

Part 1: (50%) Geometric Transformation

程式碼:

```
function varargout = HW01(varargin)

gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @HW01_OpeningFcn, ...
                  'gui_OutputFcn',  @HW01_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function HW01_OpeningFcn(hObject, eventdata, handles, varargin)

handles.output = hObject;
guidata(hObject, handles);

function varargout = HW01_OutputFcn(hObject, eventdata, handles)

varargout{1} = handles.output;

function pushbutton1_Callback(hObject, eventdata, handles)

% 讀取圖片
filename = uigetfile({'*.jpg;*.tif;*.png;*.gif;*.bmp','All Image Files'});
handles.I = imread(filename);

% 轉成灰階
```

```
[handles.row, handles.col, handles.lay] = size(handles.I);
```

```
if handles.lay == 1
```

```
    handles.I = handles.I;
```

```
    fprintf('1 layers image\n');
```

```
elseif handles.lay == 3
```

```
    R = uint8(handles.I(:, :, 1));
```

```
    G = uint8(handles.I(:, :, 2));
```

```
    B = uint8(handles.I(:, :, 3));
```

```
    handles.I = 0.299*R + 0.587*G + 0.114*B;
```

```
    fprintf('3 layers image\n');
```

```
end
```

```
axes(handles.axes1); imshow(handles.I);
```

```
handles.output = hObject;
```

```
guidata(hObject, handles);
```

```
function pushbutton2_Callback(hObject, eventdata, handles)
```

```
% Trapezoidal Transformation
```

```
img = handles.I;
```

```
[row, col] = size(handles.I);
```

```
img_new = zeros(row, col);
```

```
for i = 1:row
```

```
    for j = 1:col
```

```
        new_x = round( 3*i/4 + j*i/(col*row) );
```

```
        new_y = round( j+i/4 - j*i/(2*col) );
```

```
        img_new(new_x, new_y) = img(i, j);
```

```
    end
```

```
end
```

```
axes(handles.axes2); imshow(uint8(img_new));
```

```
function pushbutton3_Callback(hObject, eventdata, handles)
```

```
% Wavy Transformation
```

```
img = handles.I;
```

```
[row, col] = size(handles.I);
```

```

img_new = zeros(row, col);

for i = 1:row
    for j = 1:col
        new_x = round(j-32*sin( i/32 ));
        new_y = round(i-32*sin( j/32 ));
        if new_x >= 1 && new_x <= row && new_y >= 1 && new_y <= col
            img_new(j, i) = img(new_x, new_y);
        end
    end
end

axes(handles.axes3); imshow(uint8(img_new));

```

```

function pushbutton4_Callback(hObject, eventdata, handles)

```

```

% Circular Transformation

```

```

img = handles.I;
[row, col] = size(handles.I);
img_new = zeros(row, col);

for i = 1:row
    for j = 1:col
        d = sqrt( (row/2)^2-(row/2-i)^2 );
        new_x = round( (j-col/2) * col/(d*2) + col/2 );
        new_y = i;
        if new_x >= 1 && new_x <= col && new_y >= 1 && new_y <= col
            img_new(i, j) = img(new_y, new_x);
        end
    end
end

axes(handles.axes4); imshow(uint8(img_new));

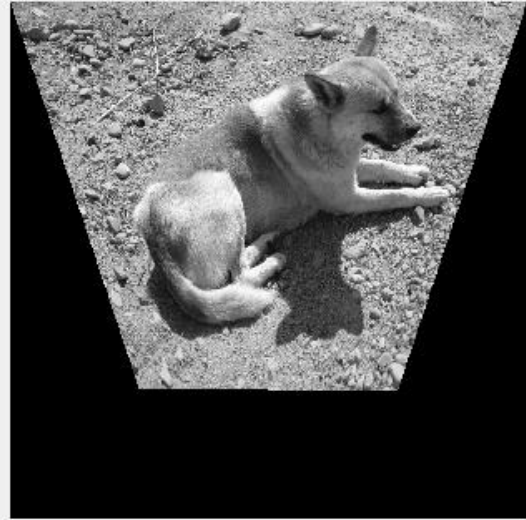
```

