

# Metody obliczeniowe w nauce i technice 2

Tomasz Zawadzki

8 maja 2019

## Zadanie 1. Analiza obrazów

Niech  $A, B$  oznaczają dwa dyskretne sygnały dwuwymiarowe. Niech  $\sigma_{A,B}$  będzie korelacją (miarą podobieństwa) takich sygnałów zdefiniowaną jako

$$\sigma_{A,B} = \sum_i \sum_j a_{ij} b_{ij}. \quad (1)$$

Niech  $C_{A,B}$  będzie korelacją dwóch sygnałów w funkcji ich wzajemnego przesunięcia

$$C_{A,B}(x, y) = \sigma_{A, T_{(x,y)}(B)} = \sum_i \sum_j a_{ij} b_{x+i, y+j}. \quad (2)$$

Niech  $Q = \text{Rot } 180^\circ(B)$ , czyli  $q_{w-i, h-j} = b_{ij}$ , a zatem korelację można przekształcić do postaci

$$C_{A,B} = A * Q = A * \text{Rot}_{180^\circ}(B), \quad (3)$$

gdzie  $*$  oznacza splot (konwolucję). Korzystając z twierdzenia o splotcie

$$A * B = \mathcal{F}^{-1}[\mathcal{F}(A * B)] = \mathcal{F}^{-1}[\mathcal{F}(A) \cdot \mathcal{F}(B)], \quad (4)$$

korelację można obliczyć przy użyciu szybkiej transformacji Fouriera (FFT) ze złożonością  $O(n \log n)$

$$C_{A,B} = \mathcal{F}^{-1}[\mathcal{F}(A) \cdot \mathcal{F}(\text{Rot}_{180^\circ}(B))], \quad (5)$$

gdzie  $n = wh$ ,  $w$  oraz  $h$  oznaczają odpowiednio szerokość i wysokość obrazu (sygnału dwuwymiarowego),  $\mathcal{F}$  oznacza transformację Fouriera, natomiast  $\cdot$  oznacza *element-by-element product*.

Rozpoznawanie wzorców na obrazie można przeprowadzić wykorzystując metodę badania korelacji obrazu wejściowego (sygnału  $A$ ) z wzorcem (sygnałem  $B$  uzupełnionym zerami do rozmiaru obrazu wejściowego). Punkty o wysokiej korelacji są oznaczane jako prawdopodobne wystąpienia wzorca. Jako kryterium wysokiej korelacji punktu  $(x_0, y_0)$  można przyjąć przypadek, gdy

- korelacja osiąga wartość zbliżoną do maksymalnej z parametrem  $\mu \in (0, 1]$

$$C_{A,B}(x_0, y_0) \geq \mu \cdot \max_{(x,y)} C(x, y), \quad (6)$$

- stosunek różnicy korelacji w punkcie do autokorelacji wzorca  $\sigma_{B,B}$  jest mniejszy niż  $\varepsilon > 0$

$$\left| \frac{C_{A,B}(x_0, y_0)}{\sigma_{B,B}} - 1 \right| < \varepsilon \quad (7)$$

Metody wartości maksymalnej nie należy stosować, jeśli istnieje prawdopodobieństwo, że wzorec nie występuje na obrazie. Współczynniki  $\mu$  oraz  $\varepsilon$  można wyznaczyć empirycznie.

Algorytm rozpoznawania wzorców na obrazie został zaimplementowany w języku Python. Wykorzystano bibliotekę PIL do otwierania plików graficznych, a także funkcje `fft2` oraz `ifft2` z pakietu `numpy.fft`, analogiczne do tych zawartych w instrukcji do laboratorium:


```
C=real(ifft2(fft2(galia).*fft2(rot90(imp,2),h,w))).
```

