深圳大学
SHENZHEN UNIVERSITY

计算机视觉

计算机视觉所

深圳大学
计算机与软件学院

主要内容

1. 图像操作
2. 图像变换
3. 图像滤波
4. 图像阈值
5. 上机实验

图像操作

基本数据结构：

cv::Mat　//矩阵

cv::Point //点坐标　例：cv::Point3d p(x0, x1, x2);

cv::Scalar //值向量 例：cv::Scalar red(0, 0, 255);

cv::Size　//大小向量 例：cv::Size sz(w, h);

cv::Rect　//矩形向量 例：cv::Rect rt(x, y, w, h);

cv::vec_ //向量 例：Vec2f v2f(x0,x1);

图像操作

图像基本信息

```
img_gray.rows; // height of image
img_gray.cols; // width of image
img_gray.channels(); // number of channels
```

图像像素

```
img_src.at<uchar>(y, x); //gray image
img_src.at<cv::Vec3b>(y, x); //color image
```

图像区域（ROI）

```
img_rgb(cv::Rect(105, 18, 50, 50)); // clip image
img_roi(roi_rect).setTo(cv::Scalar(255, 255, 255)); //roi
```

图像操作

基本运算

最大，最小，均值，方差
求和，非零数
加、减、乘、点乘
等其矩阵操作

cv::addWeighted(roi1, alpha, roi2, beta, 0.0, roi1);

图像变换

图像缩放

void cv::**resize** (InputArray src, OutputArray dst, Size dsize, double fx = 0, double fy = 0, int interpolation =INTER_LINEAR)

图像镜像

void cv::**flip**(InputArray src, OutputArray dst, int flipCode)

$$\texttt{dst}_{ij} = \begin{cases} \texttt{src}_{\texttt{src.rows}-i-1,j} & if\ \texttt{flipCode} = 0 \\ \texttt{src}_{i,\texttt{src.cols}-j-1} & if\ \texttt{flipCode} > 0 \\ \texttt{src}_{\texttt{src.rows}-i-1,\texttt{src.cols}-j-1} & if\ \texttt{flipCode} < 0 \end{cases}$$

图像变换

仿射变换（目标校正）
cv::warpAffine()

视角变换（相机标定）
cv::warpPerspective():

