

# Praktikum 5

## Image Transformation (1)

### 1. Resizes image

Program berikut ini menunjukkan fungsi untuk merubah ukuran (memperkecil atau memperbesar) sebuah gambar.

```
#include <cv.h> //main OpenCV header
#include <highgui.h> //GUI header

int main()
{
    // Set up images
    const char* name = "Resize Image";
    IplImage* img = cvLoadImage("image.jpg", 3);
    IplImage* out = cvCreateImage(
        cvSize(
            img->width/2,
            img->height/2
        ),
        IPL_DEPTH_8U, 3 );

    // Show original
    cvNamedWindow( "Original", 1 );
    cvShowImage( "Original", img );

    // Perform histogram equalization
    cvResize( img, out, CV_INTER_LINEAR );

    // Show histogram equalized
    cvNamedWindow( name, 1 );
    cvShowImage( name, out );

    cvWaitKey();
    cvReleaseImage( &img );
    cvReleaseImage( &out );
    return 0;
}
```

Petunjuk praktikum:

- Ubah parameter pada `cvResize()` pada program di atas dengan metode interpolasi lainnya, kemudian amati perubahan pada gambar.
- Ubahlah program di atas menjadi program untuk memperbesar gambar.
- Jelaskan fungsi berikut ini beserta dengan parameter yang ada di dalamnya.

- o `cvResize()`

## 2. Affine Transform with Trackbar

Program berikut ini menunjukkan fungsi untuk melakukan *warp* (membengkokkan), *stretch* (melebarkan), *rotate* (memutar) dan *resize* (merubah ukuran) pada sebuah gambar.

```
#include <cv.h>
#include <cxcore.h>
#include <highgui.h>

int angle_switch_value = 0;
int angleInt = 0;
int scale_switch_value = 0;
int scaleInt = 0;

void switch_callback_a( int position ){
    angleInt = position;
}

void switch_callback_s( int position ){
    scaleInt = position;
}

int main(int argc, char**)
{
    // Set up variables
    cvPoint2D32f srcTri[3], dstTri[3];
    cvMat* rot_mat = cvCreateMat(2,3,CV_32FC1);
    cvMat* warp_mat = cvCreateMat(2,3,CV_32FC1);
    IplImage *src, *dst;
    const char* name = "Affine_Transform";

    // Load image
    src=cvLoadImage("image.jpg");
    dst = cvCloneImage( src );
    dst->origin = src->origin;
    cvZero( dst );
    cvNamedWindow( name, 1 );

    // Create angle and scale
    double angle = 0.0;
    double scale = 1.0;

    // Create trackbars
    cvCreateTrackbar( "Angle", name, &angle_switch_value, 4, switch_callback_a );
    cvCreateTrackbar( "Scale", name, &scale_switch_value, 4, switch_callback_s );

    // Compute warp matrix
    srcTri[0].x = 0;
    srcTri[0].y = 0;
    srcTri[1].x = src->width - 1;
    srcTri[1].y = 0;
    srcTri[2].x = 0;
    srcTri[2].y = src->height - 1;

    dstTri[0].x = src->width*0.0;
    dstTri[0].y = src->height*0.25;
    dstTri[1].x = src->width*0.90;
    dstTri[1].y = src->height*0.15;
    dstTri[2].x = src->width*0.10;
    dstTri[2].y = src->height*0.75;

    cvGetAffineTransform( srcTri, dstTri, warp_mat );
    cvWarpAffine( src, dst, warp_mat );
    cvCopy ( dst, src );

    while( 1 ) {
        switch( angleInt ){
            case 0:
                angle = 0.0;
                break;
            case 1:
                angle = 20.0;
                break;
        }
    }
}
```

```

        case 2:
            angle = 40.0;
            break;
        case 3:
            angle = 60.0;
            break;
        case 4:
            angle = 90.0;
            break;
    }
    switch( scaleInt ){
        case 0:
            scale = 1.0;
            break;
        case 1:
            scale = 0.8;
            break;
        case 2:
            scale = 0.6;
            break;
        case 3:
            scale = 0.4;
            break;
        case 4:
            scale = 0.2;
            break;
    }

    // Compute rotation matrix
    cvPoint2D32f center = cvPoint2D32f( src->width/2, src->height/2 );
    cv2DRotationMatrix( center, angle, scale, rot_mat );

    // Do the transformation
    cvWarpAffine( src, dst, rot_mat );

    cvShowImage( name, dst );

    if( cvWaitKey( 15 ) == 27 )
        break;
}

cvReleaseImage( &dst );
cvReleaseMat( &rot_mat );
cvReleaseMat( &warp_mat );

return 0;
}

