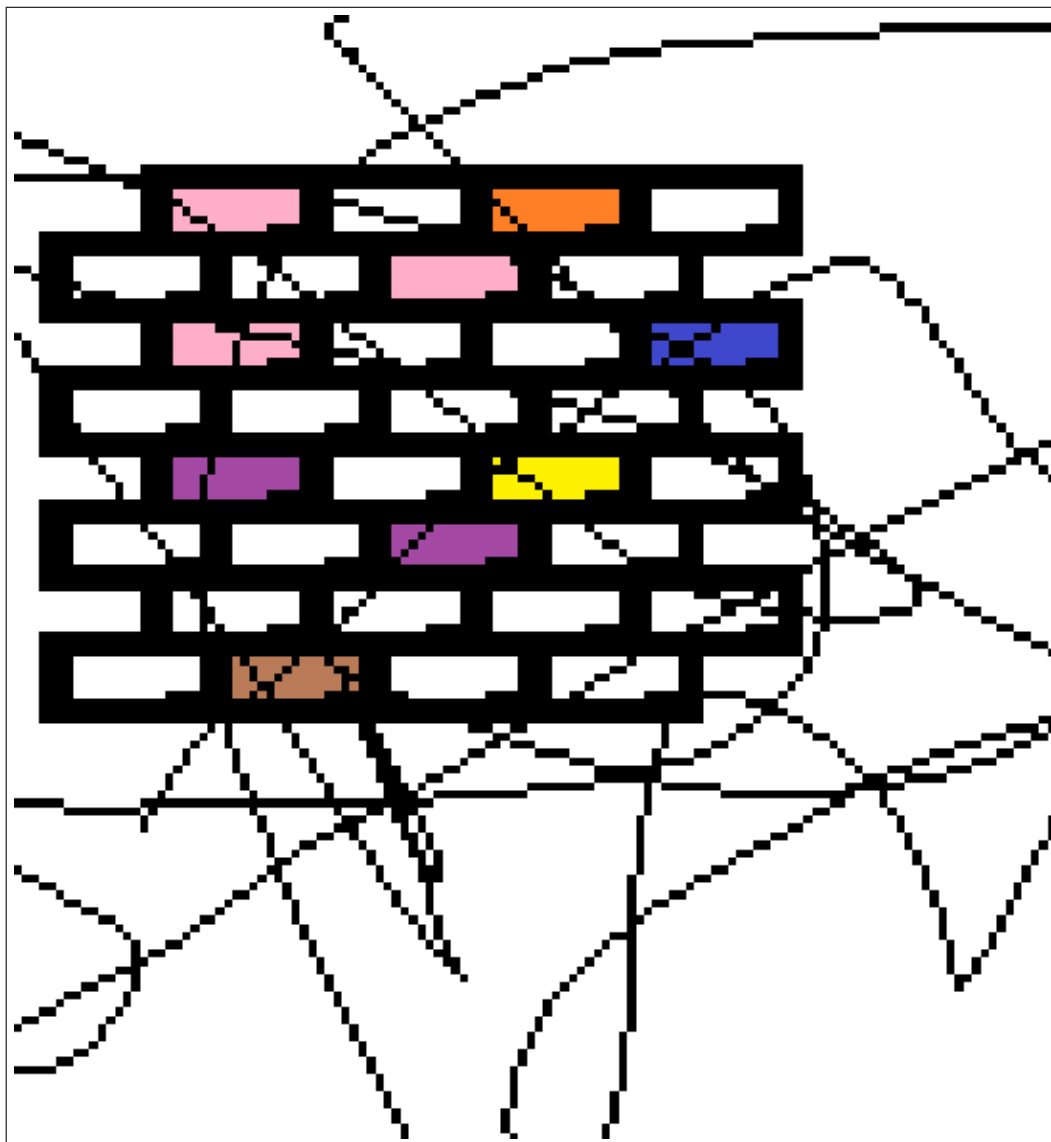
Rysunek 5: Obraz wejściowy numer 2.



