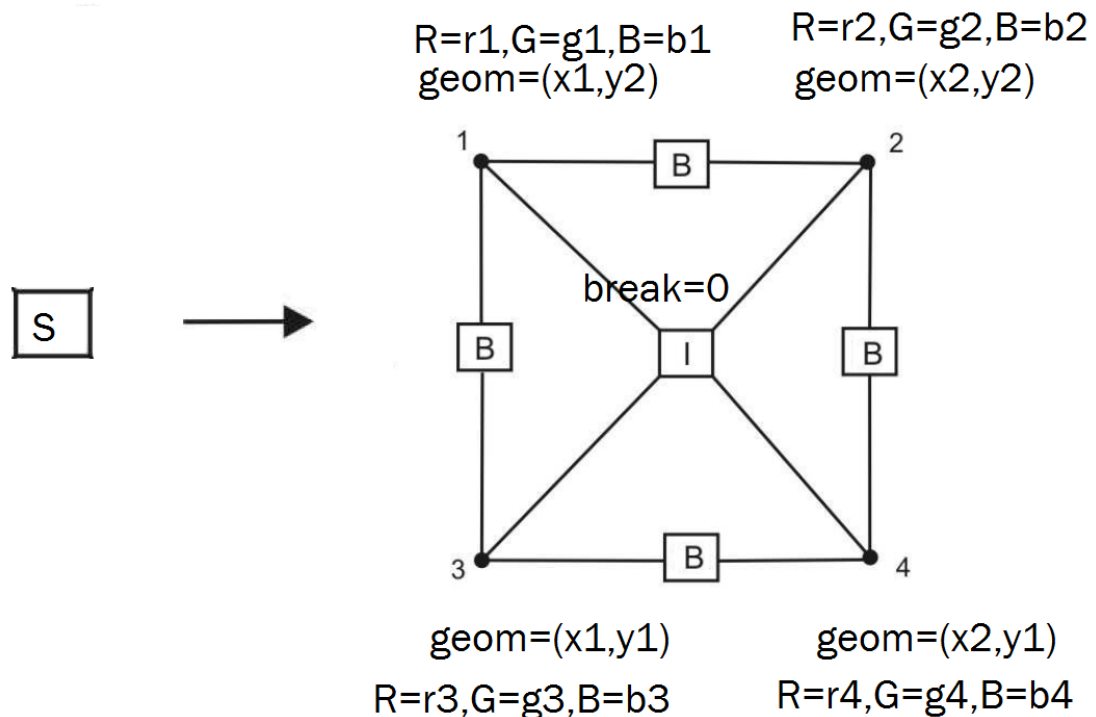


Dekompozycja algorytmu kompresji bitmap na produkcję hipergramatyki grafowej i taski (Maciej Paszynski (12.2018))