```

### Petunjuk praktikum:

- Ubah parameter pada `cvResize()` pada program di atas dengan metode interpolasi lainnya, kemudian amati perubahan pada gambar.
- Ubahlah program diatas menjadi program untuk memperbesar gambar.
- Jelaskan fungsi berikut ini beserta dengan parameter yang ada di dalamnya.
  - `cvZero()`
  - `cvCopy()`
  - `cv2DRotationMatrix()`
  - `cvGetAffineTransform()`
  - `cvWarpAffine()`

### 3. Animation: Rotate and Resize

Program berikut ini menunjukkan animasi untuk menggabungkan fungsi memutar sekaligus merubah sebuah gambar.

```
#include <cv.h>
#include <highgui.h>
#include <math.h>

int main()
{
    IplImage* src;
    IplImage* dst;

    int delta;
    int angle;

    src = cvLoadImage("image.jpg", 1);
    dst = cvCloneImage( src );
    delta = 1; angle = 0;
    cvNamedWindow( "src", 1 );
    cvShowImage( "src", src );

    while(1)
    {
        float m[6];
        double factor = (cos(angle*CV_PI/180.) + 1.1)*3;
        CvMat M = cvMat( 2, 3, CV_32F, m );

        int w = src->width;
        int h = src->height;

        m[0] = (float)(factor*cos(-angle*2*CV_PI/180.));
        m[1] = (float)(factor*sin(-angle*2*CV_PI/180.));
        m[2] = w*0.5f;
        m[3] = -m[1];
        m[4] = m[0];
        m[5] = h*0.5f;

        cvGetQuadrangleSubPix( src, dst, &M);
        cvNamedWindow( "dst", 1 ); cvShowImage( "dst", dst );

        if( cvWaitKey(5) == 27 )
            break;
        angle = (angle + delta) % 360;
    }
    return 0;
}
```

Petunjuk praktikum:

- Jelaskan algoritma pada program diatas khususnya yang berada dalam perulangan while(1).
- Jelaskan fungsi berikut ini beserta dengan parameter yang ada di dalamnya.
  - cvMat()
  - cvGetQuadrangleSubPix()

## 4. Morphological Transformation (Erode and Dilate)

Program berikut ini menunjukkan pemrosesan morfologi pada gambar dengan fungsi erode dan dilate.

```
#include <cv.h>
#include <cxcore.h>
#include <highgui.h>

int main()
{
    IplImage* newImg = NULL;
    IplImage* dilateImg = NULL;
    IplImage* erodeImg = NULL;
    cvNamedWindow("src", 1);
    cvNamedWindow("dilate",1);
    cvNamedWindow("erode",1);

    //load original image
    newImg = cvLoadImage("image.jpg",0);
    cvShowImage( "src", newImg );

    //make a copy of the original image
    dilateImg=cvCloneImage( newImg );
    erodeImg=cvCloneImage( newImg );

    //dilate image
    cvDilate(newImg,dilateImg,NULL,4);

    //erode image
    cvErode(newImg,erodeImg,NULL,4);

    cvShowImage( "dilate", dilateImg );
    cvShowImage( "erode", erodeImg );

    cvWaitKey(0);
    cvDestroyWindow( "src" );
    cvDestroyWindow( "dilate" );
    cvDestroyWindow( "erode" );
    cvReleaseImage( &newImg );
    cvReleaseImage( &dilateImg );
    cvReleaseImage( &erodeImg );

    return 0;
}
```

Petunjuk praktikum:

- Jelaskan konsep morphological processing pada gambar dengan menggunakan fungsi erode dan dilate.
- Jelaskan fungsi berikut ini beserta dengan parameter yang ada di dalamnya.
  - cvErode()
  - cvDilate()