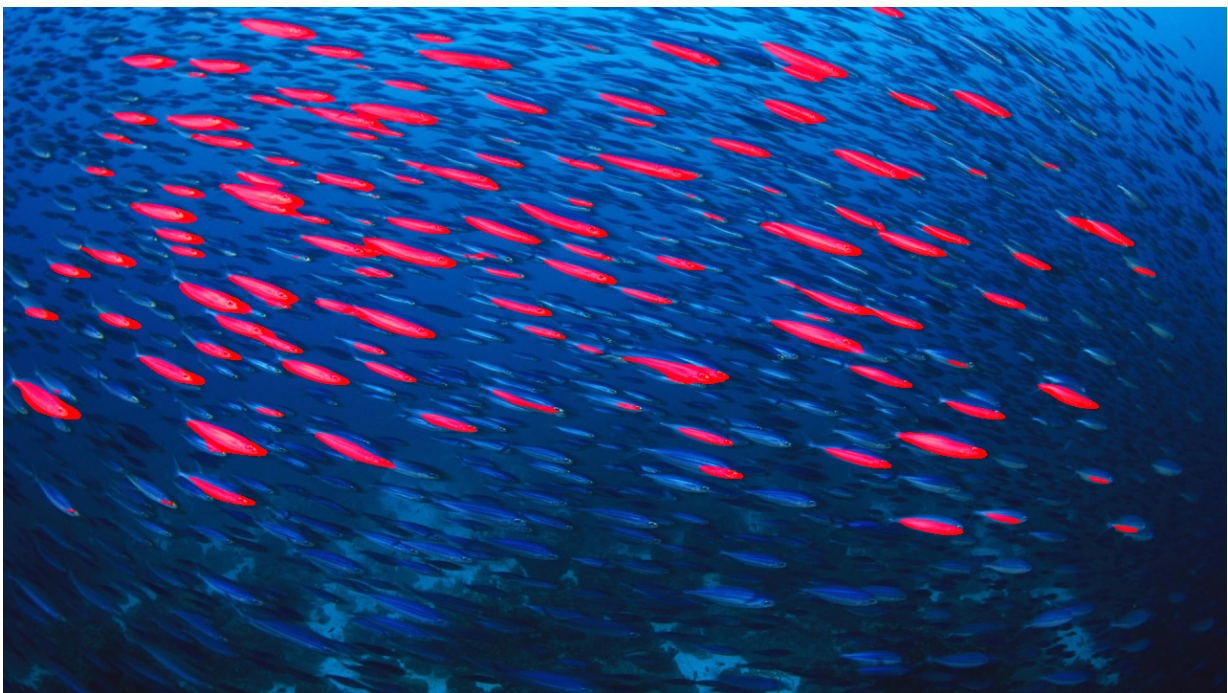
**Gallia est omnis divisa in partes tres, quarum unam incolunt Belgae, aliam Aquitani, tertiam qui ipsorum lingua Celtae, nostra Galli appellantur. Hi omnes lingua, institutis, legibus inter se differunt. Gallos ab Aquitanis Garumna flumen, a Belgis Matrona et Sequana dividit. Horum omnium fortissimi sunt Belgae, propterea quod a cultu atque humanitate provinciae longissime absunt, minimeque ad eos mercatores saepe commeant atque ea quae ad effeminandos animos pertinent important, proximique sunt Germanis**

Wzorzec: **e**

Gallia **e**st omnis divisa in partes **tres**, quarum unam incolunt **Belgae**, aliam Aquitani, **tertiam** qui ipsorum lingua **Celtae**, nostra Galli **appellantur**. Hi omnes lingua, institutis, **legibus** inter **se** differunt. Gallos ab Aquitanis Garumna flumen, a **Belgis** Matrona **et** **Sequana** dividit. Horum omnium fortissimi sunt **Belgae**, **propterea** quod a cultu atque **humanitate** provinciae **longissime** absunt, **minimeque** ad **eos** **mercatores** **saepe** **commeant** atque **ea** quae ad **effeminandos** animos **pertinent** important, **proximique** sunt **Germanis**

Rys. 1. galia.png

Wzorzec: 



Rys. 2. school.jpg




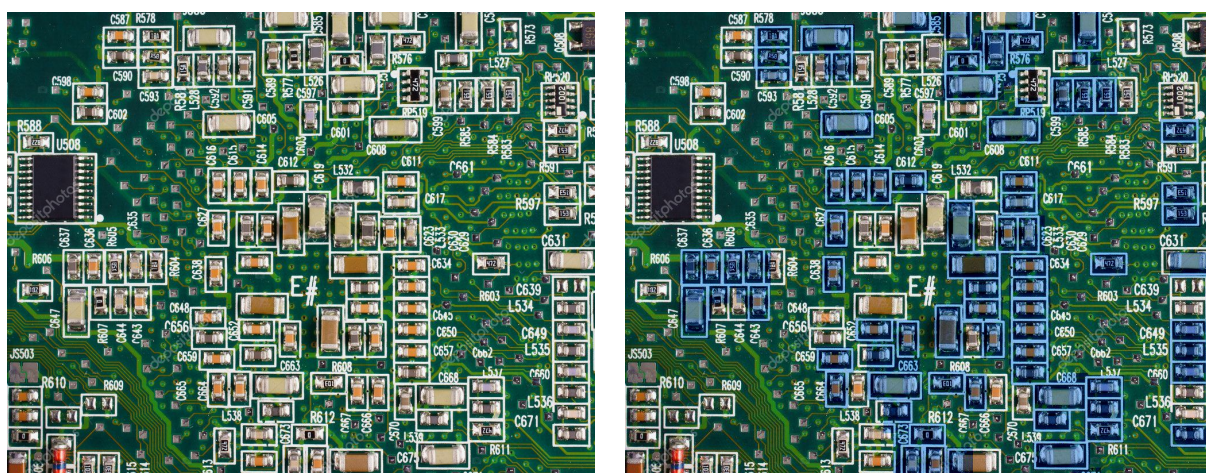
W przypadku tekstu na obrazie `galia.png` rozpoznanie litery *e* zostało zrealizowane metodą maksymalnej korelacji z parametrem  $\mu = 0.9$  dla bitmapy będącej wersją obrazu w skali szarości. Kolory zostały odwrócone z uwagi na fakt, że wysokie wartości korelacji powinny być osiągane dla pikseli, które stanowią fragment litery, a nie tła obrazu. Odnalezione oraz odpowiednio oznaczone zostały wszystkie wystąpienia tego wzorca. Dla mniejszych współczynników  $\mu$  algorytm rozpoznawał jako literę *e* również podobne znaki, np. *o*, *a*, *u* itd. Natomiast dla większej wartości korelacji nie wszystkie wystąpienia wzorca zostały prawidłowo odnalezione.

