

Proszę wykonać oddzielnie podane niżej zadania dla obrazka „rzeczka.jpg”. Nie należy korzystać z gotowych funkcji dokonujących obróbki obrazka.



Teoria dot. filtrów z maską (inaczej nazywany filtrem tablicowym, macierzowym, konwolucyjnym) $N \times N$

Filtry z maską $N \times N$ (zwykle o rozmiarze 3×3 , 5×5 ...) służą do obróbki obrazu, czyli sygnału 2-wymiarowego. Każdy z kolorów jest obrabiany osobno. Parametrami tego filtru jest tablica, najczęściej kwadratowa, o nieparzystej szerokości. Tablica ta zawiera wartości rzeczywiste. Wartości wyjściowe obrazu liczone są na podstawie wartości wejściowych tego piksela oraz pikseli sąsiednich pomnożonych przez odpowiednie mnożniki brane z maski. W zależności od podanych mnożników można uzyskać bardzo wszechstronne efekty graficzne. Mnożniki te zapisywane są w postaci tablicy maski. Przykładem niech będzie wykorzystanie filtra o rozmiarze 5×5 do obróbki sygnału.

Wejście (sygnał wejściowy) :							tablica maski					wyjście																																																																																																																			
We= <table><tr><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td></tr><tr><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td></tr><tr><td>33</td><td>34</td><td>35</td><td>36</td><td>37</td><td>38</td><td>39</td></tr><tr><td>63</td><td>64</td><td>65</td><td>66</td><td>67</td><td>68</td><td>69</td></tr><tr><td>73</td><td>74</td><td>75</td><td>76</td><td>77</td><td>78</td><td>79</td></tr><tr><td>83</td><td>84</td><td>85</td><td>86</td><td>87</td><td>88</td><td>89</td></tr></table>							13	14	15	16	17	18	19	23	24	25	26	27	28	29	33	34	35	36	37	38	39	63	64	65	66	67	68	69	73	74	75	76	77	78	79	83	84	85	86	87	88	89	M <table><tr><td>0,4</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0,5</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>-1.1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0,6</td></tr></table>					0,4	0	0	0	0	0	0	0,5	0	0	0	0	-1.1	0	0	0	0	0	0	0	0	0	0	0	0,6	Wy <table><tr><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td></tr><tr><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td></tr><tr><td>...</td><td>...</td><td>+ 0,4*13 + 0,5*25 - 1,1*35 + 0,6*77</td><td>+ 0,4*14 + 0,5*26 - 1,1*36 + 0,6*78</td><td>+ 0,4*15 + 0,5*27 - 1,1*37 + 0,6*79</td><td>...</td><td>...</td></tr><tr><td>...</td><td>...</td><td>+ 0,4*23 + 0,5*35 - 1,1*65 + 0,6*87</td><td>+ 0,4*24 + 0,5*36 - 1,1*66 + 0,6*88</td><td>+ 0,4*25 + 0,5*37 - 1,1*67 + 0,6*89</td><td>...</td><td>...</td></tr><tr><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td></tr><tr><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td></tr></table>							+ 0,4*13 + 0,5*25 - 1,1*35 + 0,6*77	+ 0,4*14 + 0,5*26 - 1,1*36 + 0,6*78	+ 0,4*15 + 0,5*27 - 1,1*37 + 0,6*79	+ 0,4*23 + 0,5*35 - 1,1*65 + 0,6*87	+ 0,4*24 + 0,5*36 - 1,1*66 + 0,6*88	+ 0,4*25 + 0,5*37 - 1,1*67 + 0,6*89
							13	14	15	16	17	18	19																																																																																																																		
							23	24	25	26	27	28	29																																																																																																																		
							33	34	35	36	37	38	39																																																																																																																		
							63	64	65	66	67	68	69																																																																																																																		
							73	74	75	76	77	78	79																																																																																																																		
83	84	85	86	87	88	89																																																																																																																									
0,4	0	0	0	0																																																																																																																											
0	0	0,5	0	0																																																																																																																											
0	0	-1.1	0	0																																																																																																																											
0	0	0	0	0																																																																																																																											
0	0	0	0	0,6																																																																																																																											
...																																																																																																																									
...																																																																																																																									
...	...	+ 0,4*13 + 0,5*25 - 1,1*35 + 0,6*77	+ 0,4*14 + 0,5*26 - 1,1*36 + 0,6*78	+ 0,4*15 + 0,5*27 - 1,1*37 + 0,6*79																																																																																																																									
...	...	+ 0,4*23 + 0,5*35 - 1,1*65 + 0,6*87	+ 0,4*24 + 0,5*36 - 1,1*66 + 0,6*88	+ 0,4*25 + 0,5*37 - 1,1*67 + 0,6*89																																																																																																																									
...																																																																																																																									
...																																																																																																																									

