

تغییر اندازه، و چرخش تصاویر

در OpenCV با استفاده از مفهومی به نام **affine transform**، امکان تغییر اندازه و همچنین چرخش تصاویر میسر می‌شود. در اینجا، تصویر در یک ماتریس دو در سه ضرب می‌شود تا انتقالات یاد شده، انجام شوند.

```
private static void rotateImage(double angle, double scale, Mat src, Mat dst)
{
    var imageCenter = new Point2f(src.Cols / 2f, src.Rows / 2f);
    var rotationMat = Cv2.GetRotationMatrix2D(imageCenter, angle, scale);
    Cv2.WarpAffine(src, dst, rotationMat, src.Size());
}
```

متد فوق کار چرخش تصویر مبدا (src) را به تصویر مقصد (dst) انجام می‌دهد. این عملیات توسط متد WarpAffine مدیریت شده و مهم‌ترین پارامتر آن، پارامتر سوم آن است که ماتریس تعریف کننده‌ی انتقالات تعریف شده توسط متد [GetRotationMatrix2D](#) است. در اینجا مرکز مشخص شده، زاویه و مقیاس، نحوه‌ی چرخش را تعریف می‌کنند. برای مشاهده‌ی بهتر تاثیر پارامترهای مختلف در اینجا، به مثال ذیل دقت کنید:

```
using OpenCvSharp;
using OpenCvSharp.CPlusPlus;

namespace OpenCVSharpSample09
{
    class Program
    {
        static void Main(string[] args)
        {
            using (var src = new Mat(@"..\..\Images\Penguin.Png", LoadMode.AnyDepth |
LoadMode.AnyColor))
            using (var dst = new Mat())
            {
                src.CopyTo(dst);

                using (var window = new Window("Resize/Rotate/Blur",
                    image: dst, flags: WindowMode.AutoSize))
                {