W przypadku ławicy ryb korelacja jest obliczana dla kanału czerwonego, ponieważ jasnoniebieskie tło stanowi znaczący problem dla wiarygodności wysokich wartości korelacji. Ta modyfikacja umożliwiła uzyskanie lepszych rezultatów niż w przypadku konwersji obrazu do skali szarości. Pomimo to nie wszystkie ryby na zdjęciu zostały odnalezione, dlatego że występują one w różnych rozmiarach oraz kątach względem brzegów obrazu, co znacznie obniża stopień ich podobieństwa.

Z uwagi na dosyć satysfakcjonujące rezultaty, postanowiłem sprawdzić działanie algorytmu dla kilku dodatkowych obrazów oraz wzorców. W przypadku zdjęcia obwodu elektronicznego konieczne było wyszukiwanie kilku różnych wzorców, uwzględniając również obroty o 90 stopni. Mimo to nie wszystkie rezystory widoczne na zdjęciu zostały prawidłowo rozpoznane. Jest to najprawdopodobniej spowodowane faktem, że na wybranym zdjęciu dookoła komponentów elektronicznych występują białe ramki, które znacząco wpływają na wartości korelacji. Możliwe, że w ogólności lepszym wyjściem byłoby dokonanie transformacji obrazów wejściowych oraz wzorców, która polegałaby na zmianie wartości każdego piksela zgodnie ze wzorem

$$p(x, y) := |p(x, y) - E[p]|, \text{ gdzie } E[p] = \frac{1}{wh} \sum_x \sum_y p(x, y). \quad (8)$$

Wzorce:  + wersje obrócone



Rys. 3. `circuit.jpg`

Algorytm działa najlepiej dla obrazu wejściowego będącego skanem zapisu nutowego jednego z utworów Chopina oraz wzorca będącego jedną nutą wyciętą z obrazu. Rozpoznane zostały wszystkie wystąpienia z uwagi na duże podobieństwo nut do siebie. W analogiczny sposób można rozpoznawać pozostałe znaki, np. klucze, znaki chromatyczne, itd. Program znajduje więc zastosowanie również w digitalizacji utworów wydanych przed pojawieniem się oprogramowania do tworzenia zapisów nutowych.

Wzorzec: ●



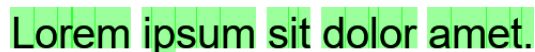
Rys. 4. chopin.jpg

## Zadanie 2. OCR

Rozpoznawanie tekstu zostało zrealizowane analogicznie do zadania 1. z laboratorium. Obrazy testowe są generowane automatycznie na podstawie wybranego kroju oraz rozmiaru czcionki przy użyciu programu `gen.py`. Aby rozpoznać tekst na obrazie, należy uruchomić program `ocr.py`.

Algorytm działa praktycznie bez zarzutu dla nieobróconych obrazów w formacie PNG. W tym przypadku problem stanowią jedynie kropka i przecinek, ponieważ są znacznie mniejsze niż pozostałe znaki, a przy tym składają się głównie z czarnych pikseli. Z tego powodu przy rozpoznawaniu znaków interpunkcyjnych próg współczynnika korelacji jest wyższy niż dla pozostałych liter oraz cyfr. Problematiczne są również znaki *I* oraz *l*, ze względu na to, że ich bitmapy występują również w bitmapach znaków *D*, *H*, *K*, *T*, *1* itd.

