

PRZETWARZANIE OBRAZÓW

Sprawozdanie nr 1

1. Zagadnienia teoretyczne.

Przesunięcie

Przesunięcie punktu $P(x, y)$ o wektor $D[dx, dy]$ polega na dodaniu do współrzędnych punktu P współrzędnych wektora d :

$$\begin{aligned}x_d &= x + d_x \\y_d &= y + d_y\end{aligned}$$

Korzystając ze współrzędnych jednorodnych możemy przesunięcie zapisać jako mnożenie współrzędnych punktu przez macierz przesunięcia:

$$\begin{pmatrix} x_d \\ y_d \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Skalowanie

Następną podstawową operacją geometryczną jest skalowanie figury. Przy założeniu, że figura opisana jest przez zbiór punktów (np. wielokąt), przemnożenie każdego z nich przez tzw. współczynniki skalowania s_x, s_y , da w efekcie odpowiednio figurę powiększoną (lub pomniejszoną). Macierz skalowania ma postać:

$$\begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

zatem cały proces skalowania punktu $P(x, y)$ można opisać następującym równaniem macierzy:

$$\begin{pmatrix} x_s \\ y_s \\ 1 \end{pmatrix} = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Obrót

Obrót punktu $P(x, y)$ o kąt względem początku układu współrzędnych opisuje następujący układ równań:

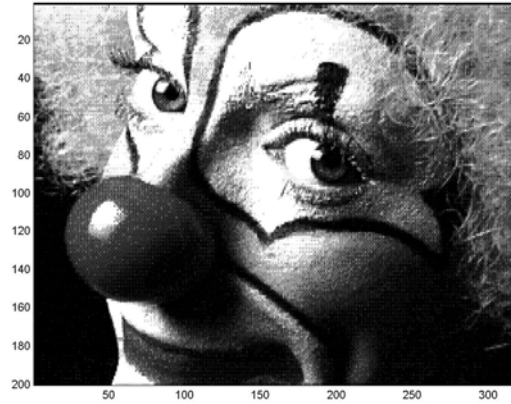
$$\begin{aligned}x_\alpha &= x \cos \alpha - y \sin \alpha \\y_\alpha &= x \sin \alpha + y \cos \alpha\end{aligned}$$

który po przekształceniu na postać macierzowa wygląda następująco:

$$\begin{pmatrix} x_\alpha \\ y_\alpha \\ 1 \end{pmatrix} = \begin{pmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

2. Realizacja zadań.

Wszystkie zadania zostały zrealizowane w programie MatLab 6.5 przy wykorzystaniu dołączonego do programu rysunku klauna w odcieniach szarości.

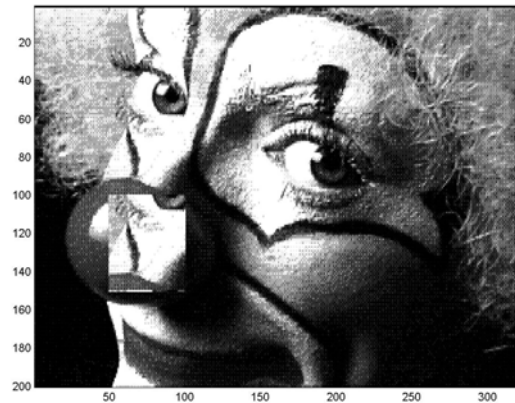


a) przesunięcie części obrazu

Przykład:

```
for x=0:50
    for y=0:50
        X(x+100,y+50)=X(x+50,y+50);
    end;
end;
```

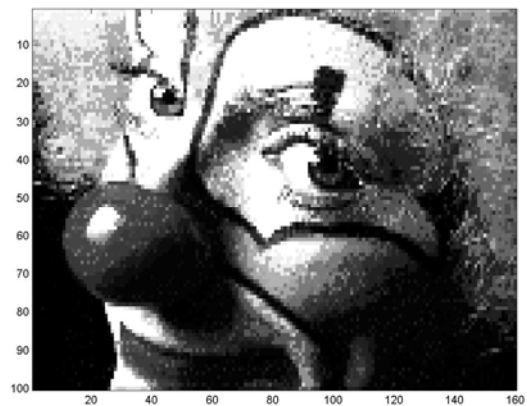
Prosty skrypt przesuwający fragment obrazka o 50 pixeli.



b) skalowanie obrazu

Przykład:

```
function skaluj (k, obraz)
[roz_y, roz_x] = size (obraz);
n_y=ceil(roz_y*k);
n_x=ceil(roz_x*k);
nowy_obraz = zeros (n_y, n_x);
for X=1:n_x,
    for Y=1:n_y,
        temp = [1/k, 0, 0; 0, 1/k, 0; 0, 0, 1] * [X; Y; 1];
        x=temp(1); y=temp(2);
        if x < 1; x = 1; elseif x > roz_x; x = roz_x; end;
```



k=0.5

```

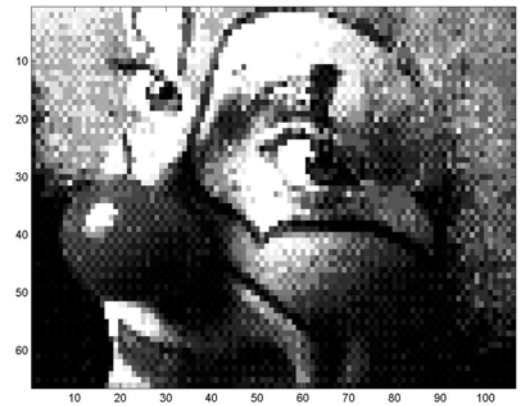
if y < 1; y = 1; elseif y > roz_y; y = roz_y; end;

fx = ceil(x);
fy = ceil(y);
cx=obraz(fy,fx);
nowy_obraz(Y, X) = cx;

end;
end;

colormap(gray);
image(nowy_obraz)