                    var angle = 0.0;
                    var scale = 0.7;

                    var angleTrackbar = window.CreateTrackbar(
                        name: "Angle", value: 0, max: 180,
                        callback: pos =>
                        {
                            angle = pos;
                            rotateImage(angle, scale, src, dst);
                            window.Image = dst;
                        });

                    var scaleTrackbar = window.CreateTrackbar(
                        name: "Scale", value: 1, max: 10,
                        callback: pos =>
                        {
                            scale = pos / 10f;
                            rotateImage(angle, scale, src, dst);
                            window.Image = dst;
                        });

                    angleTrackbar.Callback.DynamicInvoke(0);
                    scaleTrackbar.Callback.DynamicInvoke(1);

                    Cv2.WaitKey();
                }
            }
        }

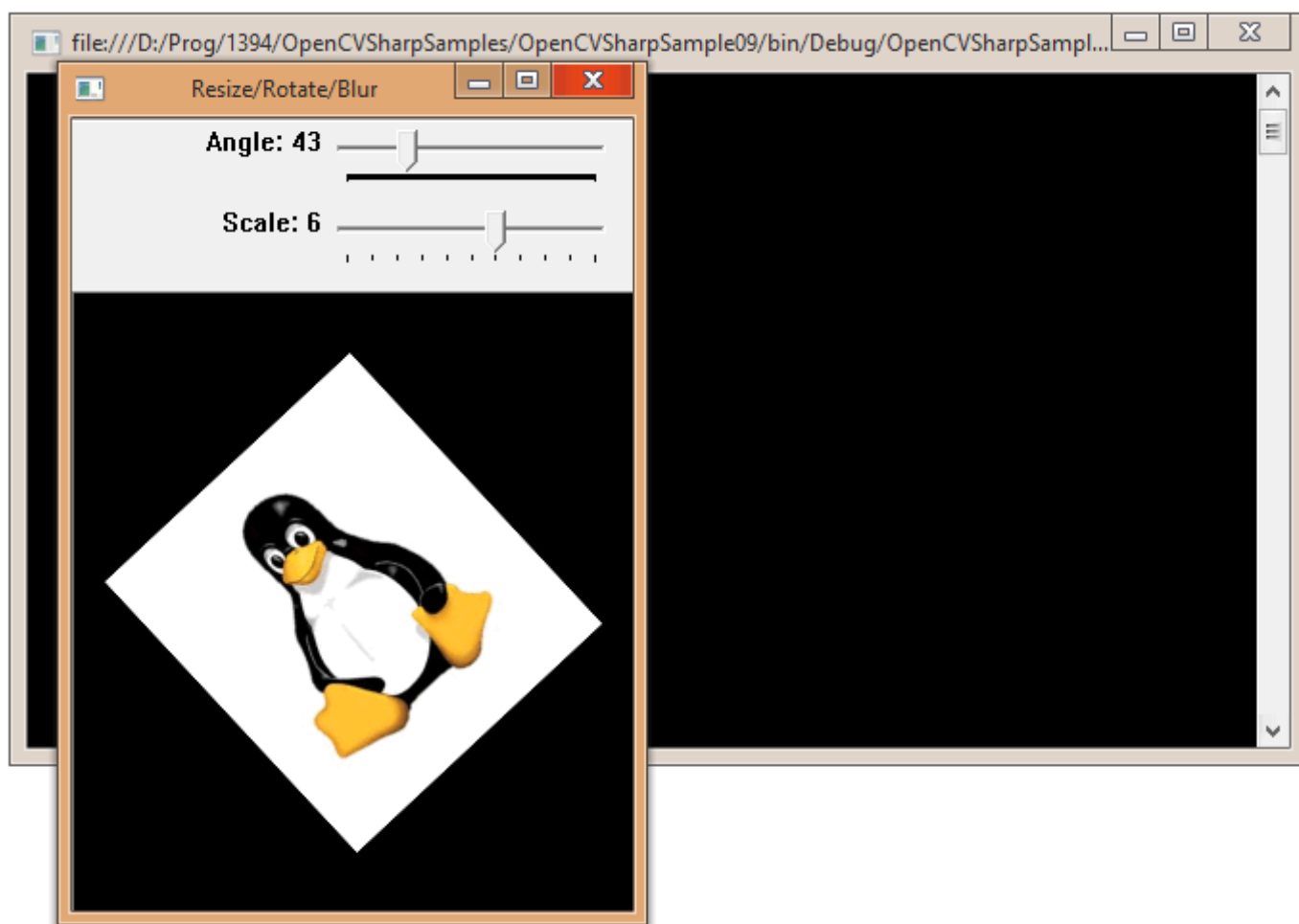
        private static void rotateImage(double angle, double scale, Mat src, Mat dst)
```

```

    {
        var imageCenter = new Point2f(src.Cols / 2f, src.Rows / 2f);
        var rotationMat = Cv2.GetRotationMatrix2D(imageCenter, angle, scale);
        Cv2.WarpAffine(src, dst, rotationMat, src.Size());
    }
}