W przypadku dosyć mocno skompresowanych obrazów w formacie JPG, algorytm nie gwarantuje odczytania całości tekstu bezbłędnie, ale najczęściej udaje się rozpoznać znaczną większość znaków. Działanie programu zostało przetestowane na kilkudziesięciu planszach kontrolnych, a efekty testów zostały zaprezentowane poniżej.



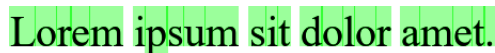
Rys. 5. Lorem ipsum, czcionka Arial, format PNG

```
Lorem ipsum sit dolor amet.
```

```
Finished in 1.71 s
```

```
Recognized 23 occurrences of 14 chars
```

```
. (1)  L (1)  a (1)  d (1)  e (2)  i (2)  l (1)  m (3)  o (3)  p (1)  
r (2)  s (2)  t (2)  u (1)
```



Rys. 6. Lorem ipsum, czcionka Times New Roman, format PNG

```
Lorem ipsum sit dolor amet.
```

```
Finished in 2.49 s
```

```
Recognized 23 occurrences of 14 chars
```

```
. (1)  L (1)  a (1)  d (1)  e (2)  i (2)  l (1)  m (3)  o (3)  p (1)  
r (2)  s (2)  t (2)  u (1)
```

ABCDEFGHIJKLMNOPQRSTUVWXYZ  
 abcdefghijklmnopqrstuvwxyz 0123456789 ?!.,  
 Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
 Etiam arcu felis, dictum id molestie nec, sagittis quis massa.

Rys. 7. Wszystkie rozpoznawane znaki, czcionka Arial, format PNG

```

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz 0123456789 ?!.,
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Etiam arcu felis, dictum id molestie nec, sagittis quis massa.

Finished in 7.10 s
Recognized 169 occurrences of 65 chars
! (1)  , (4)  . (3)  0 (1)  1 (1)  2 (1)  3 (1)  4 (1)  5 (1)  6 (1)
7 (1)  8 (1)  9 (1)  ? (1)  A (1)  B (1)  C (1)  D (1)  E (2)  F (1)
G (1)  H (1)  I (1)  J (1)  K (1)  L (2)  M (1)  N (1)  O (2)  P (1)
R (1)  S (1)  T (1)  U (1)  V (1)  W (1)  X (1)  Y (1)  Z (1)  a (8)
b (1)  c (7)  d (5)  e (10) f (2)  g (3)  h (1)  i (16) j (1)  k (1)
l (5)  m (8)  n (4)  o (6)  p (3)  q (2)  r (5)  s (12) t (11) u (6)
v (1)  w (1)  x (1)  y (1)  z (1)

```

ABCDEFGHIJKLMNOPQRSTUVWXYZ  
 abcdefghijklmnopqrstuvwxyz 0123456789 ?!.,  
 Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
 Etiam arcu felis, dictum id molestie nec, sagittis quis massa.

Rys. 8. Wszystkie rozpoznawane znaki, czcionka Times New Roman, format PNG

```

ABCDEFGHIJKi LMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz 0123456789 ?!.,
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Etiam arcu felis, dictum id molestie nec, sagittis quis massa.

Finished in 12.17 s
Recognized 170 occurrences of 66 chars
! (1)  , (4)  . (3)  0 (1)  1 (1)  2 (1)  3 (1)  4 (1)  5 (1)  6 (1)
7 (1)  8 (1)  9 (1)  ? (1)  A (1)  B (1)  C (1)  D (1)  E (2)  F (1)
G (1)  H (1)  I (1)  J (1)  K (1)  L (2)  M (1)  N (1)  O (1)  P (1)
Q (1)  R (1)  S (1)  T (1)  U (1)  V (1)  W (1)  X (1)  Y (1)  Z (1)
a (8)  b (1)  c (7)  d (5)  e (10) f (2)  g (3)  h (1)  i (17) j (1)
k (1)  l (5)  m (8)  n (4)  o (6)  p (3)  q (2)  r (5)  s (12) t (11)
u (6)  v (1)  w (1)  x (1)  y (1)  z (1)

```



lorem ipsum sit dolor amet.

Rys. 9. Lorem ipsum, czcionka Arial, format JPG

icrem ipsu.m sit cicicr a.met.

lorem ipsum sit dolor amet.