Production P1



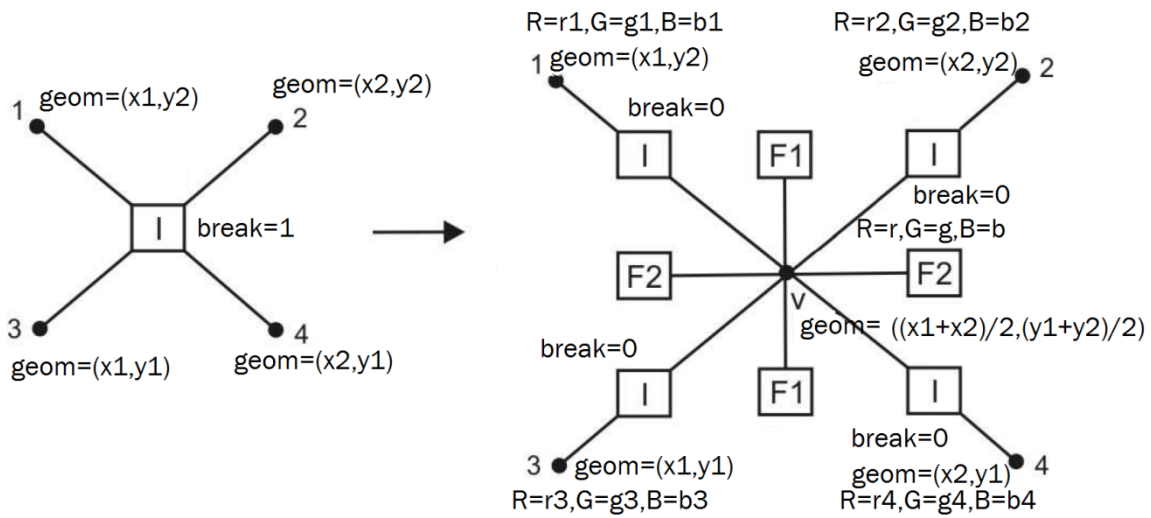
1. Proszę zaimplementować produkcję P1. Implementacja powinna zawierać tworzenie odpowiedniego hipergrafu w strukturze danych. Proszę żeby wierzchołki grafu 1,2,3,4 zawierały atrybuty: $geom=(x,y)$ współrzędne geometryczne wierzchołka (ta będą współrzędne skrajnych pixeli bitmapy, typu integer), hiperkrawędź i żeby zawierała atrybut $break=0$, oraz wierzchołki hipergrafu 1,2,3,4 zawierały atrybut R,G,B (współczynniki z zakresu 0 do 255 które będą używane do przybliżania bitmapy). Wartości atrybutów RGB proszę pobrać z pikseli bitmapy które znajdują się w wierzchołkach

2. Proszę napisać test jednostkowy do produkcji P1 dla jakiejś bitmapy

Proszę zwizualizować produkcję (wynikowy hipergraf wraz z atrybutami).

Proszę sprawdzić strukturę uzyskanego hipergrafu, proszę sprawdzić atrybuty RGB z bitmapą, proszę sprawdzić wartości $geom$ z rozmiarem bitmapy

Produkcja P2

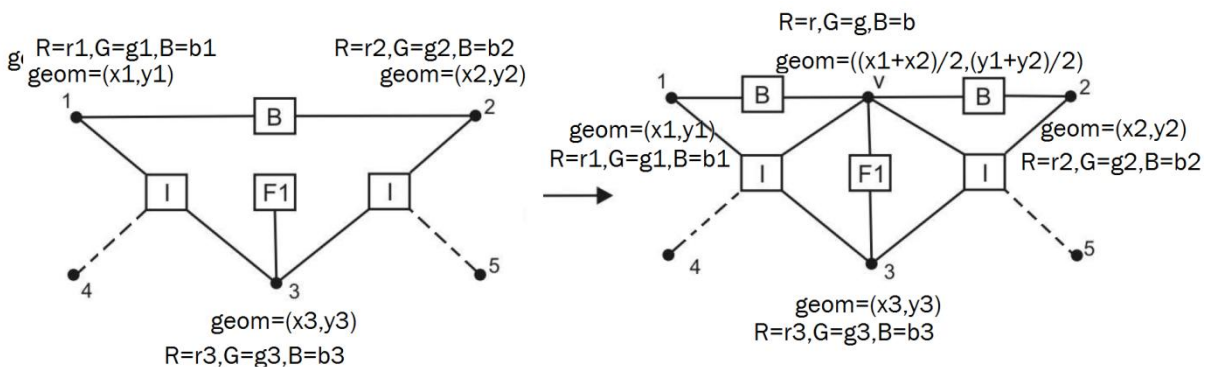


3. Proszę zaimplementować produkcję P2. Proszę żeby dodany nowy wierzchołek miał ustawione stosownie współrzędne geometryczne w atrybucie geom oraz żeby jego współczynniki przybliżania bitmapy RGB były ustawione na wartość piksela w tym punkcie. Proszę nie zmieniać atrybutów innych wierzchołków

4. Proszę napisać test jednostkowy do produkcji P2

Proszę zwizualizować produkcję (wynikowy hipergraf wraz z atrybutami). Proszę sprawdzić strukturę uzyskanego hipergrafu, proszę sprawdzić atrybuty wierzchołków RGB z pikselami z bitmapy, proszę sprawdzić wartości geom z rozmiarem bitmapy,