```

با این خروجی:



در این مثال، مانند مطلب [قسمت قبل](#)، ابتدا یک پنجره‌ی سازگار با C++ API ایجاد شده و سپس دو tracker به آن اضافه شده‌اند. این trackers کار دریافت ورودی اطلاعات را از کاربر به عهده دارند (دریافت مقادیر زاویه‌ی چرخش و مقیاس) و مقادیر دریافتی از آن‌ها، در نهایت به متد rotateImage ارسال می‌شوند. این متد کار چرخش و تغییر مقیاس تصویر اصلی را انجام داده و نتیجه را به تصویر dst کپی می‌کند. در آخر تصویر dst در پنجره به روز شده و نمایش داده می‌شود.

تغییر اندازه‌ی تصاویر

اگر صرفاً قصد تغییر اندازه‌ی تصاویر را دارید (بدون چرخش آن‌ها)، متد ویژه‌ای به نام [Resize](#) برای این منظور تدارک دیده شده‌است:

```

var resizeTrackbar = window.CreateTrackbar(
    name: "Resize", value: 1, max: 100,

```

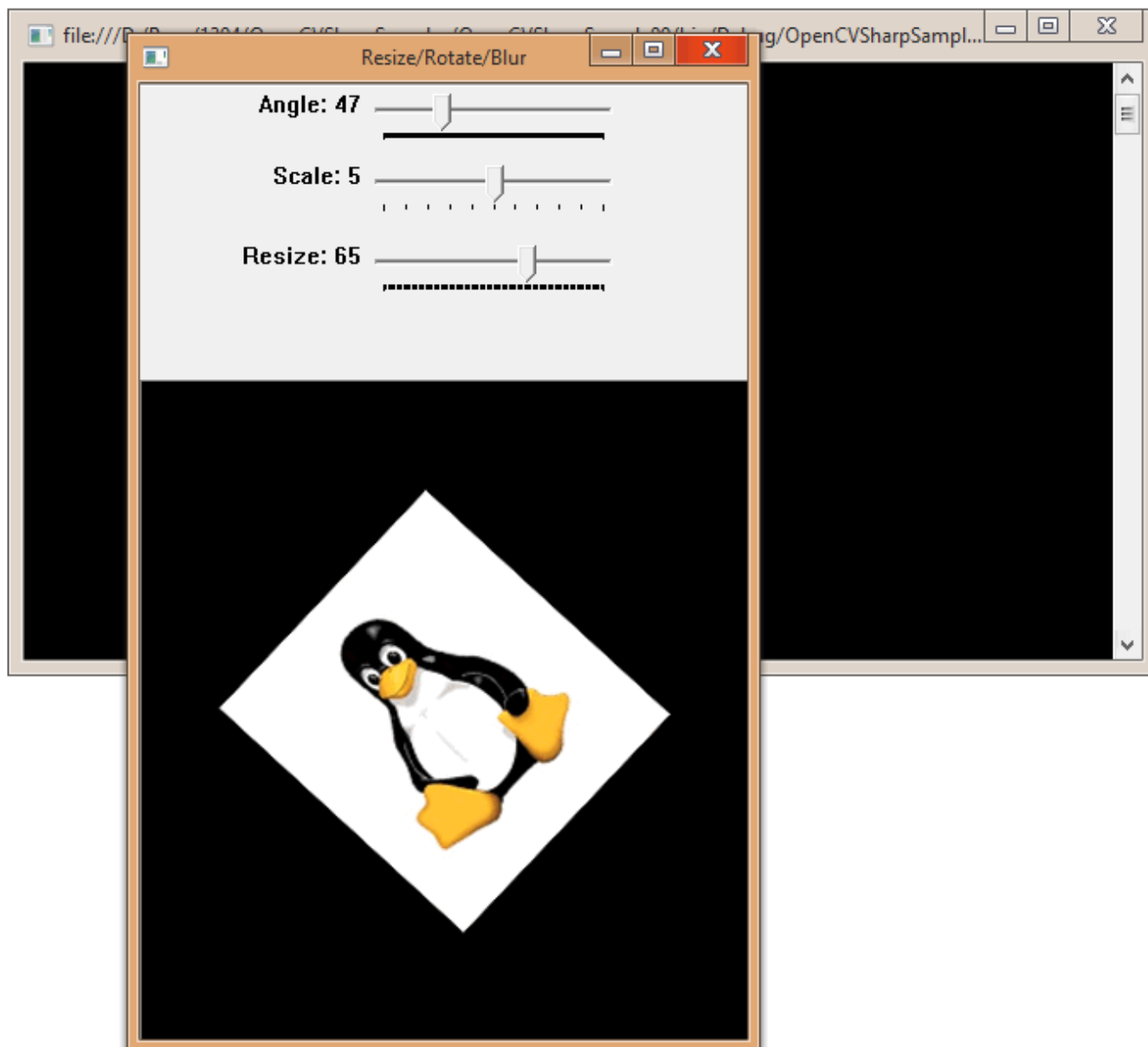
```
callback: pos =>
{
    Cv2.Resize(src, dst,
        new Size(src.Width + pos, src.Height + pos),
        interpolation: Interpolation.Cubic);
    window.Image = dst;
});
```

در اینجا یک tracker دیگر به پنجره‌ی اصلی اضافه شده و توسط آن کار تعیین تغییر اندازه‌ی تصویر انجام می‌شود. نکته‌ی مهم این متد، امکان تعیین الگوریتم تغییر اندازه است که برای مثال در اینجا از Interpolation.Cubic استفاده شده‌است (احتمالا با این نام‌ها در برنامه‌های معروف کار با تصاویر، مانند فتوشاپ آشنایی دارید).

اگر می‌خواهید مقادیر پارامترهای چرخشی تصویر نیز در اینجا اعمال شوند، می‌توان به نحو ذیل عمل کرد:

```
var resizeTrackbar = window.CreateTrackbar(
    name: "Resize", value: 1, max: 100,
    callback: pos =>
    {
        rotateImage(angle, scale, src, dst);
        Cv2.Resize(dst, dst,
            new Size(src.Width + pos, src.Height + pos),
            interpolation: Interpolation.Cubic);
        window.Image = dst;
    });
```

در این کد ابتدا تصویر اصلی چرخش یافته و سپس در متد Resize از این تصویر چرخش یافته، به عنوان src استفاده می‌شود (هر دو پارامتر متد Resize به dst تنظیم شده‌اند).



