

08.09.2020

Mariusz Przysiężniuk

Analiza i przetwarzanie obrazów cyfrowych

Projekt I

# Opis

Celem projektu jest zaimplementowanie przypisanych przekształceń na obrazach.

Przekształcenia, które zostały przypisane dla mnie:

1. Przekształcenie afiniczne
2. Filtracja odchylenia standardowego
3. Gradient morfologiczny
4. Usuwanie elementów przeciętych brzegiem

## Użyte technologie

Program został zaimplementowany w języku Python z użyciem biblioteki opencv.

# Uruchomianie programu

1. **python main.py [ --input '/path/to/image' ]** – przekształcenie zostanie wykonane na wskazanym zdjęciu.
2. **python main.py** – przekształcenie zostanie wykonane na zdjęciu zdefiniowanym odgórnie 'F\_dzieciol.png'

Po uruchomieniu programu zostaniemy zapytani o wybranie przekształcenia, które ma zostać wykonane.

```
python main.py

Affine Transformation:
  1 - Translate
  2 - Scale
  3 - Rotate
  4 - Lean
  5 - Matrix
6 - Standard Deviation Filtration
7 - Morphological Gradient (image - erode)
8 - Morphological Gradient (dilate - image)
9 - Morphological Gradient (dilate - erode)/2
0 - Logic

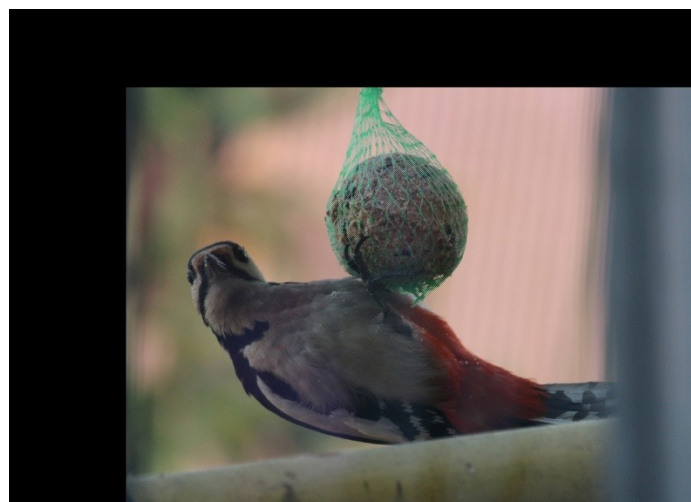
Choose operation: |
```

Po wybraniu operacji, jeśli to konieczne, zostaniemy poproszeni o podanie odpowiednich parametrów.

# Przekształcenie afiniczne

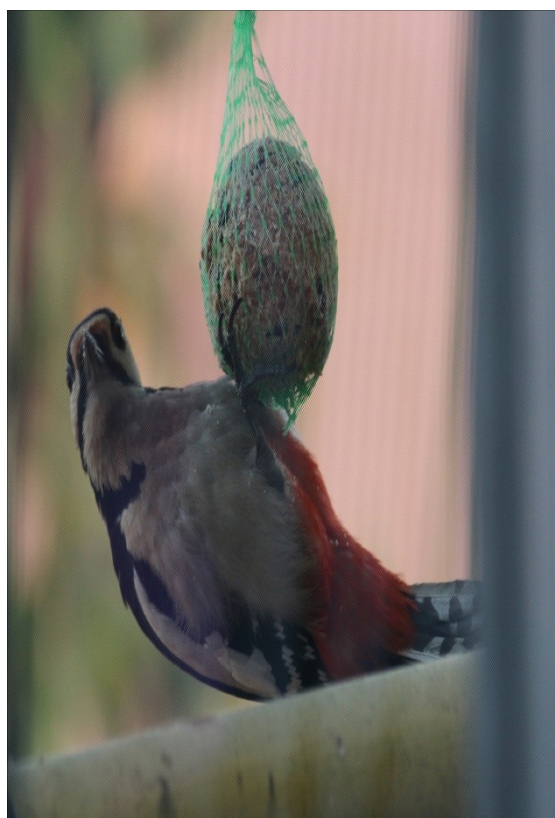
**Translacja** – wynikiem translacji jest przesunięty oraz poszerzony, o zadane parametry, obraz wejściowy