Produkcja P3

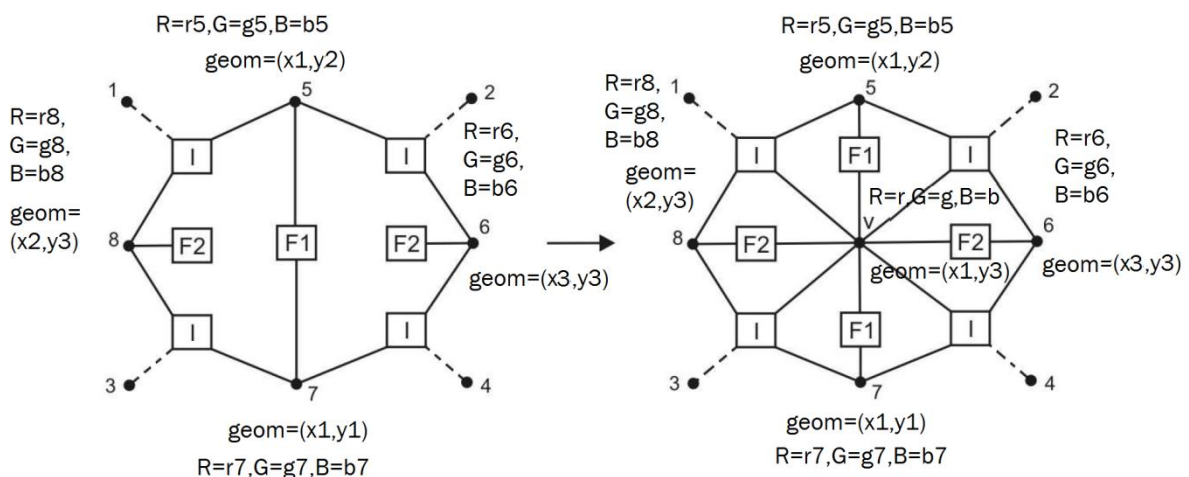


5. Proszę zaimplementować produkcję P3. Proszę żeby dodany nowy wierzchołek miał ustawione stosownie współrzędne geometryczne w atrybucie geom oraz żeby jego współczynniki przybliżania bitmapy RGB były ustawione na wartości z odpowiedniego pixela. Proszę nie zmieniać atrybutów innych wierzchołków

6. Proszę napisać test jednostkowy do produkcji P3

Proszę zwizualizować produkcję (wynikowy hipergraf wraz z atrybutami). Proszę sprawdzić strukturę uzyskanego hipergrafu, proszę sprawdzić atrybuty wierzchołków RGB z pikselami z bitmapy, proszę sprawdzić wartości geom z rozmiarem bitmapy,

Produkcja P4



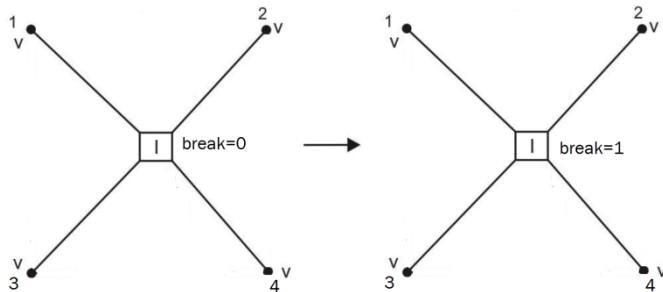
7. Proszę zaimplementować produkcję P4. Proszę żeby dodany nowy wierzchołek miał ustawione stosownie współrzędne geometryczne w atrybucie geom oraz żeby jego współczynniki przybliżania bitmapy RGB były ustawione na wartości pobrane z odpowiedniego pixela. Proszę nie zmieniać atrybutów innych wierzchołków.