مات کردن تصاویر

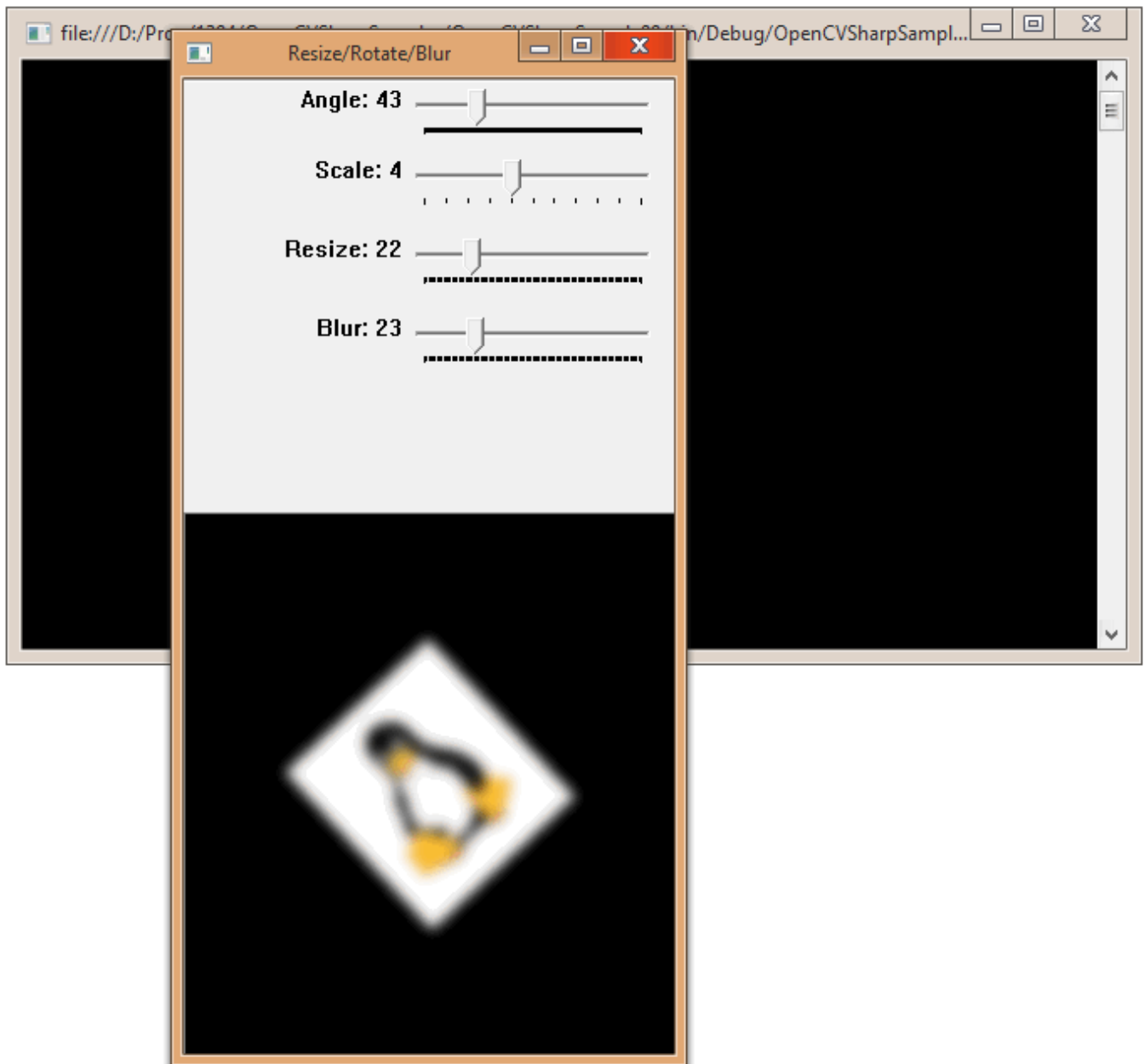
در OpenCV با استفاده از متدهای [GaussianBlur](#) و یا [medianBlur](#) ، می‌توان تصاویر را مات کرد که نمونه‌ای از آن‌را در ادامه ملاحظه می‌کنید:

```
var blurTrackbar = window.CreateTrackbar(
    name: "Blur", value: 1, max: 100,
    callback: pos =>
    {
        if (pos % 2 == 0) pos++;

        rotateImage(angle, scale, src, dst);
        Cv2.GaussianBlur(dst, dst, new Size(pos, pos), sigmaX: 0);
        window.Image = dst;
    });
```

در اینجا ابتدا تصویر اصلی به متد چرخش تصویر ارسال شده و نتیجه‌ی آن در متد `GaussianBlur` استفاده خواهد شد. اندازه‌ی مشخص شده‌ی در این متد باید توسط اعداد فرد تعیین گردد. پارامتر `sigmaX` به معنای standard deviation در جهت `x` است و

اگر صفر تعیین شود، برای محاسبه‌ی آن از پارامتر اندازه‌ی تعیین شده کمک گرفته خواهد شد.



کدهای کامل این مثال را [از اینجا](#) می‌توانید دریافت کنید.