結果呈現與討論:

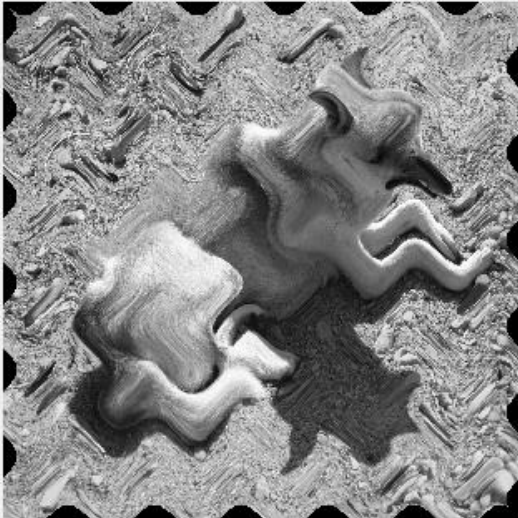
Original Image



Trapezoidal Transformation



Wavy Transformation



Circular Transformation



Trapezoidal Transformation 利用座標的轉換關係，將 pixel 移至梯形。Wavy Transformation 利用 \sin 函數，將縱橫的 pixel 重新排成 \sin 波的形狀。最後 Circular Transformation 計算每一列的三角函數，將每一列的 pixel 壓縮成圓形的數量排列。

Part 2: (50%) Hough Transform

```
function varargout = HW02(varargin)
```

```
gui_Singleton = 1;
```

```
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @HW02_OpeningFcn, ...
                  'gui_OutputFcn',  @HW02_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
```

```

if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if narginout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function HW02_OpeningFcn(hObject, eventdata, handles, varargin)

handles.output = hObject;
guidata(hObject, handles);

function varargout = HW02_OutputFcn(hObject, eventdata, handles)

varargout{1} = handles.output;

function pushbutton1_Callback(hObject, eventdata, handles)

[row, col] = size(handles.I);

% 算出面積周長
imgfltrd = handles.imgfltrd;
[accum, axis_rho, axis_theta, lineprm, lineseg] = Hough_Grd(imgfltrd);
[line] = DrawLines_Polar(size(imgfltrd), lineprm);
for i = 1:1:size(lineprm, 1)-4
    a = [3 4 7 8];
    p2(i, 1:2) = polyfit(line(:,2*a(i)-1),line(:,2*a(i)),1);
end

pad2 = zeros(row, col);
for i = 1:row
    for j = 1:col
        if j-polyval(p2(1, 1:2),i) > 0 && j-polyval(p2(2, 1:2),i) < 0 &&...
            j-polyval(p2(3, 1:2),i) < 0 && j-polyval(p2(4, 1:2),i) > 0
            pad2(i, j) = 1;
        end
    end
end

```

```

    end
end
pad2 = imrotate(pad2, 90);
data = regionprops(pad2, 'all');
Area = data.Area*(0.5^2);
Perimeter = data.Perimeter*0.5;

set(handles.text2, 'String', ['Area: ', num2str(Area), ' mm^2']);
set(handles.text3, 'String', ['Perimeter: ', num2str(Perimeter), ' mm']);

% 顯示圖片
imgfltrd = imrotate(handles.imgfltrd, 180);
[accum, axis_rho, axis_theta, lineprm, lineseg] = Hough_Grd(imgfltrd);
[line] = DrawLines_Polar(size(imgfltrd), lineprm);

cla(handles.axes2, 'reset');
axes(handles.axes2);
set(gca, 'xdir', 'reverse');
hold on;
imagesc(imgfltrd); colormap('gray'); axis image;
for i = 1:1:size(lineprm, 1)-4
    a = [3 4 7 8];
    plot(line(:, 2*a(i)-1), line(:, 2*a(i)), 'LineWidth', 2);
end
hold off;

handles.output = hObject;
guidata(hObject, handles);

function pushbutton2_Callback(hObject, eventdata, handles)

[row, col] = size(handles.I);

% 算出面積周長
imgfltrd = handles.imgfltrd;
[accum, axis_rho, axis_theta, lineprm, lineseg] = Hough_Grd(imgfltrd);
[line] = DrawLines_Polar(size(imgfltrd), lineprm);
for i = 1:1:size(lineprm, 1)-4
    a = [1 2 5 6];
    p(i, 1:2) = polyfit(line(:, 2*a(i)-1), line(:, 2*a(i)), 1);
end

