

CAD/CAE

Zadanie 3

Autor: Jan Augustyn

1. Wybrana bitmapa

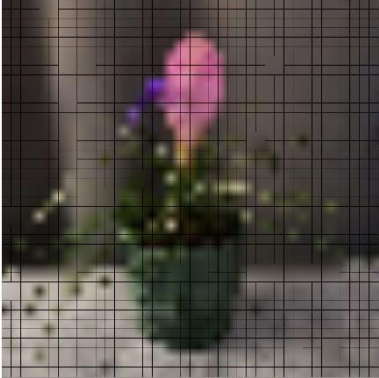

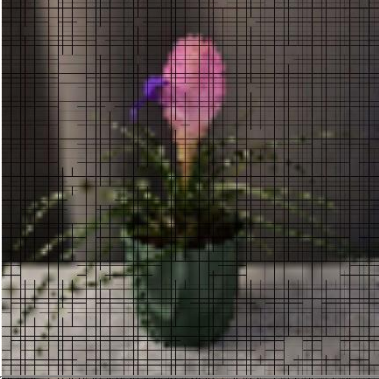

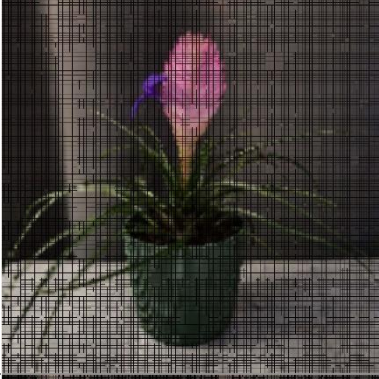

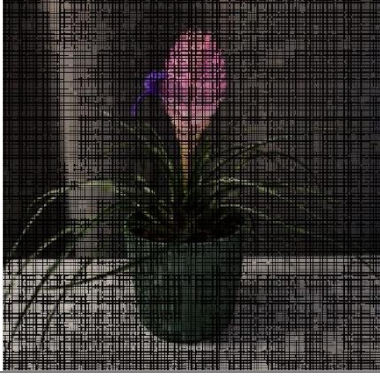



Rysunek 1 – Zdjęcie mojej nowej rośliny - Oplątwa niebieskawa. (1024x1025 px, 200.3 KB)

