Zadanie 2 z listy 4 - "Kompresja Danych"

Łukasz Klasiński

18 kwietnia 2020

Zadanie 2

Jednym z zastosowań kodowania arytmetycznego jest bezstratna kompresja obrazów z wieloma odcieniami szarości. Wykorzystuje się wówczas m.in. „kodowanie na poziomie bitów" polegające na tym, że najpierw koduje się ciąg złożony z „najstarszych" bitów każdego piksela, następnie ciąg złożony z drugich bitów kolejnych pikseli, itd. Metoda ta daje lepsze efekty, gdy kolejne odcienie szarości reprezentujemy nie przy pomocy kolejnych liczb naturalnych lecz za pomocą kodów Graya, które sąsiednim wartościom przyporządkowują ciągi bitów różniące się na dokładnie jednej pozycji.

```
1. Uzasadnij dlaczego kody Graya dają lepsze efekty od „standardowej" reprezentacji?
2. Wykaż, że kody Graya dwóch sąsiednich liczb różnią się na dokładnie jednej pozycji.
```

1. Uzasadnij dlaczego kody Grey'a dają lepze efekty od „standardowej" reprezentacji?

Metoda kompresji takich czarno-białych zdjęć, tak naprawdę polega na przekształceniu zdjęcia na n `subzdjęć` (gdzie n to ilość bitów w liczbie odcieni szarości), które mają 2 odcienie szarości (czyli wyłącznie piksele czarne lub białe). Zauważmy teraz, że na zdjęciach kolejne piksele średnio mają podobny odcień szarości (taki sam, bądź różny o kilka stopni). Niestety ze względu na to, że sąsiednie liczby binarne mogę się różnić znaczącą liczbą bitów, przy stworzeniu takich `subzdjęć` dane będą w pewnym stopniu niespójne. W`subzdjęciu` będą pojawiały się czarne plamy, gdzie w oryginalnym obrazie takowych nie ma. Przez to kompresja takich ciągów pikseli nie jest optymalna, ponieważ dane w pewnym stopniu przypominają ciągi losowe. Dzięki zastosowaniu kodowania gray'a do prezentowania odcieni pikseli, dane w poszczególnych `subzdjęciach` są bardziej spójne dzięki czemu algorytmy kompresji lepiej sobie z nimi radzą.

2. Wykaż, że kody Graya dwóch sąsiednich liczb różnią się na dokładnie jednej pozycji.

Kody Grey'a możemy łatwo wyliczyć korzystając z następującej rekursji:

$$g_0 = b_0$$

$$g_k = b_k \oplus (b_k/2)$$

gdzie g_k to k-ty symbol grey'a, a b_k to binarna reprezentacja k-tej liczby naturalnej, a $\oplus$ jest operacją XOR.

Do pokazania faktu, że kolejne liczby grey'a różnią się o 1, wystarczy pokazać że $g_k \oplus g_{k+1}$ da liczbę binarną z pojedyńczą jedynką. Podstawmy to zatem do wzoru na kolejne liczby grey'a:

$$g_k \oplus g_{k+1} = b_k \oplus (b_k/2) \oplus b_{k+1} \oplus (b_{k+1}/2)$$

Ponieważ operacja XOR jest łączna, to możemy zmienić kolejność wyrażenia na:

$$b_k \oplus b_{k+1} \oplus (b_k/2) \oplus (b_{k+1}/2)$$

Teraz można zauważyć, że $\oplus(b_k/2) \oplus (b_{k+1}/2)$ to jest XOR kolejnych liczb binarnych, z shiftem bitów w prawo o 1. Shift możemy wyciągnąć poza nawias, ponieważ dla XOR nie ma znaczenia w którym momencie robimy shift:

$$[b_k \oplus b_{k+1}] \oplus [(b_k \oplus b_{k+1}) >> 1]$$

Teraz wystarczy obserwacja, że operacja XOR na dwóch kolejnych liczbach binarnych zawsze zwraca liczbę binarną, która składa się z samych jedynek. W takim razie, otrzymujemy:

$$1^l \oplus 1^{l-1} = 10^{l-1}$$

czyli to co chcieliśmy pokazać.

$\square$