Rys. 10. Lorem ipsum, czcionka Times New Roman, format JPG

i  
a.bcciefc.hiii imncpc.rstu.vw.xv.z 01i234i56789 ?!.,  
i , , , , ,  
icr.e.m,, ip,s.,u.m,, cicicr s.it a.m,,e.t. cc,n,s.e.cte.tu.r a.ciip,is.cin,c. e.iit.  
i i  
etia.m a.rcu. feiis, ciictu.m ic.i mciestie nec, sa.c.ittis c.u.is ma.ssa..

abcdefghijklmnopqrstuvwxy 0123456789 ?!.,  
lorem ipsum dolor sit amet, consectetur adipiscing elit.  
etiam arcu felis, dictum id molestie nec, sagittis quis massa.

Rys. 11. Wszystkie rozpoznawane znaki, czcionka Arial, format JPG

lorcm.. ipsu.m.. sit d!olor a.m..ct.

abcdefghijklmnopqrstuvwxy 0123456789 ?!.,  
lorem ipsum dolor sit amet, consectetur adipiscing elit.  
etiam arcu felis, dictum id molestie nec, sagittis quis massa.

Rys. 12. Wszystkie rozpoznawane znaki, czcionka Times New Roman, format JPG

!.  
a.bcd!cfgh.ijklm..n.opqrstu.vwx.v.z 01.234.56789 ?!.,  
! !. . .  
lo.rcm.. ips,u.m.. d!o.lo.r sit a.m..ct, co.ns,c.ctctu.r ad!ip.isc.in.g clit.  
! !  
ctiam.. a.rcu. f.clis, d!ictu.m.. id!m..olcstic n.c.c, sagittis qu.is m. a.ssa..



Dla każdego znaku z zestawu rozpoznawanych znaków generowana jest bitmapa, a następnie obliczana jest jej korelacja z bitmapą obrazu wejściowego. Punkty, w których wartość tej korelacji jest bliska autokorelacji wzorca, są oznaczane jako prawdopodobne wystąpienia wzorca. Współczynnik  $\varepsilon$  występujący we wzorze (7) zależy od znaku (czy jest literą *I*, *l* lub znakiem interpunkcyjnym), kroju czcionki oraz formatu pliku wejściowego.

W każdym ze skupisk punktów z prawdopodobnym wystąpieniem wzorca znajdowany jest ten z maksymalnym współczynnikiem korelacji. Punkty te są następnie sortowane względem współrzędnej pionowej (z odpowiednim zaokrągleniem do pewnej wielokrotności liczby naturalnej), a następnie poziomej współrzędnej.

Powstały w ten sposób ciąg wystąpień wzorców jest konwertowany na tekst przy użyciu prostego algorytmu wstawiającego spacje, w przypadku gdy punkty są w większej odległości niż szerokość poprzedniego znaku. W podobny sposób tekst jest uzupełniany znakami nowej linii.

Z uwagi na wysokie wartości korelacji osiągnięte przez znaki interpunkcyjne, dopuszczalne jest wstawienie kropki albo przecinka tylko bezpośrednio po literze lub cyfrze, przy czym linia nie może rozpoczynać się od znaku interpunkcyjnego. Pozwala to ograniczyć liczbę błędnych identyfikacji znaku do minimum wynikającego z kompresji danych oraz niedoskonałości obrazu wejściowego.

Pewną niedoskonałością stosowanej metody rozpoznawania znaków jest wyszukiwanie wzorców na całym obrazie, a nie jego fragmentach. Zdecydowanie lepszym rozwiązaniem mogłoby okazać się podzielenie obrazu ze względu na słowa, a następnie słów na litery. Wówczas rozpoznany znakiem byłby ten, którego bitmapa osiągałaby największy współczynnik korelacji z fragmentem bitmapy obrazu wejściowego. Metoda ta nie została jednak wybrana z uwagi na większą złożoność obliczeniową, a także implementacyjną, w porównaniu tej z zadania 1, która sprawuje się doskonale w przypadku znaków z dużą liczbą elementów charakterystycznych, ale gorzej w przypadku znaków podobnych do siebie, np. *1*, *I*, *l* itd. W tym przypadku sugerowanym rozwiązaniem jest dopasowanie progu współczynnika korelacji dla każdego znaku osobno.

## Bibliografia

- [1] [https://www.icsr.agh.edu.pl/~mownit/pdf/16\\_fourier\\_v2.pdf](https://www.icsr.agh.edu.pl/~mownit/pdf/16_fourier_v2.pdf)
- [2] [https://brain.fuw.edu.pl/edu/index.php/Twierdzenia\\_o\\_splocie\\_i\\_o\\_próbkowaniu\\_\(aliasing\)](https://brain.fuw.edu.pl/edu/index.php/Twierdzenia_o_splocie_i_o_próbkowaniu_(aliasing))