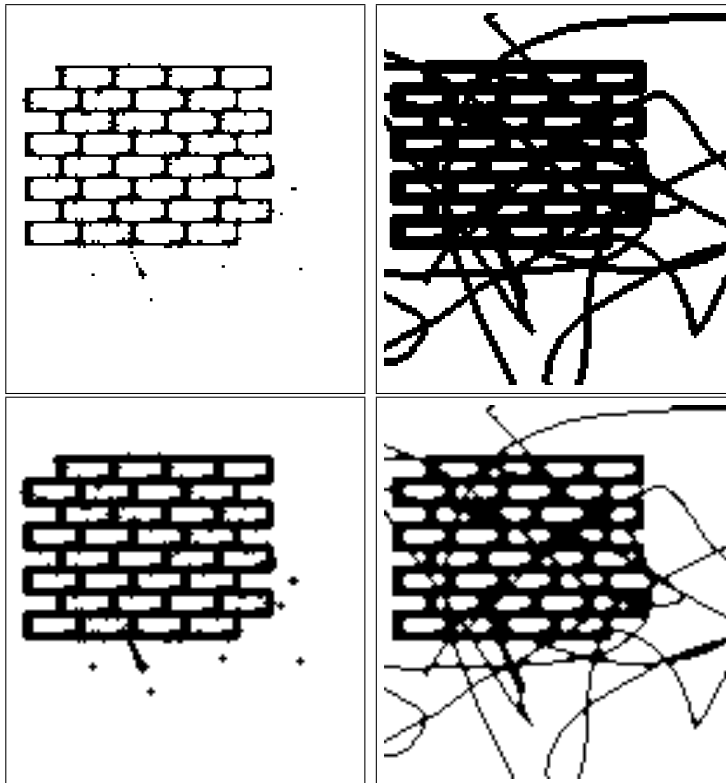


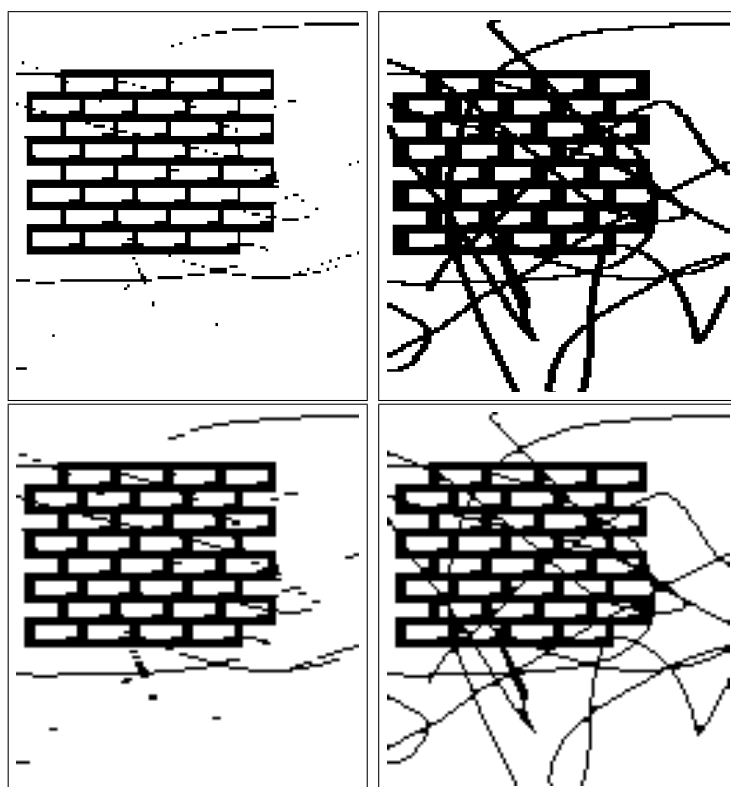


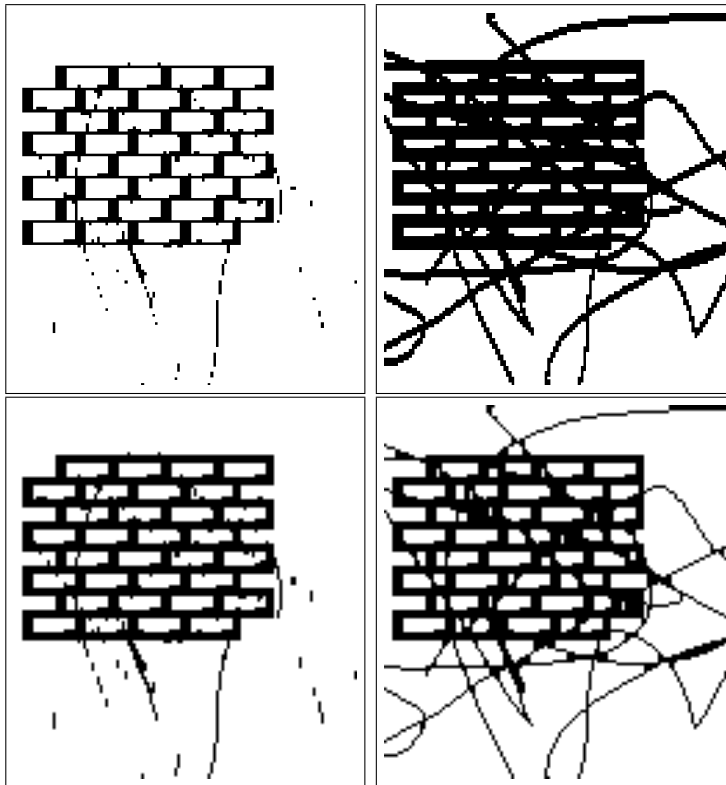


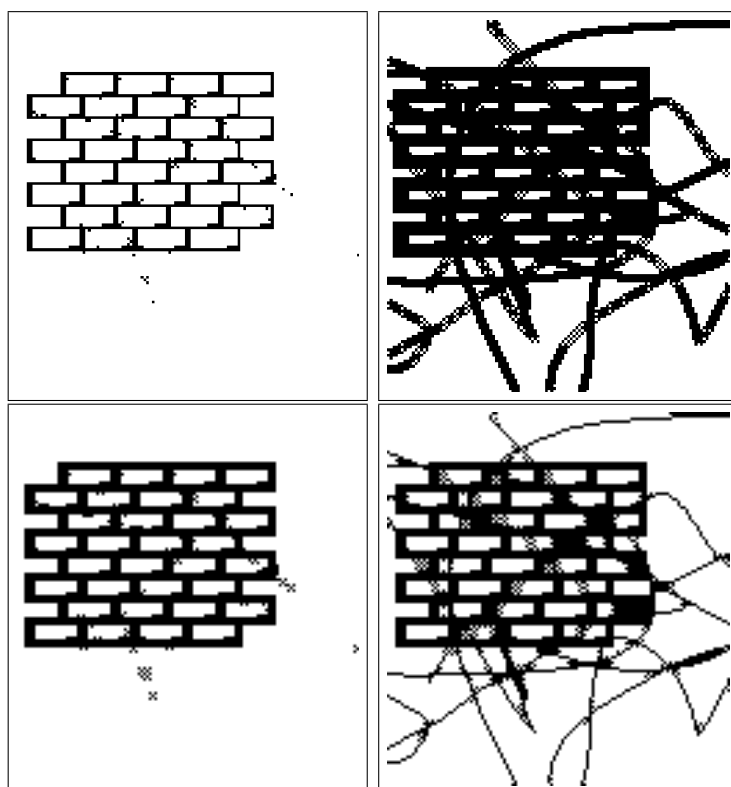


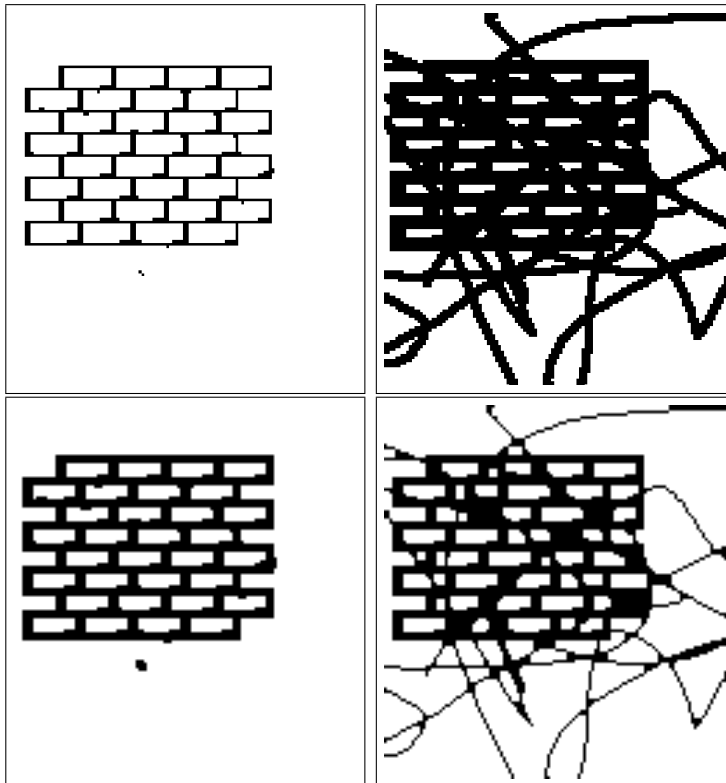
Rysunek 6: Obraz wejściowy numer 3.