Warto zwrócić uwagę, że filtry te nie zwracają normalnie wartości dla wartości wejściowych leżących na obrzeżach sygnału wejściowego.

Można popatrzeć co brane jest pod uwagę do wyliczania wartości piksela ($x=4$; $y=3$) :

13	14 *	15 *	16 *	17 *	18 *	19
	0,4	0	0	0	0	
23	24 *	25 *	26 *	27 *	28 *	29
	0	0	0,5	0	0	
33	34 *	35 *	36 *	37 *	38 *	39
	0	0	-1,1	0	0	

63	64 * 0	65 * 0	66 * 0	67 * 0	68 * 0	69
73	74 * 0	75 * 0	76 * 0	77 * 0	78 * 0,6	79
83	84	85	86	87	88	89

Brane są więc wartości pikseli sąsiednich pomnożone przez tablicę maski.

Normalizacja maski

Tablica z maską uważana jest za znormalizowaną, jeśli suma jej wartości wynosi 1, wtedy również obrazek w wyniku działania filtra będzie mniej więcej tak samo jasny po obróbce. Normalizacja maski nie jest konieczna. Oto przykład działania:

Tablica maski przed normalizacją	Tablica z maską znormalizowaną																											
<table><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>2</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr></table>	0	1	0	1	2	1	0	1	0	<p>W przypadku, kiedy wartości w masce są nieujemne, wtedy w celu normalizacji wystarczy podzielić wartości w masce przez sumę starych wartości.</p> <table><tr><td>0</td><td>1/6</td><td>0</td></tr><tr><td>1/6</td><td>2/6</td><td>1/6</td></tr><tr><td>0</td><td>1/6</td><td>0</td></tr></table>	0	1/6	0	1/6	2/6	1/6	0	1/6	0									
0	1	0																										
1	2	1																										
0	1	0																										
0	1/6	0																										
1/6	2/6	1/6																										
0	1/6	0																										
Tablica maski przed normalizacją	Tablica z maską znormalizowaną																											
<table><tr><td>0</td><td>-1</td><td>0</td></tr><tr><td>-1</td><td>2</td><td>-1</td></tr><tr><td>0</td><td>-1</td><td>0</td></tr></table>	0	-1	0	-1	2	-1	0	-1	0	<p>W przypadku istnienia w masce wartości ujemnych nie ma jedynej sugerowanej metody służącej do normalizacji. Można tego dokonać na wiele różnych sposobów, na przykład:</p> <table><tr><td>0</td><td>-1</td><td>0</td></tr><tr><td>-1</td><td>5</td><td>-1</td></tr><tr><td>0</td><td>-1</td><td>0</td></tr></table> <p>albo</p> <table><tr><td>0</td><td>-0,25</td><td>0</td></tr><tr><td>-0,25</td><td>2</td><td>-0,25</td></tr><tr><td>0</td><td>-0,25</td><td>0</td></tr></table>	0	-1	0	-1	5	-1	0	-1	0	0	-0,25	0	-0,25	2	-0,25	0	-0,25	0
0	-1	0																										
-1	2	-1																										
0	-1	0																										
0	-1	0																										
-1	5	-1																										
0	-1	0																										
0	-0,25	0																										
-0,25	2	-0,25																										
0	-0,25	0																										