8. Proszę napisać test jednostkowy do produkcji P4

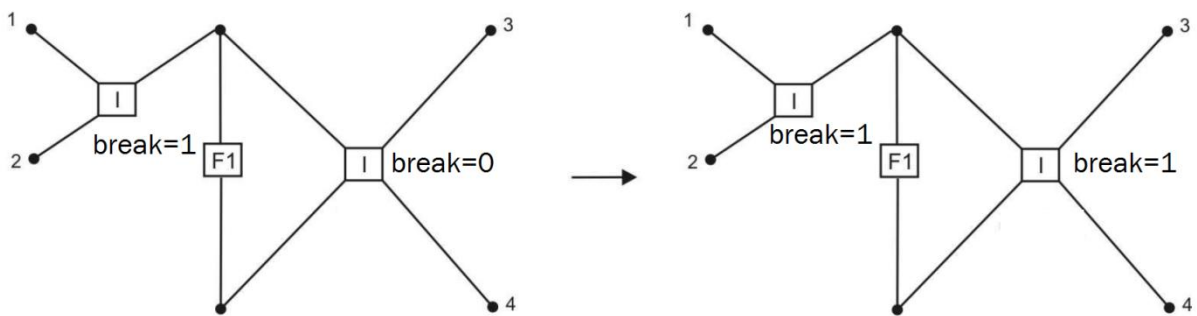
Proszę zwizualizować produkcję (wynikowy hipergraf wraz z atrybutami). Proszę sprawdzić strukturę uzyskanego hipergrafu, proszę sprawdzić atrybuty

wierzchołków RGB z pikselami z bitmapy, proszę sprawdzić wartości geom z rozmiarem bitmapy,

Produkcja P5



Produkcja P6



9. Proszę zaimplementować produkcję P5 która zmienia atrybut break z 0 na 1 dla hiperkrawędzi (wyrażając naszą intencje złamania krawędzi)

Proszę zaimplementować produkcje P6 która propaguje zamiar złamania małego elementu na dużego sąsiada, poprzez stosowne ustawienie flagi break

Proszę żeby wykonanie produkcji P5 uruchamiało algorytm który sprawdza sąsiednie większe krawędzie i ustawia im flage break (wykonuje produkcje P6 na wszystkie możliwe sposoby, rekurencyjnie wiele razy)