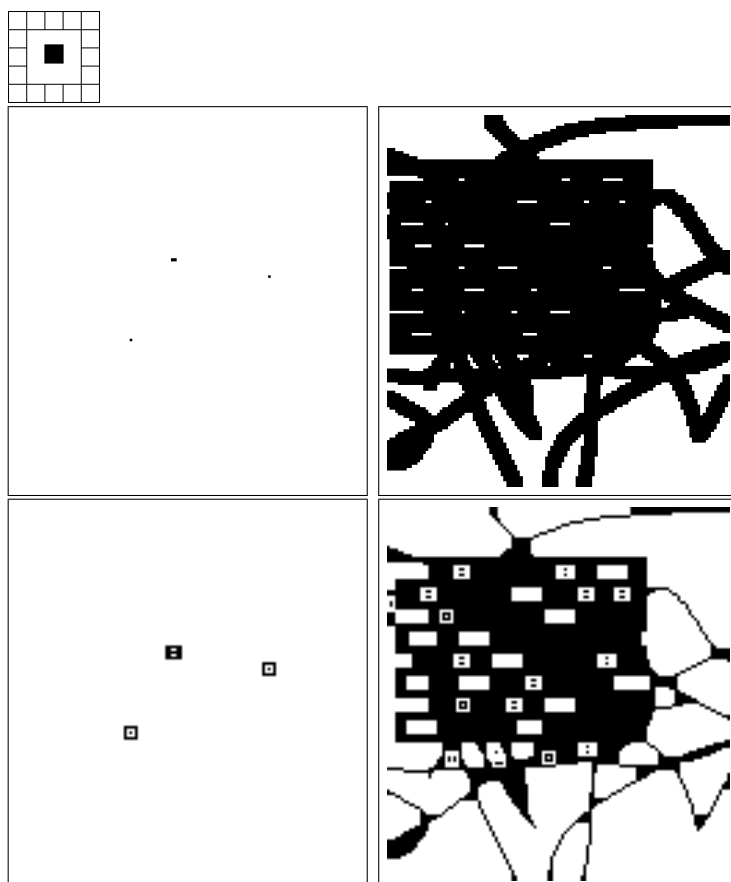












6 Wnioski

Napisanie programu, który wykonuje podstawowe operacje morfologiczne należało do prostych. Nie miałem trudności z analizą zadania, ze względu na łatwość w niemalże natychmiastowym zauważeniu efektów pracy. Zadany temat uważam za bardzo interesujący ze względu na możliwe wykorzystanie go do rozwiązywania bardziej skomplikowanych problemów, związanych z cyfrowym przetwarzaniem obrazów.

Literatura

Dodatek
Szczegółowy opis typów
i funkcji