Dodawanie marginesu

Niestety, filtry z maską nie potrafią wprost wyznaczyć wartości wyjściowej dla brzegów. Aby temu zapobiec można dodać margines o szerokości uzależnionej od rozmiaru filtra. Dla filtra 3x3 będzie to margines o rozmiarze 1, dla filtra 5x5 margines 2. Wartości na marginesie można ustalić na różny sposób, najbardziej odpowiednim wydaje się skopiowanie wartości z najbliższego, oryginalnego piksela. Przykład działania:

Wejście (sygnał wejściowy) :							Wyjście, po dodaniu marginesu o rozmiarze 2 (dla filtra 5x5)										
13	14	15	16	17	18	19	13	13	13	14	15	16	17	18	19	19	19
23	24	25	26	27	28	29	13	13	13	14	15	16	17	18	19	19	19
33	34	35	36	37	38	39	13	13	13	14	15	16	17	18	19	19	19
63	64	65	66	67	68	69	23	23	23	24	25	26	27	28	29	29	29

73	74	75	76	77	78	79	33	33	33	34	35	36	37	38	39	39	39
83	84	85	86	87	88	89	63	63	63	64	65	66	67	68	69	69	69
							73	73	73	74	75	76	77	78	79	79	79
							83	83	83	84	85	86	87	88	89	89	89
							83	83	83	84	85	86	87	88	89	89	89
							83	83	83	84	85	86	87	88	89	89	89

Zastosowanie filtra z maską i marginesem

W celu zastosowania filtra maskowego również wobec wartości leżących na obrzeżach można tymczasowo dodać możliwie najmniejszy margines, wykonać filtrowanie i usunąć dodany margines. Oto przykład.

Wejście	Maska	Dodany margines o rozmiarze 1 (ponieważ filtr jest wielkości 3x3)	Zastosowanie filtra
<div> <div>232425</div> <div>333435</div> <div>636465</div> <div>737475</div> </div>	<div> <div>00,20</div> <div>0,30,40</div> <div>000,1</div> </div>	<div> <div>2323242525</div> <div>2323242525</div> <div>3333343535</div> <div>6363646565</div> <div>7373747575</div> <div>7373747575</div> </div>	<div> <div>n/dn/dn/dn/dn/d</div> <div> n/d + 0,2*23 + 0,3*23 + 0,4*23 + 0,1*34 + 0,2*24 + 0,3*23 + 0,4*24 + 0,1*35 + 0,2*25 + 0,3*24 + 0,4*25 + 0,1*35 n/d </div> <div> n/d + 0,2*23 + 0,3*33 + 0,4*33 + 0,1*64 + 0,2*24 + 0,3*33 + 0,4*34 + 0,1*65 + 0,2*25 + 0,3*34 + 0,4*35 + 0,1*65 n/d </div> <div> n/d + 0,2*33 + 0,3*63 + 0,4*63 + 0,1*74 + 0,2*34 + 0,3*63 + 0,4*64 + 0,1*75 + 0,2*35 + 0,3*64 + 0,4*65 + 0,1*75 n/d </div> <div> n/d + 0,2*63 + 0,3*73 + 0,4*73 + 0,1*74 + 0,2*64 + 0,3*73 + 0,4*74 + 0,1*75 + 0,2*65 + 0,3*74 + 0,4*75 + 0,1*75 n/d </div> <div>n/dn/dn/dn/dn/d</div> </div>

Zastosowanie filtra z maską dla obrazów kolorowych

Filtry z maską NxN stosuje się wobec sygnałów 2-wymiarowych. W przypadku obróbki obrazów kolorowych (czyli tablic 3-wymiarowych) należy przetwarzać osobno wartości dotyczące poszczególnych kolorów.