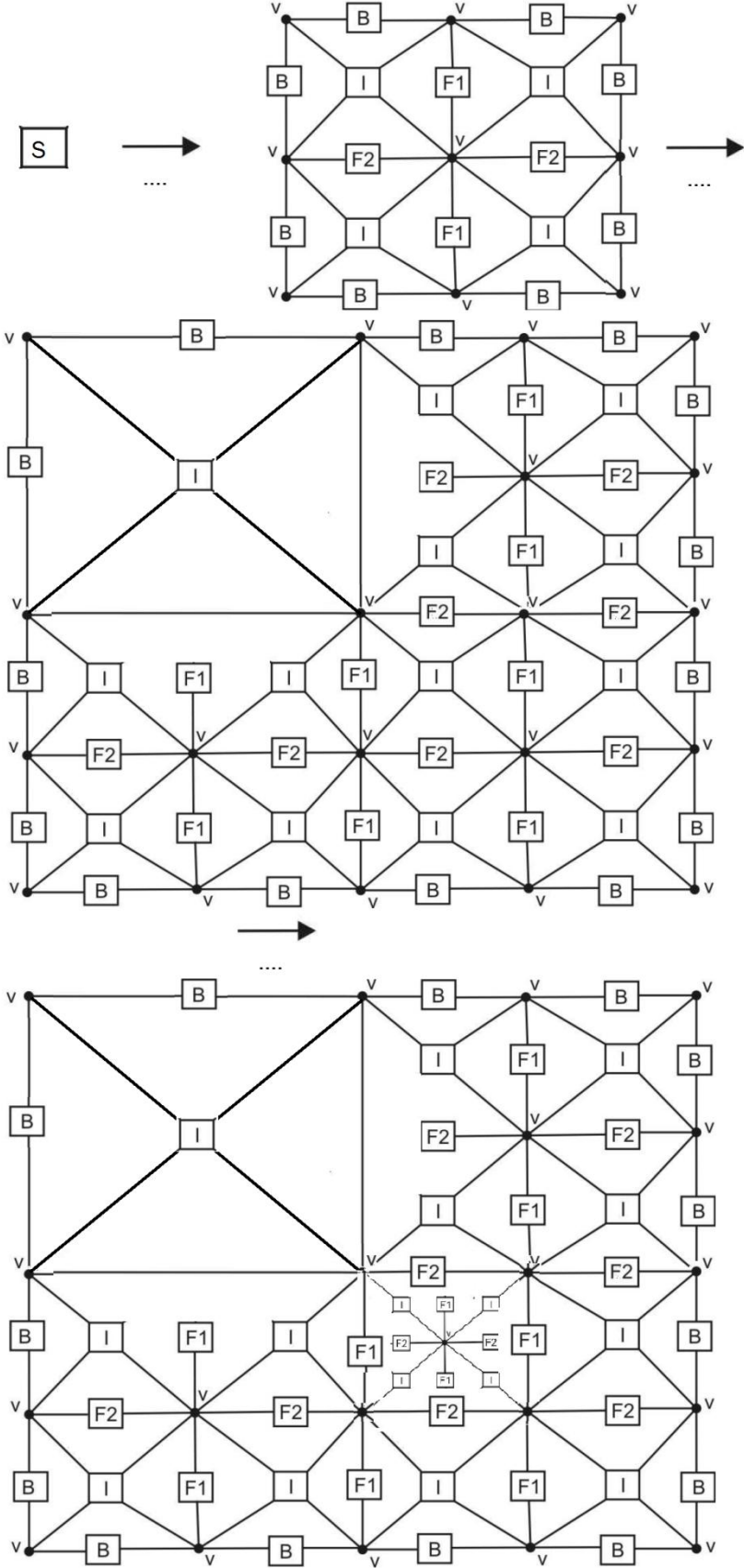
10. Proszę napisać następujący test generacji i propagacji intencji złamania krawędzi. Należy uruchomić produkcje

P1,P5 (próba P6) P2,P3,P3,P3,P3

P5 (próba P6),P5 (próba P6),P5 (próba P6),P2,P2,P2,P3,P3,P3,P3,P3,P3,P4,P4

P5 (próba P6), P2 oraz na końcu P5 na najmniejszym elemencie i propagacja chęci złamań P6.

Proszę sprawdzić czy duży element ma teraz ustawioną flagę break=1



11 . Proszę napisać sterownik który czyta plik wejściowy w podanym niżej formacie i będzie wykonywał odczytane produkcje. Proszę po wykonaniu produkcji P5 sprawdzać sąsiadów czy da się wykonać produkcję P6

Kod produkcji (1,2,3,4,5)

Dla produkcji P1 atrybuty

x_1, y_1, x_2, y_2

R1,G1,B1,R2,G2,B2,R3,G3,B3,R4.B4,G4

Dla produkcji P2

atrybuty

x_1, y_1, x_2, y_2 (po nich znajdujemy miejsce zastosowanie $(x_1+x_2)/2, (y_1+y_2)/2$)

R,G,B

Dla produkcji P3

atrybuty

x_1, x_2, y_1 (po nich znajdujemy miejsce zastosowanie $(x_1+x_2)/2, y_1$)

R,G,B

Dla produkcji P4

atrybuty

x_1, y_1, x_2, y_2 (po nich znajdujemy miejsce zastosowanie $(x_1+x_2)/2, y_1$)

R,G,B

Dla produkcji P5

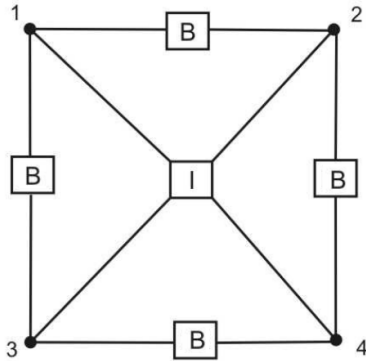
atrybuty

x_1, y_1 (po nich znajdujemy miejsce zastosowanie x_1, y_1)