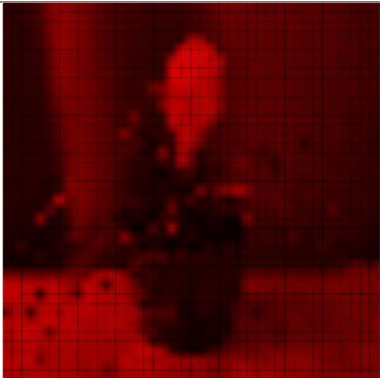

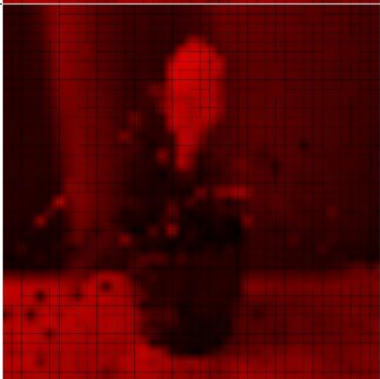

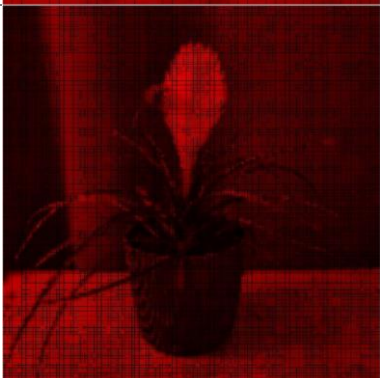
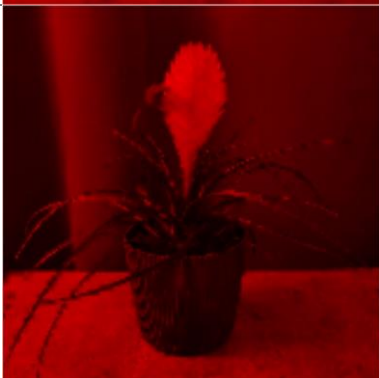
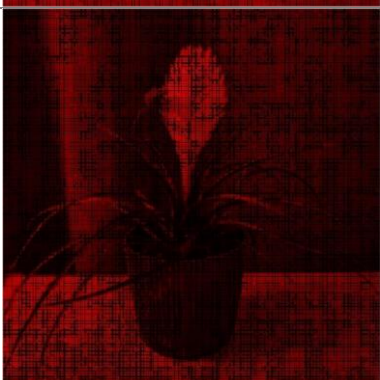
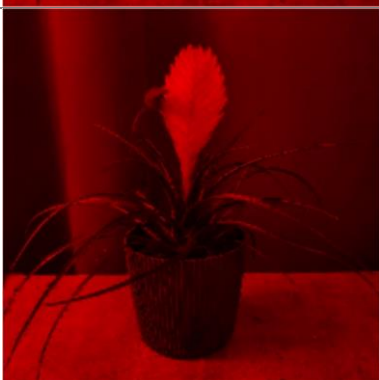
2. Dane wejściowe

- elementsx = 20
- elementsy= 20
- maxerror = 0.1
- max_refinement_level = 4

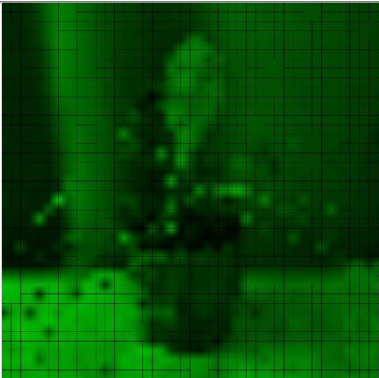
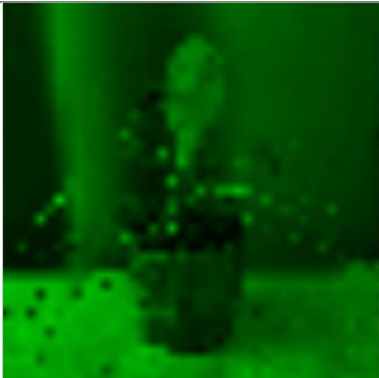

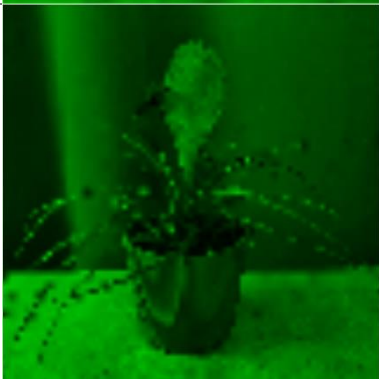
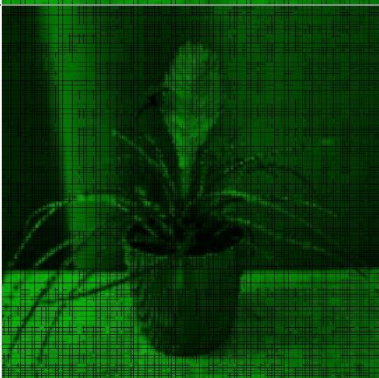