Wskazówka odnośnie wydajności

Skrypt realizujący powyższe zadania można napisać w wieloraki sposób. Osoby, które napiszą część odpowiedzialną za użycie maski używając **4 pętli for** powinni zamiast obróbki obrazka w oryginalnej wielkości zdecydować się na obróbkę obrazka pomniejszonego, gdyż proces ten będzie zajmował dużo czasu. Prędkość wykonywania programu może być w niezbyt trudny sposób przyspieszona przez wykonywanie tylko **2 pętli for**. Aby tego dokonać należy najpierw przerobić program tak, by w pętlach zewnętrznych przechodzić po kolejnych wartościach w masce, a w pętlach wewnętrznych po pikselach w obrazku, kolejnym krokiem jest usunięcie z użycia pętli wewnętrznych.

Zadanie 1.

Proszę napisać skrypt, który wykona operacje przedstawioną w powyższym przykładzie (podrozdział „Zastosowanie filtra z maską i marginesem”), przy czym skrypt powinien umożliwiać pracę z dowolną maską 3x3 i dowolnym sygnałem 2-wymiarowym.

Zadanie 2.

Proszę napisać skrypt, który będzie dokonywał rozmywania obrazu kolorowego za pomocą filtra widocznego poniżej. Przed zastosowanie tej maski proszę ją znormalizować. Proszę napisać funkcję, która będzie umożliwiała pracę w obrazkiem monochromatycznym (dane w postaci tablicy 2-wymiarowej) i kolorowym (dane w postaci tablicy o wymiarze trzecim równym 3).

5	6	5
6	6	6
5	6	5



Zadanie 3.

Proszę napisać skrypt, który będzie dokonywał wyostrażania obrazu kolorowego za pomocą filtra widocznego poniżej. Przed zastosowanie tej maski nie trzeba jej normalizować, a po zastosowaniu tego filtra wartości trzeba przyciąć do przedziału [0; 255] (patrz funkcja *PrzytnijZakres*)

0	-1	0
-1	5	-1
0	-1	0

org



zmieniony



Zadanie 4.

Proszę napisać skrypt, który zmieni obrazek na odcienie szarości (za pomocą funkcji średnia), aby później dokonać wykrywania krawędzi w poziomie za pomocą filtra widocznego poniżej, a na końcu wyliczy moduł z wartości.

0	0	0
-1	0	1
0	0	0



Zadanie 5.

(Filtr 3x3 **Sobel**) Proszę napisać skrypt, który nie zmieni obrazek na odcienie szarości po czym dokona wykrywania krawędzi w pionie za pomocą filtra widocznego poniżej, a na końcu wyliczy moduł z wartości otrzymanych po obróbce tym filtrem.

-1	-2	-1
0	0	0
1	2	1



Zadanie 6.

Proszę napisać skrypt, który zmieni obrazek na odcienie szarości, później zastosuje złożenie przez funkcję maksimum obrazów będących wynikiem działania wartości bezwzględnej z filtrów wykrywających krawędzie w poziomie i pionie. Schemat działania widoczny jest niżej.

- Niech *obr1* to obrazek oryginalny przetworzony przez filtr wykrywający krawędzie w pionie (widoczny poniżej), następnie należy wyliczyć wartość bezwzględną.

0	-1	0
0	0	0
0	1	0

- Niech *obr2* to obrazek oryginalny przetworzony przez filtr wykrywający krawędzie w poziomie (widoczny poniżej), następnie trzeba obliczyć moduł.