12. Proszę napisać procedurę rysowania przybliżonej bitmapy w oparciu o cztery współczynniki $r_1, r_2, r_3, r_4, g_1, g_2, g_3, g_4, b_1, b_2, b_3, b_4$ rozpięte na kwadratowym hipergrafie o współrzędnych $(x_1, y_1) (x_2, y_2)$

Procedura wypełnia tablice APPROX_R, APPROX_G, APPROX_B przechowywane w pamięci globalnej

R=r1,G=g1,B=b1 R=r2,G=g2,B=b2
geom=(x1,y2) geom=(x2,y2)



geom=(x1,y1) geom=(x2,y1)
R=r3,G=g3,B=b3 R=r4,G=g4,B=b4

WE: (x1,y1),(x2,y2) r1,r2,r4,r4, g1,g2,g3,g4, b1,b2,b3,b4

APPROX_R[*][*]=0

APPROX_G[*][*]=0

APPROX_B[*][*]=0

Pętla px=x1,x2

Pętla py=y1,y2

$$\text{APPROX_R}[px][py] += r1 * \left(1 - \frac{px-x1}{x2-x1}\right) \frac{py-y1}{y2-y1}$$

$$\text{APPROX_G}[px][py] += g1 * \left(1 - \frac{px-x1}{x2-x1}\right) \frac{py-y1}{y2-y1}$$

$$\text{APPROX_B}[px][py] += b1 * \left(1 - \frac{px-x1}{x2-x1}\right) \frac{py-y1}{y2-y1}$$

$$\text{APPROX_R}[px][py] += r2 * \left(\frac{px-x1}{x2-x1}\right) \frac{py-y1}{y2-y1}$$

$$\text{APPROX_G}[px][py] += g2 * \left(\frac{px-x1}{x2-x1}\right) \frac{py-y1}{y2-y1}$$

$$\text{APPROX_B}[px][py] += b2 * \left(\frac{px-x1}{x2-x1}\right) \frac{py-y1}{y2-y1}$$

$$\text{APPROX_R}[px][py] += r3 * \left(1 - \frac{px-x1}{x2-x1}\right) \left(1 - \frac{py-y1}{y2-y1}\right)$$

$$\text{APPROX_G}[px][py] += g3 * \left(1 - \frac{px-x1}{x2-x1}\right) \left(1 - \frac{py-y1}{y2-y1}\right)$$

$$\text{APPROX_B}[px][py] += b3 * \left(1 - \frac{px-x1}{x2-x1}\right) \left(1 - \frac{py-y1}{y2-y1}\right)$$

$$\text{APPROX_R}[px][py] += r4 * \frac{px-x1}{x2-x1} \left(1 - \frac{py-y1}{y2-y1}\right)$$

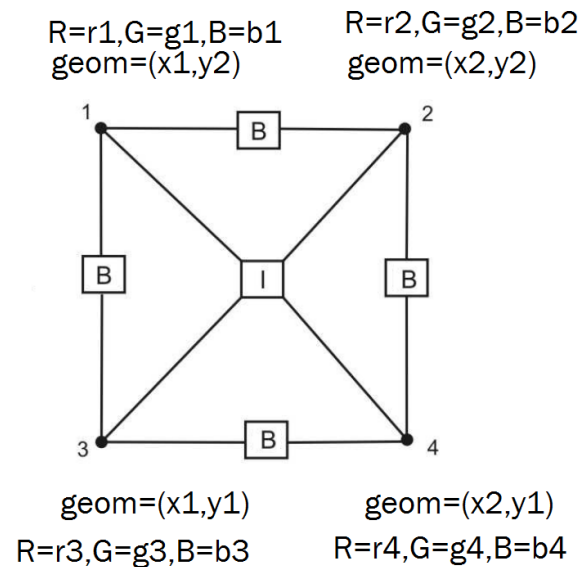
$$\text{APPROX_G}[px][py] += g4 * \frac{px-x1}{x2-x1} \left(1 - \frac{py-y1}{y2-y1}\right)$$

$$\text{APPROX_B}[px][py] += b4 * \frac{px-x1}{x2-x1} \left(1 - \frac{py-y1}{y2-y1}\right)$$

Rysowanie bitmapy [x1:x2][y1:y2] w oparciu o dane kolorów RGB w tablicach APPROX_R, APPROX_G, APPROX_B

13. Proszę napisać test jednostkowy rysowania bitmapy z punktu 12 przygotowując hipotetyczne dane. Na przykład jak uzyskać całą czerwoną, całą zieloną, całą niebieską bitmapę? Jak uzyskać całą białą lub całą czarną bitmapę?