```



k=0.33

c) obracanie obrazu

Przykład:

```

function obroc (kat, obraz)

s=sin(kat);
c=cos(kat);

[roz_y, roz_x] = size (obraz);

if (s >= 0 && c >= 0)
    px = round(roz_y * s);
    py = 0;

elseif (s <= 0 && c <= 0)
    px = round(c*(-roz_x));
    py = round(c*(-roz_y) - s * roz_x);

elseif (s >= 0 && c <= 0)
    px = round(roz_y * s - roz_x*c);
    py = round(roz_y*(-c));

else
    px = 0;
    py = round(s*(-roz_x));
end

Rozn_y = ceil(abs(roz_y * c) + abs(roz_x * s));
Rozn_x = ceil(abs(roz_x * c) + abs(roz_y * s));
nowy_obraz = zeros (Rozn_y, Rozn_x);

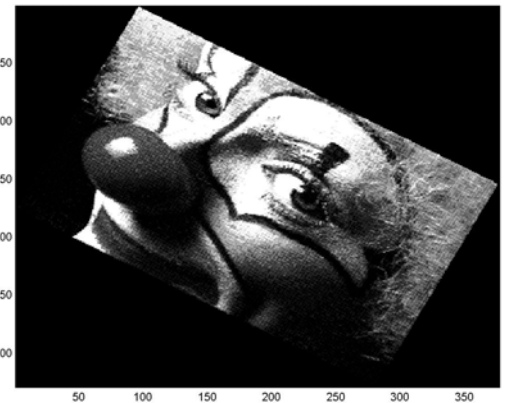
od=1-px; do=Rozn_x-px;
od_y=1-py; do_y=Rozn_y-py;

for X=od:do,
    for Y=od_y:do_y,

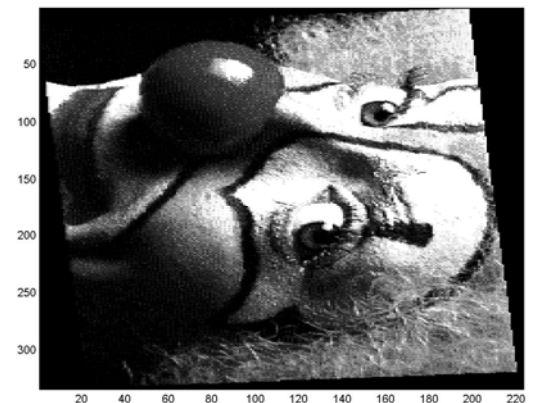
        temp = [c, s, 0; -s, c, 0; 0, 0, 1] * [X; Y; 1];

        x = temp(1); y = temp(2);

```



kat=0.5



kat=1.5

```

if (x < 1 || y < 1 || x > roz_x || y > roz_y)
    continue;
end

fx = ceil(x);
fy = ceil(y);
cx=obraz(fy,fx);
nowy_obraz(Y+py, X+px) = cx;
end;
end;

colormap(gray);
image (nowy_obraz);

```

3. Wnioski z zadań.

Najprostszym zadaniem było wykonanie przesunięcia części obrazu w drugie miejsce, polegało ono jedynie na przekopiowaniu pixeli z jednej lokacji do drugiej. Natomiast kolejne zadania nie należały do najprostszych (chyba, że ktoś lubi MatLaba;). Skalowanie obrazka wymaga wybieranie tych pixeli, które mają pozostać na nowym rysunku (zależnie czy powiększamy czy zmniejszamy rysunek). Obrazek po skalowaniu (kolejne pixele) są umieszczane na osobnym 'buforze' o wymiarach, jakie ma mieć obraz po zeskalowaniu, co eliminuje kilka drobnych problemów. Ostatnie zadanie polegało na obrocie obrazka o dowolny kąt, tutaj każdy pixel jest przeskalowany z żądanym kątem. Oczywiście i w tym przypadku jest wykorzystywany nowy 'bufor' o wymiarach niezbędnych do poprawnego obrócenia obrazu.

Problemem, który napotkałem podczas pisania skryptów było indexowanie MatLaba od jedynki, a także wychodzenie podczas obliczeń wartości ujemnych podczas obracania obrazu. Rozwiązaniem tego problemu są linie skryptu od 8 do 23. Wyniki działania skryptów są pokazane obok źródeł skryptów, nie są one wielce efektowne, ale przedstawiają założenia, które miały zostać zrobione.