```

```

pad = zeros(row, col);
for i = 1:row
    for j = 1:col
        if j-polyval(p(1, 1:2),i) > 0 && j-polyval(p(2, 1:2),i) < 0 &&...
            j-polyval(p(3, 1:2),i) > 0 && j-polyval(p(4, 1:2),i) < 0
            pad(i, j) = 1;
        end
    end
end
end
pad = imrotate(pad, 90);

data = regionprops(pad, 'all');
Area = data.Area*(0.5^2);
Perimeter = data.Perimeter*0.5;

set(handles.text4, 'String', ['Area: ', num2str(Area), ' mm^2']);
set(handles.text5, 'String', ['Perimeter: ', num2str(Perimeter), ' mm']);

% 顯示圖片
imgfltrd = imrotate(handles.imgfltrd, 180);
[accum, axis_rho, axis_theta, lineprm, lineseg] = Hough_Grd(imgfltrd);
[line] = DrawLines_Polar(size(imgfltrd), lineprm);

cla(handles.axes2,'reset');
axes(handles.axes2);
set(gca,'xdir','reverse');
hold on;
imagesc(imgfltrd); colormap('gray'); axis image;
for i = 1:1:size(lineprm, 1)-4
    a = [1 2 5 6];
    plot(line(:,2*a(i)-1), line(:,2*a(i)), 'LineWidth', 2);
    p(i, 1:2) = polyfit(line(:,2*a(i)-1),line(:,2*a(i)),1);
end
hold off;

handles.output = hObject;
guidata(hObject, handles);

function pushbutton3_Callback(hObject, eventdata, handles)

```

```

% 讀取圖片
filename = uigetfile({'*.jpg;*.tif;*.png;*.gif;*.bmp','All Image Files'});
handles.I = imread(filename);

% 轉成灰階
[handles.row, handles.col, handles.lay] = size(handles.I);

if handles.lay == 1
    handles.I = handles.I;
    fprintf('1 layers image\n');
elseif handles.lay == 3
    R = uint8(handles.I(:, :, 1));
    G = uint8(handles.I(:, :, 2));
    B = uint8(handles.I(:, :, 3));
    handles.I = 0.299*R + 0.587*G + 0.114*B;
    fprintf('3 layers image\n');
end

axes(handles.axes1); imshow(handles.I);

handles.output = hObject;
guidata(hObject, handles);

function pushbutton5_Callback(hObject, eventdata, handles)

filter = [1 1 1 1 1; 1 2 2 2 1; 1 2 4 2 1; 1 2 2 2 1; 1 1 1 1 1];
filter = filter / sum(filter(:));
imgfltrd = filter2( filter , handles.I );

[accum, axis_rho, axis_theta, lineprm, lineseg] = Hough_Grd(imgfltrd);

% 呈現霍夫轉換累積矩陣
axes(handles.axes3);
imshow(imadjust(mat2gray(accum*255)), [], 'XData',axis_theta*(180/pi), 'YData',axis_rho, ...
'InitialMagnification','fit');
xlabel('\theta (degrees)'), ylabel('\rho');
axis on, axis normal, hold on;
title('Accumulation Array from Hough Transform');
colormap(hot), colorbar;

% 劃出直線

```



```

hold on;
axes(handles.axes2);
imagesc(imgfltrd); colormap('gray'); axis image;
[line] = DrawLines_Polar(size(imgfltrd), lineprm);
title('Image (Blurred) with Lines Detected');
hold off;

```

```
handles.imgfltrd = imgfltrd;
```

```

handles.output = hObject;
guidata(hObject, handles);

```

結果呈現與討論：