14. Obliczanie błędu aproksymacji na elemencie hipergrafu



Pętla px=x1,x2

Pętla py=y1,y2

DIFF_R[px][py]=BITMAP_R[px][py];

DIFF_G[px][py]=BITMAP_G[px][py];

DIFF_B[px][py]=BITMAP_B[px][py];

Pętla px=x1,x2

Pętla py=y1,y2

$$\text{DIFF_R}[px][py]=r1*\left(1-\frac{px-x1}{x2-x1}\right)\frac{py-y1}{y2-y1}$$

$$\text{DIFF_G}[px][py]=g1*\left(1-\frac{px-x1}{x2-x1}\right)\frac{py-y1}{y2-y1}$$

$$\text{DIFF_B}[px][py]=b1*\left(1-\frac{px-x1}{x2-x1}\right)\frac{py-y1}{y2-y1}$$

$$\text{DIFF_R}[px][py]=r2*\left(\frac{px-x1}{x2-x1}\right)\frac{py-y1}{y2-y1}$$

$$\text{DIFF_G}[px][py]=g2*\left(\frac{px-x1}{x2-x1}\right)\frac{py-y1}{y2-y1}$$

$$\text{DIFF_B}[px][py]=b2*\left(\frac{px-x1}{x2-x1}\right)\frac{py-y1}{y2-y1}$$

$$\text{DIFF_R}[px][py]=r3*\left(1-\frac{px-x1}{x2-x1}\right)\left(1-\frac{py-y1}{y2-y1}\right)$$

$$\text{DIFF_G}[px][py]=g3*\left(1-\frac{px-x1}{x2-x1}\right)\left(1-\frac{py-y1}{y2-y1}\right)$$

$$\text{DIFF_B}[px][py]=b3*\left(1-\frac{px-x1}{x2-x1}\right)\left(1-\frac{py-y1}{y2-y1}\right)$$

$$\text{DIFF_R}[px][py]=r4*\frac{px-x1}{x2-x1}\left(1-\frac{py-y1}{y2-y1}\right)$$

$$\text{DIFF_G}[px][py]=g4*\frac{px-x1}{x2-x1}\left(1-\frac{py-y1}{y2-y1}\right)$$

$$\text{DIFF_B}[px][py]=b4*\frac{px-x1}{x2-x1}\left(1-\frac{py-y1}{y2-y1}\right)$$

ERROR=0

Pętla px=x1,x2

Pętla py=y1,y2

ERROR +=

$$0.5*(\text{DIFF_R}[px][py])^2+0.3*(\text{DIFF_G}[px][py])^2+0.2*(\text{DIFF_B}[px][py])^2$$

15. Proszę napisać test jednostkowy obliczania błędu aproksymacji bitmapy z punktu 14 przygotowując hipotetyczne dane. Na przykład jak uzyskać całą

czerwoną, całą zieloną, całą niebieską bitmapę? Jak uzyskać całą białą lub całą czarną bitmapę?