## 5. Morphological Transformation with Trackbar

Program berikut ini menunjukkan pemrosesan morfologi pada gambar dengan fungsi Open-Close dan Erode-Dilate dengan menggunakan beberapa bentuk transformasi (rectangle, ellipse, cross).

```
#include <cv.h>
#include <highgui.h>
#include <stdlib.h>
#include <stdio.h>

IplImage* src = 0;
IplImage* dst = 0;

IplConvKernel* element = 0;
int element_shape = CV_SHAPE_RECT;

//the address of variable which receives trackbar position update
int max_iters = 10;
int open_close_pos = 0;
int erode_dilate_pos = 0;

// callback function for open/close trackbar
void OpenClose(int pos)
{
    int n = open_close_pos - max_iters;
    int an = n > 0 ? n : -n;
    element = cvCreateStructuringElementEx( an*2+1, an*2+1, an, an, element_shape, 0
);
    if( n < 0 )
    {
        cvErode(src,dst,element,1);
        cvDilate(dst,dst,element,1);
    }
    else
    {
        cvDilate(src,dst,element,1);
        cvErode(dst,dst,element,1);
    }
    cvReleaseStructuringElement(&element);
    cvShowImage("Open/Close",dst);
}

// callback function for erode/dilate trackbar
void ErodeDilate(int pos)
{
    int n = erode_dilate_pos - max_iters;
    int an = n > 0 ? n : -n;
    element = cvCreateStructuringElementEx( an*2+1, an*2+1, an, an, element_shape, 0
);
    if( n < 0 )
    {
        cvErode(src,dst,element,1);
    }
    else
    {
        cvDilate(src,dst,element,1);
    }
    cvReleaseStructuringElement(&element);
    cvShowImage("Erode/Dilate",dst);
}

int main( int argc, char** argv )
{
    char* filename = argc == 2 ? argv[1] : (char*)"image.jpg";
    if( (src = cvLoadImage(filename,1)) == 0 )
        return -1;

    printf( "Hot keys: \n"
           "\tESC - quit the program\n"
           "\tr - use rectangle structuring element\n"
    );
}
```

```

        "\te - use elliptic structuring element\n"
        "\tc - use cross-shaped structuring element\n"
        "\tSPACE - loop through all the options\n" );

dst = cvCloneImage(src);

//create windows for output images
cvNamedWindow("Open/Close",1);
cvNamedWindow("Erode/Dilate",1);

open_close_pos = erode_dilate_pos = max_iters;
cvCreateTrackbar("iterations",
"Open/Close", &open_close_pos, max_iters*2+1, OpenClose);
cvCreateTrackbar("iterations",
"Erode/Dilate", &erode_dilate_pos, max_iters*2+1, ErodeDilate);

for(;;)
{
    int c;

    OpenClose(open_close_pos);
    ErodeDilate(erode_dilate_pos);
    c = cvWaitKey(0);

    if( (char)c == 27 )
        break;
    if( (char)c == 'e' )
        element_shape = CV_SHAPE_ELLIPSE;
    else if( (char)c == 'r' )
        element_shape = CV_SHAPE_RECT;
    else if( (char)c == 'c' )
        element_shape = CV_SHAPE_CROSS;
    else if( (char)c == ' ' )
        element_shape = (element_shape + 1) % 3;
}

//release images
cvReleaseImage(&src);
cvReleaseImage(&dst);

//destroy windows
cvDestroyWindow("Open/Close");
cvDestroyWindow("Erode/Dilate");

return 0;
}

```

Petunjuk praktikum:

- Ubah bentuk transformasi dengan menekan tombol berikut ini:
  - "e" - ellipse
  - "r" - rectangular
  - "c" - cross
  - "space" - loop iteration
geser trackbar dengan iterasi yang berbeda-beda. Kemudian amati perbedaannya.
- Jelaskan konsep morphological processing Open-Close pada program diatas, jelaskan perbedaannya dengan fungsi Erode-Dilate.

## Tugas: Advanced Geometrical Transformation

Buatlah program untuk transformasi geometri menggunakan fungsi berikut ini:

- `cvWarpPerspective()`
- `cvRemap()`
- `cvLogPolar()`