Wyniki dla  $x = 300$ ,  $y = 200$



**Skalowanie** – wynikiem jest obraz z rozmiarem przemnożonym przez zadane współczynniki.

Wyniki dla  $x = 1$ ,  $y = 2$

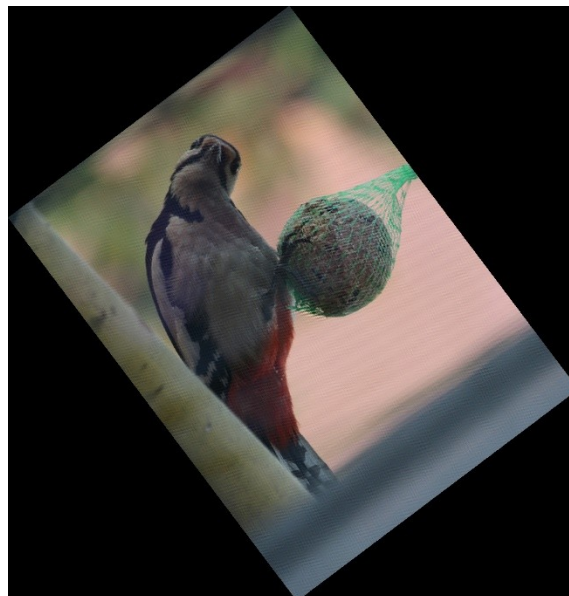


**Rotacja** – zdjęcie jest obracane o zadany kąt.

Użyte wzory:

1.  $x = \cos(\theta) * (x1 - x0) + \sin(\theta) * (y1 - y0)$
2.  $y = -\sin(\theta) * (x1 - x0) + \cos(\theta) * (y1 - y0)$

Wyniki dla 53 stopni

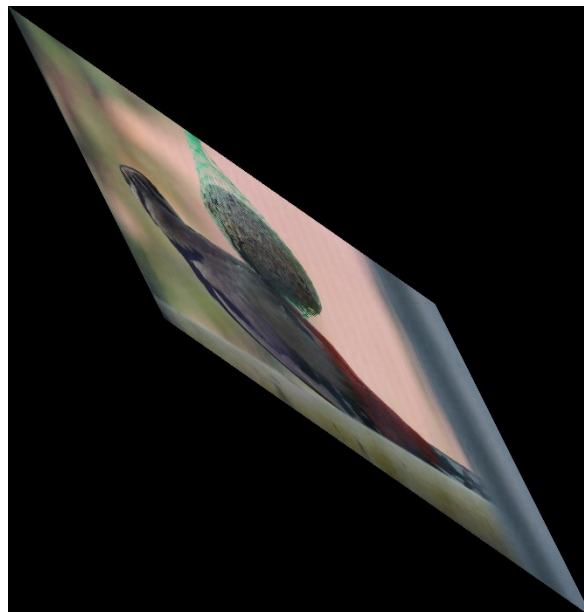


**Pochylenie** – wynikiem jest zdjęcie pochylone o zadane współczynniki

Użyte wzory:

1.  $x = x_0 + a \cdot y_0$
2.  $y = y_0 + b \cdot x_0$

Wyniki dla  $a = 0.5$ ,  $b = 0.7$



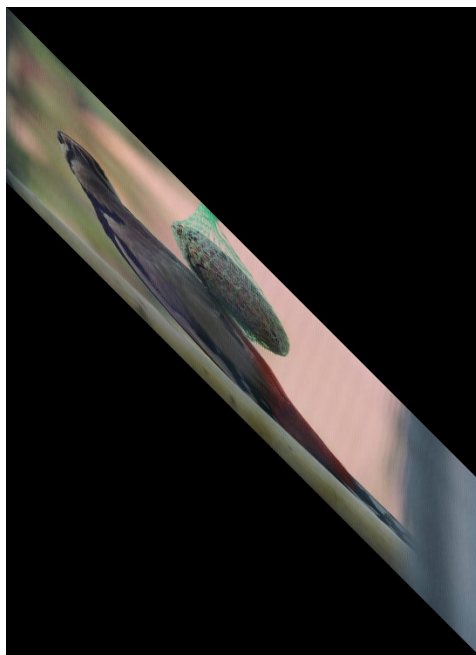
## Przekształcenie o macierz – rozmiar 2x3 [ a, b; c, d; e, f ]