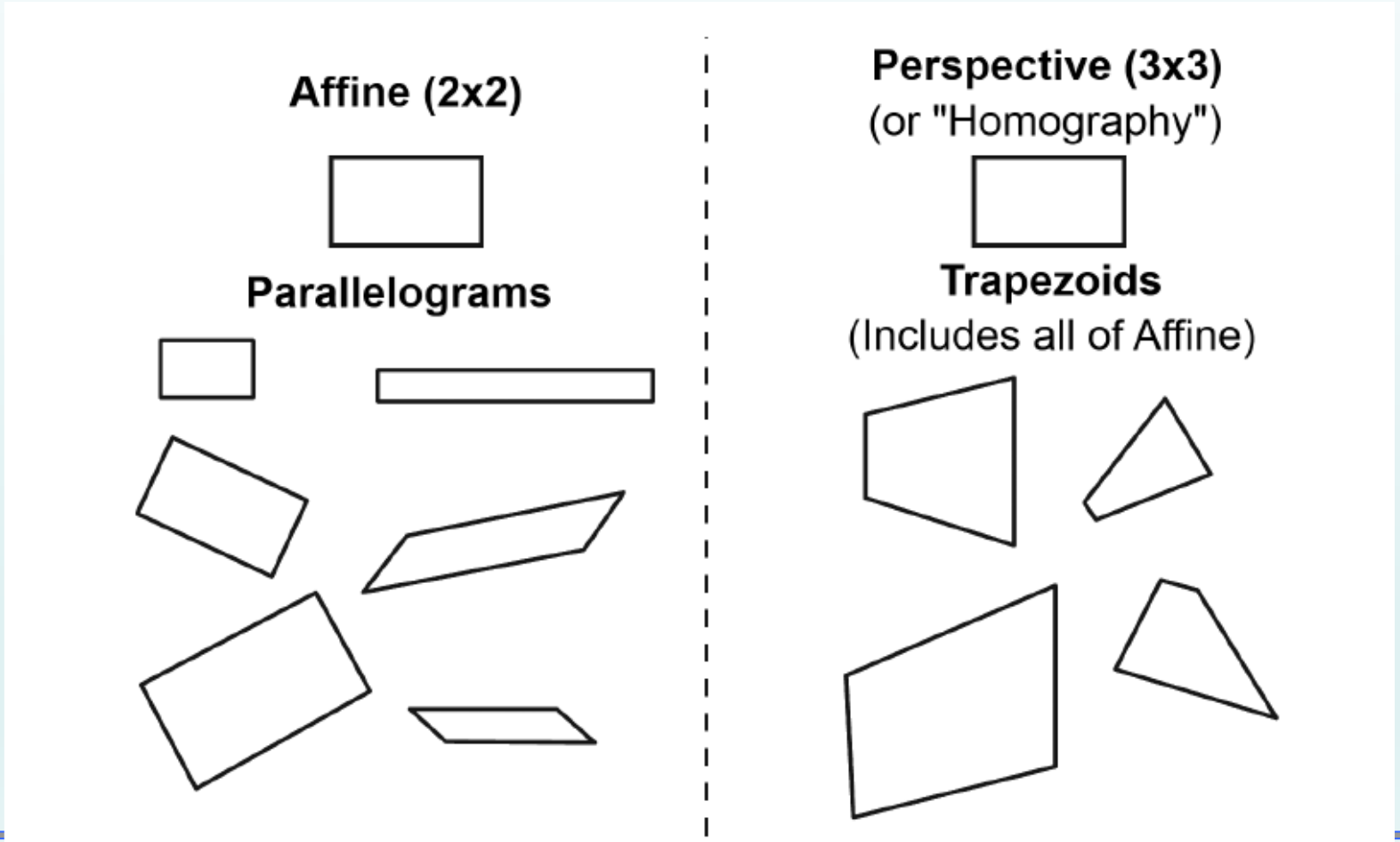


Figure 11-3. Affine and perspective transformations

图像变换

旋转（特殊的仿射变换）

void cv::**warpAffine** (InputArray src, OutputArray dst, InputArray M, Size dsize, int flags = INTER_LINEAR, int borderMode = BORDER_CONSTANT, const Scalar & borderValue = Scalar())

旋转矩阵

Mat cv::getRotationMatrix2D (Point2f center, double angle, double scale)

Ex1：

```
cv::Mat rot_mat = cv::getRotationMatrix2D(center, rotate_degree, 1.0);
cv::warpAffine(img_src, img_rotate, rot_mat, src_sz);
```