16. Algorytm oznaczania hiperkrawędzi do adaptacji

Założenie: Przechowujemy listę referencji do hiperkrawędzi I

WE: Hipergraf, epsilon (dopuszczalny błąd przybliżenia bitmapy), maxstep (maksymalna liczba iteracji)

Pętla step=1,maxstep

 Pętla i po hiperkrawędziach I

 Oblicz błąd przybliżenia dla hiperkrawędzi i

 Jeśli ten błąd jest większy niż epsilon, to

 - uruchom na tej hiperkrawędzi produkcję P5
 (która ustawia break na 1)

 - uruchom algorytm poprawiania złamań za pomocą produkcji P6

 Jeśli na żadnej hiperkrawędzi błąd nie był większy niż epsilon, to stop

 Powtarzaj

 Pętla i po hiperkrawędziach I

 Uruchom produkcję P2

 (jeśli hiperkrawędź I ma break=1 i cztery nóżki
 to produkcja P2 ją złamie)

 Pętla i po hiperkrawędziach B

 Uruchom produkcję P3

 (jeśli krawędź B sąsiaduje ze złamanym wnętrzem to
 produkcja P3 ją złamie)

 Pętla i po hiperkrawędziach F

 Uruchom produkcję P4

(jeśli krawędź F sąsiaduje ze dwoma złamanymi wnętrzami
to produkcja P4 ją złamie)

dopóki *którakolwiek z produkcji P2, P3, P4 została wykonana wewnątrz*

Koniec pętli step

17. Proszę napisać test jednostkowy obliczania błędu aproksymacji bitmapy z punktu 16 przygotowując hipotetyczne dane. Na przykład jak uzyskać całą czerwoną, całą zieloną, całą niebieską bitmapę? Jak uzyskać całą białą lub całą czarną bitmapę?

18. Proszę napisać algorytm rysowania przybliżonej bitmapy

APPROX_R[1:maxx][1:maxy]=0

APPROX_G[1:maxx][1:maxy]=0

APPROX_B[1:maxx][1:maxy]=0

Pętla i po hiperkrawędziach I

Odczytaj (x1,y1) (x2,y2) z tej hiperkrawędzi

Pętla px=x1,x2

Pętla py=y1,y2

$$\text{APPROX_R}[px][py] += r1 * \left(1 - \frac{px-x1}{x2-x1}\right) \frac{py-y1}{y2-y1}$$

$$\text{APPROX_G}[px][py] += g1 * \left(1 - \frac{px-x1}{x2-x1}\right) \frac{py-y1}{y2-y1}$$

$$\text{APPROX_B}[px][py] += b1 * \left(1 - \frac{px-x1}{x2-x1}\right) \frac{py-y1}{y2-y1}$$

$$\text{APPROX_R}[px][py] += r2 * \left(\frac{px-x1}{x2-x1}\right) \frac{py-y1}{y2-y1}$$

$$\text{APPROX_G}[px][py] += g2 * \left(\frac{px-x1}{x2-x1}\right) \frac{py-y1}{y2-y1}$$

$$\text{APPROX_B}[px][py] += b2 * \left(\frac{px-x1}{x2-x1}\right) \frac{py-y1}{y2-y1}$$

$$\text{APPROX_R}[px][py] += r3 * \left(1 - \frac{px-x1}{x2-x1}\right) \left(1 - \frac{py-y1}{y2-y1}\right)$$

$$\text{APPROX_G}[px][py] += g3 * \left(1 - \frac{px-x1}{x2-x1}\right) \left(1 - \frac{py-y1}{y2-y1}\right)$$

$$\text{APPROX_B}[px][py] += b3 * \left(1 - \frac{px-x1}{x2-x1}\right) \left(1 - \frac{py-y1}{y2-y1}\right)$$

$$\text{APPROX_R}[px][py] += r4 * \frac{px-x1}{x2-x1} \left(1 - \frac{py-y1}{y2-y1}\right)$$

$$\text{APPROX_G}[px][py] += g4 * \frac{px-x1}{x2-x1} \left(1 - \frac{py-y1}{y2-y1}\right)$$

$$\text{APPROX_B}[px][py] += b4 * \frac{px-x1}{x2-x1} \left(1 - \frac{py-y1}{y2-y1}\right)$$

19. Proszę napisać test jednostkowy obliczania błędu aproksymacji bitmapy z punktu 18 przygotowując hipotetyczne dane. Na przykład jak uzyskać całą czerwoną, całą zieloną, całą niebieską bitmapę? Jak uzyskać całą białą lub całą czarną bitmapę?