Użyte wzory:

$$1. x = x_0 \cdot a + y_0 \cdot c + e$$

$$2. y = y_0 \cdot b + y_0 \cdot d + f$$

Wyniki dla `[[1, 1], [0, 0.5], [0, 0]]`

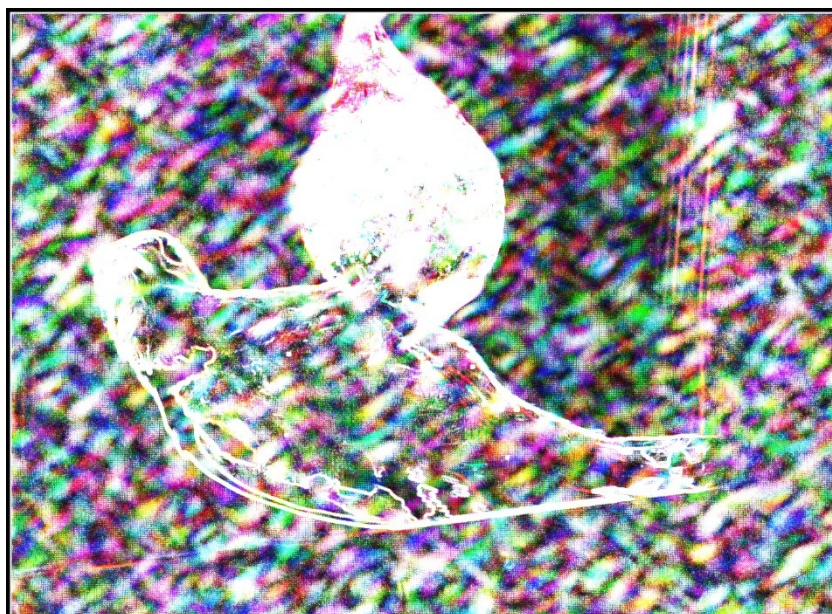




# Filtracja odchylenia standardowego

Obraz zostaje poszerzony o rozmiar maski. Obliczane jest odchylenie standardowe dla każdego piksela przez sąsiednie piksele. Obraz jest normalizowany, aby zakres pikseli  $<0,255>$  nie został przekroczony.

Wyniki dla maski 7



# Gradient morfologiczny

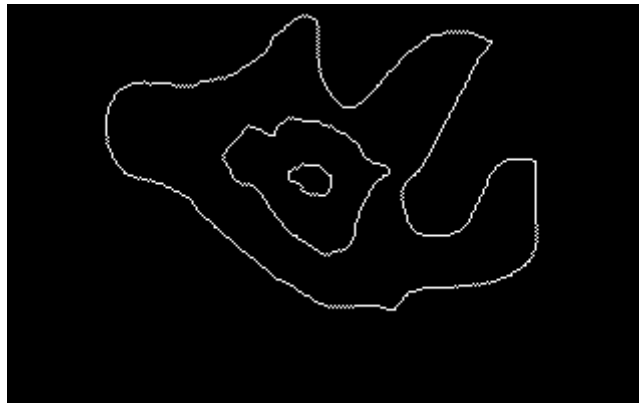
Zaimplementowany 3 metodami:

1. obraz wejściowy – erozja
2. dylatacja – obraz wejściowy
3.  $(\text{dylatacja} - \text{erozja})/2$

Erozja elementem 'plus' o rozmiarze 3x3 – dla każdego piksela znajdującego się w środku elementu 'plus' zmieniana jest wartość na wartość minimalną spośród pikseli, na których element ma wartość true.

Dylatacja działa analogicznie, lecz w jej przypadku brana jest wartość maksymalna.

Wyniki gradientu



# Usuwanie elementów przeciętych brzegiem

Obraz wejściowy jest kopiowany i zapisywany w dodatkowej zmiennej **black**, po czym w **black** zerowane są wszystkie elementy z wyjątkiem tych, które są na krawędziach. Następnie jest robiona rekonstrukcja. Na przemienne wykonywana dylatacja i iloczyn logiczny do momentu, kiedy obraz z początku iteracji jest równy obrazowi po dylatacji i iloczynie. Na końcu robimy różnicę między obrazem wejściowym, a obrazem **black**.

