

Instrukcja	Zaawansowane przetwarzanie obrazów
2	Temat: Operacje arytmetyczne i morfologiczne na obrazie Przygotował: mgr inż. Tomasz Michno

1 Wstęp

1.1 Operacje arytmetyczne

Są to operacje, które stosują operatory arytmetyczne oraz logiczne na dwóch lub więcej obrazach.

1.1.1 Dodawanie (Addition) +

Operator ten ma zastosowanie najczęściej do dwóch obrazków, które muszą mieć te same rozmiary. Następnie wartości ich pikseli są sumowane i wstawiane do trzeciego, wynikowego obrazu:

$$W(x, y) = I_1(x, y) + I_2(x, y),$$

gdzie:

W - wartość piksela wynikowego obrazu

I_1 - wartość piksela w pierwszym obrazie

I_2 - wartość piksela w drugim obrazie

1.1.2 Odejmowanie (Subtraction) -

Obrazki muszą mieć te same rozmiary. Następnie wartości ich pikseli są odejmowane i wstawiane do trzeciego, wynikowego obrazu:

$$W(x, y) = |I_1(x, y) - I_2(x, y)|$$

1.1.3 Mnożenie (Subtraction) *

Obrazki muszą mieć te same rozmiary. Następnie wartości ich pikseli są mnożone przez siebie i wstawiane do trzeciego, wynikowego obrazu:

$$W(x, y) = I_1(x, y) * I_2(x, y)$$

1.1.4 AND &

Obrazki muszą mieć te same rozmiary. Następnie wartości ich pikseli wykonywana jest operacja AND:

$$W(x, y) = I_1(x, y) \& I_2(x, y)$$

1.1.5 OR |

Obrazki muszą mieć te same rozmiary. Następnie wartościach ich pikseli wykonywana jest operacja OR:

$$W(x, y) = I_1(x, y) | I_2(x, y)$$

1.1.6 Negacja/Inwersja !

Wykonywane na jednym obrazku:

$$W(x, y) = WHITE - I(x, y),$$

gdzie:

WHITE - kolor biały (255, 255, 255)

1.2 Operacje morfologiczne

Operacje morfologiczne są operacjami, które zmieniają obraz np. zmniejszając krawędzie. Należą do nich: erozja, dylatacja, otwarcie i zamknięcie. Zostaną przedstawione funkcje OpenCV, które służą do ich zastosowania na obrazie.

1.2.1 Erozja

Interfejs C:

```
void cvErode( const CvArr* src, CvArr* dst, IplConvKernel* element=NULL,
int iterations)
```

gdzie:

src - obraz źródłowy

dst - obraz wynikowy

element - element strukturalny, jeśli podano NULL, zostanie użyty domyślny element 3x3

iterations - liczba iteracji

Interfejs C++:

```
cv::erode ( InputArray src, OutputArray dst, InputArray kernel, Point
anchor = Point(-1,-1), int iterations = 1, int borderType = BORDER_CONSTANT,
const Scalar & borderValue = morphologyDefaultBorderValue() )
```

gdzie:

src - obraz źródłowy (typu cv::Mat)

dst - obraz wynikowy (typu cv::Mat)

kernel - element strukturalny

iterations - liczba iteracji

Więcej informacji w dokumentacji opencv: https://docs.opencv.org/3.4/d4/d86/group__imgproc__filter.html#gaeb1e0c1033e3f6b891a25d0511362aeb
Interfejs Python: https://docs.opencv.org/3.4/db/df6/tutorial_erosion_dilatation.html

1.2.2 Dylatacja

```
void cvDilate( const CvArr* src, CvArr* dst, IplConvKernel* element=NULL,
int iterations)
```

gdzie:

src - obraz źródłowy

dst - obraz wynikowy

element - element strukturalny, jeśli podano NULL, zostanie użyty domyślny element 3x3

iterations - liczba iteracji

Odpowiedniki w interfejsach C++: `cv::dilate` i Python: `cv2.dilate`.

1.2.3 Otwarcie

Do operacji otwarcia można zastosować poniższą formułę:

```
dilate(erode(image))
```

1.2.4 Zamknięcie

Do operacji zamknięcia można zastosować poniższą formułę:

```
erode(dilate(image))
```

2 Zadania

1. Napisz program, który będzie wykonywał przedstawione operacje arytmetyczne na dwóch różnych obrazach (dla negacji na jednym). Funkcje należy napisać własnoręcznie, używając wyłącznie operacji na pikselach z OpenCV (nie wolno korzystać z operacji arytmetycznych dostarczanych przez tę bibliotekę). Zaobserwuj, co się stanie, gdy podamy na wejście dwa takie same obrazy.
2. Napisz program, który będzie wykonywał przedstawione operacje morfologiczne z wykorzystaniem OpenCV.