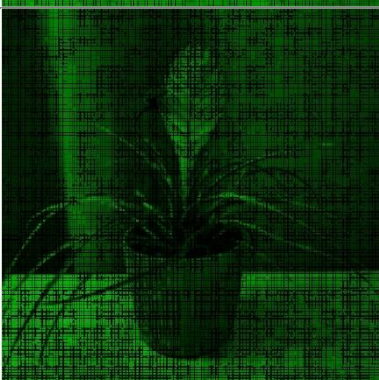

3. Sekwencje siatek oraz aproksymacje dla całej bitmapy

Iteracja	Aproksymacja z siatkami	Aproksymacja
1		
2		
3		
4		



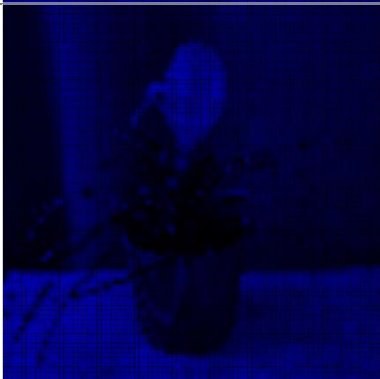




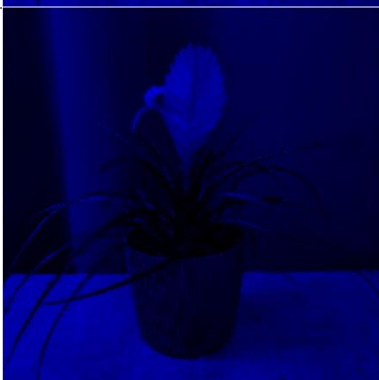
4. Sekwencje siatek oraz aproksymacje dla części Red

Iteracja	Aproksymacja z siatkami	Aproksymacja
1		
2		
3		
4		

5. Sekwencje siatek oraz aproksymacje dla części Green

Iteracja	Aproksymacja z siatkami	Aproksymacja
1		
2		
3		
4		

6. Sekwencje siatek oraz aproksymacje dla części Blue

Iteracja	Aproksymacja z siatkami	Aproksymacja
1		
2		
3		
4		

7. Modyfikacje kodu funkcji *bitmap_h*

```
function bitmap_h_unicolor(filename,elementsx,elementsy,maxerror, \
max_refinement_level,color_edges_black, color)
if_then_else = @(varargin) varargin{end - varargin{1}};
colors = {'r', 'g', 'b'};

% validate color
if (~ismember(colors, color))
    disp("Provided invalid color!")
    disp("Valid colors: red='r', green='g' or blue='b'.")
    return;
end

% read image from file
XX = imread(filename);

% extract red, green and blue components
RR = if_then_else(color=='r', XX(:,:,1), zeros(size(XX(:,:,1))));
GG = if_then_else(color=='g', XX(:,:,2), zeros(size(XX(:,:,2))));
BB = if_then_else(color=='b', XX(:,:,3), zeros(size(XX(:,:,3))));
```

Kod 1 - Fragment kodu 'bitmap_h_uniform.m' pozwalający na aproksymację bitmapy dla pojedynczego koloru.

```
refinemenet_level = refinemenet_level + 1;

% display iteration approx

RR_CP = RR;
GG_CP = GG;
BB_CP = BB;

% interpolate all active elements - recreate bitmap red green and blue compoments
for i=1:total_elements
    if (elements(i).active)
        interpolate_elem(i,color_edges_black);
    end
end

% recreate bitmap from red, green and blue compoments
RGB=XX;
RGB(:,:,1) = RR;
RGB(:,:,2) = GG;
RGB(:,:,3) = BB;

% display image
type = if_then_else(color_edges_black, 'Grid', 'Approx');
fig_name = sprintf("%s - Iteration: %d/%d, Img: '%s', Color: '%s', Max error: %g", ...
    type, refinemenet_level, max_refinement_level, filename, color, maxerror);
figure('Name', fig_name, 'NumberTitle', 'off');
imshow(RGB);
axis off;
img_path = sprintf("%s/%s_%d.jpg", color, type, refinemenet_level);
%saveas(gcf, img_path);
exportgraphics(gca, img_path, 'Resolution', 300);
RR = RR_CP;
GG = GG_CP;
BB = BB_CP;
end
```

Kod 2 - Fragment kodu pozwalający rysować wyniki aproksymacji w każdej iteracji i zapisywać w pliku '.jpg'.