0	0	0
1	0	-1
0	0	0

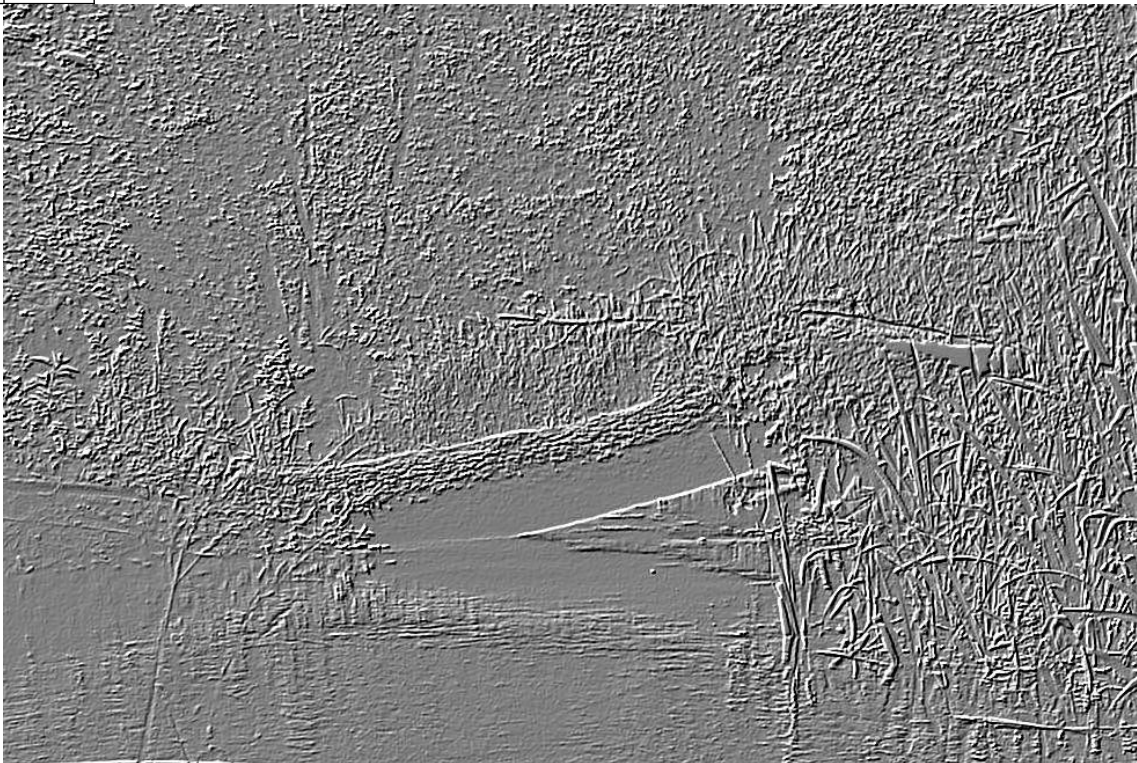
- Niech obrazek wynikowy to złożenie *obr1* i *obr2* przez funkcję maksimum.



Zadanie 7.

(Efekt **Emboss**) Proszę napisać skrypt, który zmieni obrazek na odcienie szarości, następnie dokona wykrywania krawędzi w skosie (zgodnie z filtrem widocznym poniżej), na końcu wartości zostaną zwiększone o 127.5 (wartość idealnie szara).

-1	-1	0
-1	0	1
0	1	1



Zadanie 8.

(operator **Frei'a-Chen'a**) Proszę napisać skrypt, który zmieni obrazek na odcienie szarości za pomocą średniej, później zastosuje złożenie przez funkcję $wy = \sqrt{wy_filtra_1^2 + wy_filtra_2^2}$ obrazów powstałych przez dwa filtry wykrywających krawędzie w poziomie i pionie. Schemat działania widoczny jest niżej.

- Niech *obr1* to kopia obrazka oryginalnego, zamieniona na odcienie szarości i przetworzona przez filtr wykrywający krawędzie w pionie (widoczny poniżej), bez przycięcia wartości do [0; 255]

-1	$-\sqrt{2}$	-1
0	0	0
1	$+\sqrt{2}$	1

- Niech *obr2* to kopia obrazka oryginalnego, zamieniona na odcienie szarości i przetworzona przez filtr wykrywający krawędzie w poziomie (widoczny poniżej), bez przycięcia wartości do [0; 255]

-1	0	1
$-\sqrt{2}$	0	$+\sqrt{2}$
-1	0	1

- Niech obrazek wyjściowy *img3* powstanie przez złożenie *obr1* i *obr2* w taki sposób, aby każda wartość w obrazku wyjściowym była liczona jako $img3 = \sqrt{img1^2 + img2^2}$