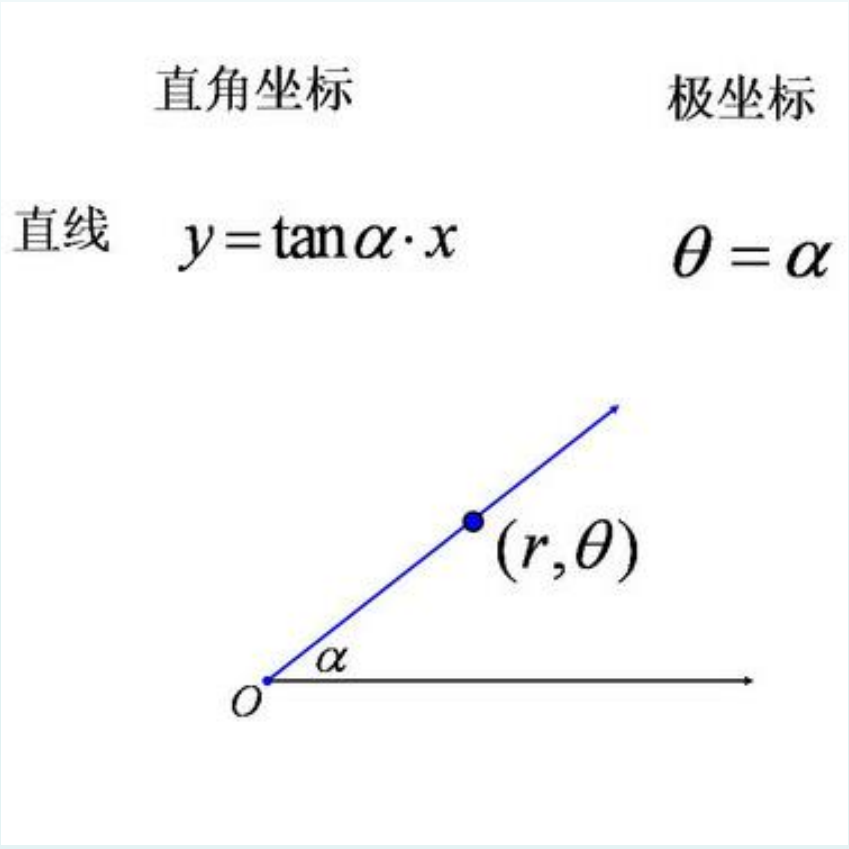
Ex2：

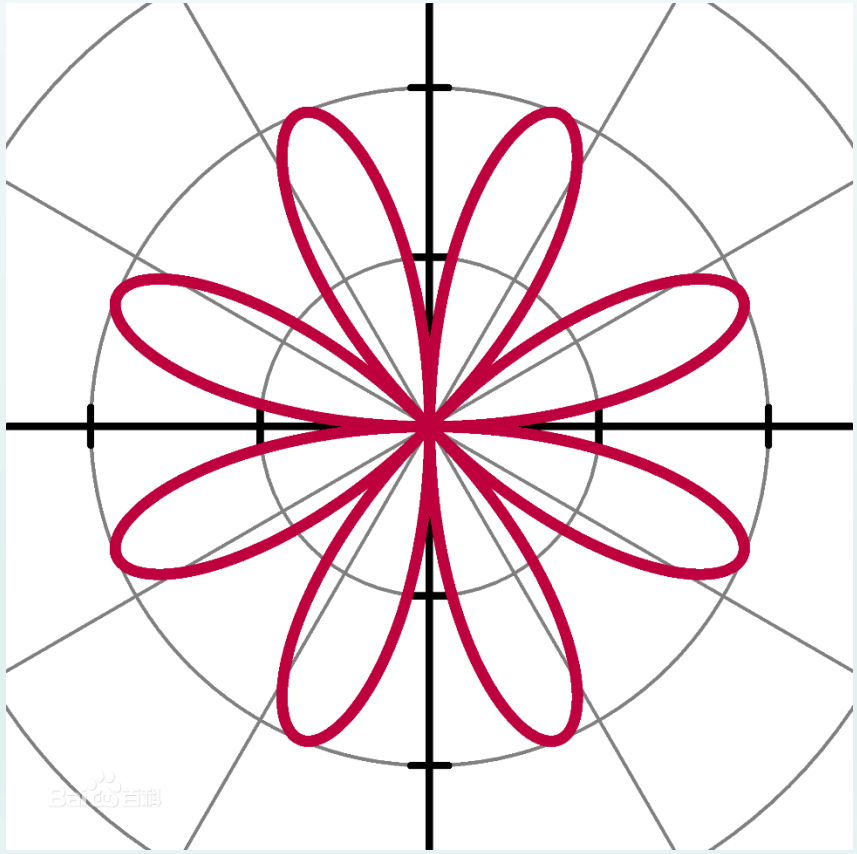
```
cv::Rect2f bbox = cv::RotatedRect(cv::Point2f(), img_src.size(), rotate_degree).boundingRect();
// adjust transformation matrix--translation transformation
rot_mat.at<double>(0, 2) += bbox.width / 2.0 - img_src.cols / 2.0;
rot_mat.at<double>(1, 2) += bbox.height / 2.0 - img_src.rows / 2.0;
cv::warpAffine(img_src, img_urotate, rot_mat, bbox.size());
```

图像变换

对数极坐标（字符识别，形状分析，缺陷检测）

void cv::logPolar (InputArray src, OutputArray dst, Point2f center, double M, int flags)

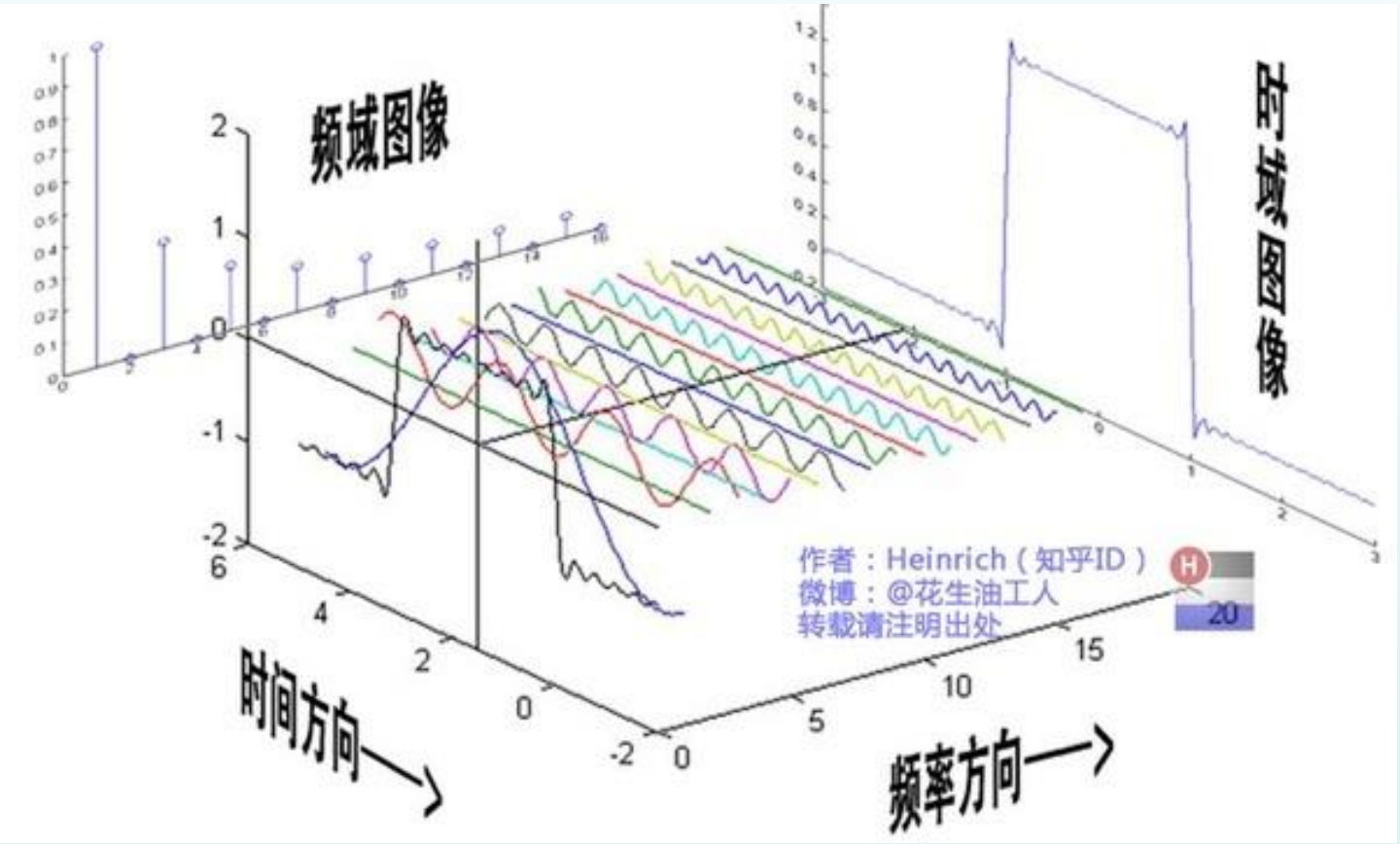


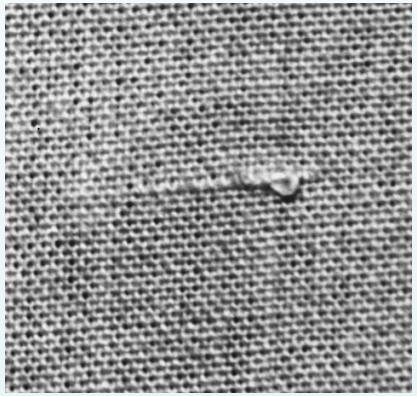


方程为 r（θ） = 2 sin 4θ的玫瑰线

图像变换

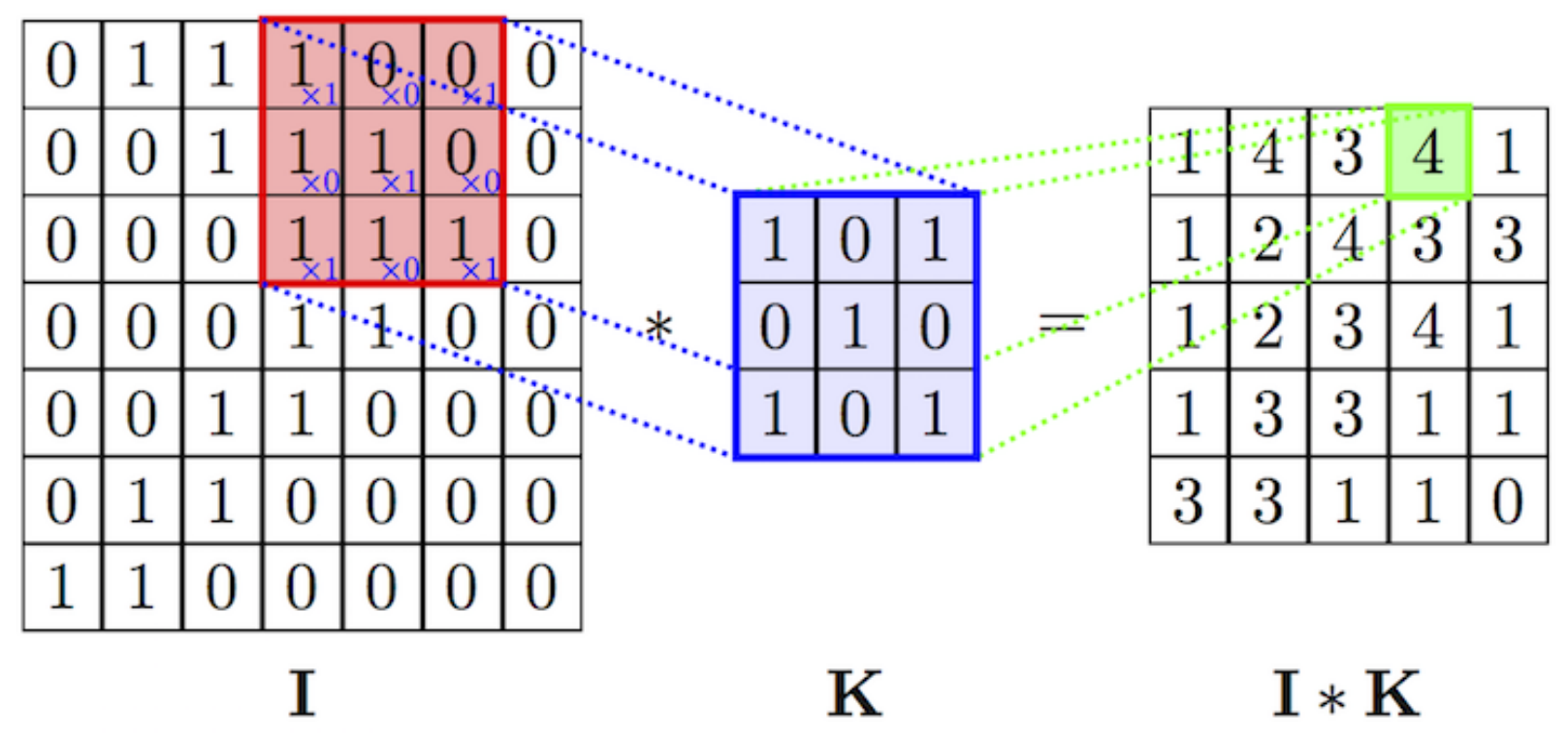
Fourier变换（去噪，缺陷检测，显著性分析）





图像滤波

空间卷积及卷积核（局部操作）



卷积核：大小，值

图像滤波

空间卷积滤波器（去噪）

均值滤波

$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

void cv::blur(InputArray src, OutputArray dst, Size ksize, Point anchor = Point(-1,-1), int borderType = BORDER_DEFAULT)

中值滤波

void cv::medianBlur(InputArray src, OutputArray dst, int ksize)

高斯滤波

void cv::GaussianBlur(InputArray src, OutputArray dst, Size ksize, double sigmaX, double sigmaY = 0, int borderType = BORDER_DEFAULT)

双边滤波

void cv::bilateralFilter(InputArray src, OutputArray dst, int d, double sigmaColor, double sigmaSpace, int borderType = BORDER_DEFAULT)

图像滤波

形态学滤波器（OCR，指纹识别）

void cv::**morphologyEx**(InputArray src, OutputArray dst, int op, InputArray kernel, Point anchor = Point(-1,-1), int iterations = 1, int borderType = BORDER_CONSTANT, const Scalar & borderValue = morphologyDefaultBorderValue())

```
enum  cv::MorphTypes {
        cv::MORPH_ERODE = 0,
        cv::MORPH_DILATE = 1,
        cv::MORPH_OPEN = 2,
        cv::MORPH_CLOSE = 3,
        cv::MORPH_GRADIENT = 4,
        cv::MORPH_TOPHAT = 5,
        cv::MORPH_BLACKHAT = 6,
        cv::MORPH_HITMISS = 7
      }
```

```
enum  cv::MorphShapes {
        cv::MORPH_RECT = 0,
        cv::MORPH_CROSS = 1,
        cv::MORPH_ELLIPSE = 2
      }
```

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

图像阈值（目标分割）

单阈值

$$dst(x,y)=\begin{cases}255, & src(x,y)>T\\ 0, & src(x,y)\le T\end{cases}$$

多阈值

$$dst(x,y)=\begin{cases}255, & src(x,y)>T_1\\ 0, & T_2\le src(x,y)\le T_1\\ 255, & src(x,y)<T_2\end{cases}$$

自适应阈值

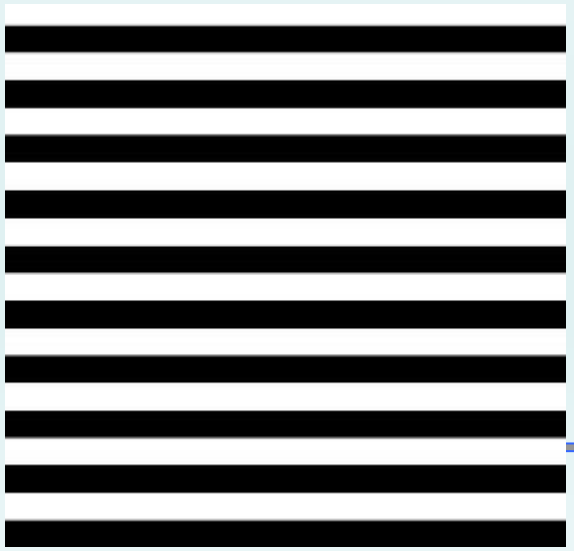
$$dst(x,y)=\begin{cases}255, & R(src(x,y))>T\\ 0, & otherwise\end{cases}$$

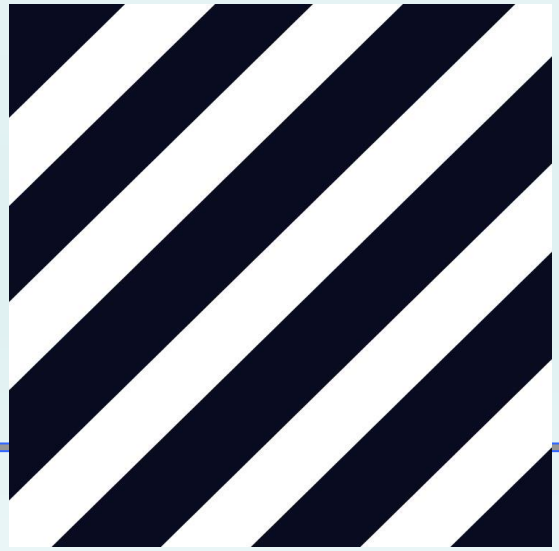
上机实验

1. 图像统计（ex-02-1）
2. 图像旋转（ex-02-2）
3. 图像去噪（ex-02-3）
4. 图像阈值（ex-02-4）
5. 形态学运算（ex-02-5）

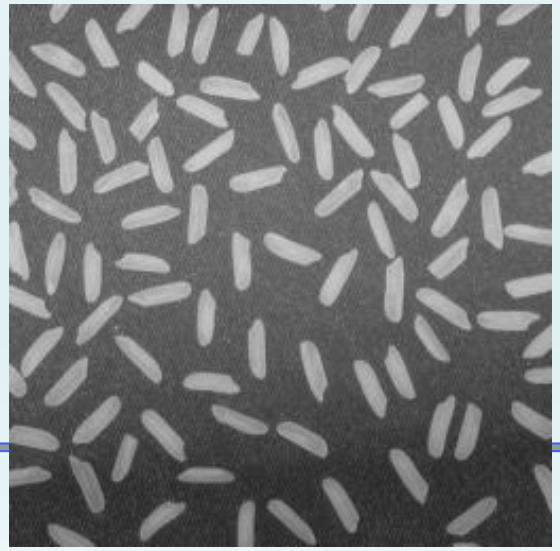
附加实验

1. Fourier变换





2. 目标计数



选题

1. 截屏软件
2. 平面目标校正
3. 环型文字识别
4. 人脸自动替换
5. 目标计数