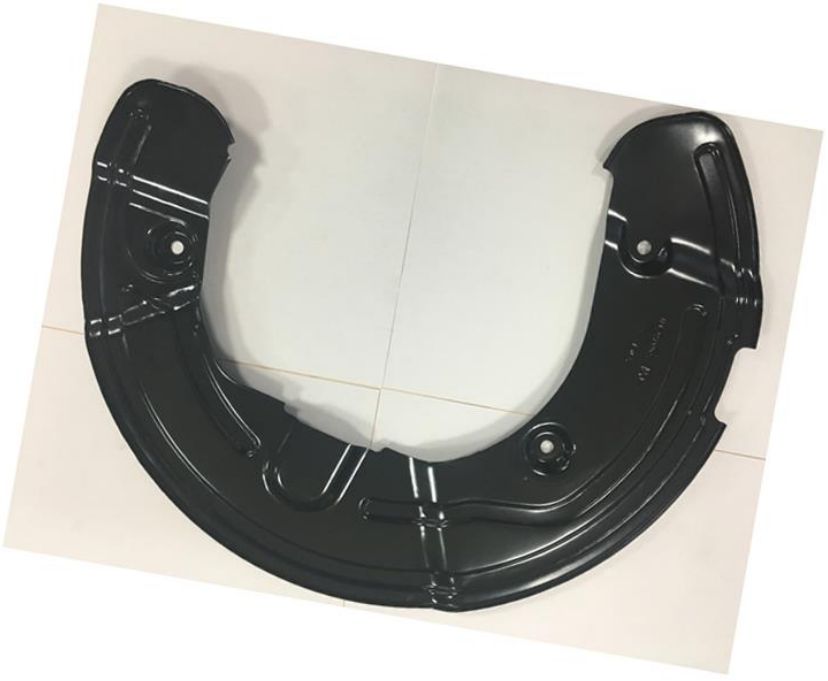
第四章

目标检测

案例分析：零件瑕疵检测

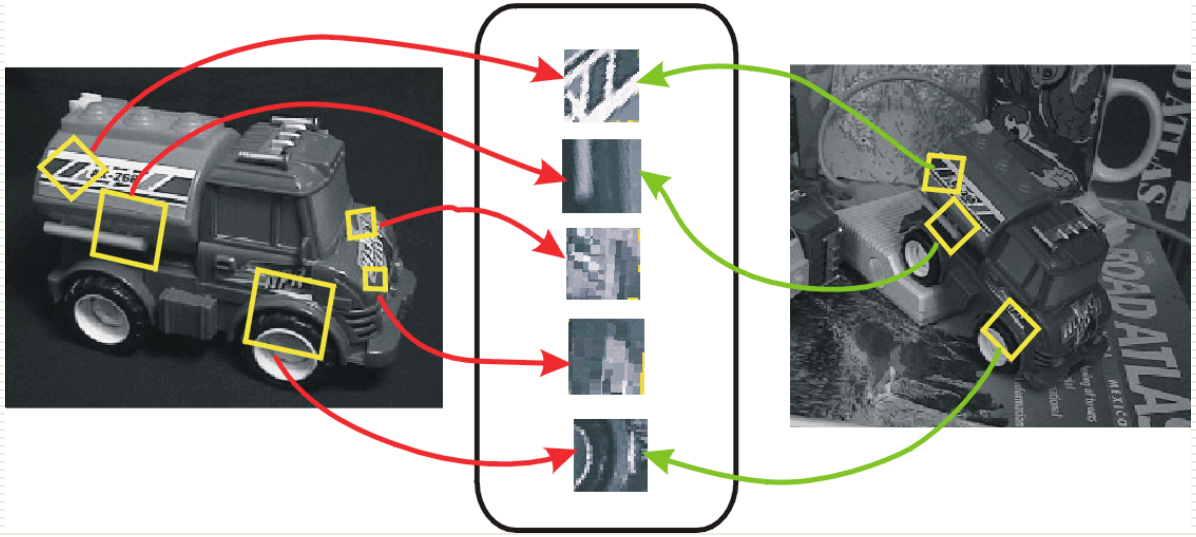
标准零件图像



实时采集零件图像

基于特征点匹配的目标检测

基于特征点的匹配方法主要有SIFT(Scale Invariant Features Transform，尺度旋转不变图像特征)、SURF(Speeded Up Robust Features)、ORB (Oriented Robust Brief) 等。



- 将一幅图像映射（变换）为一个局部特征向量集；
- 特征向量具有平移、缩放、旋转不变性，同时对光照变化、仿射及投影变换也有一定不变性。

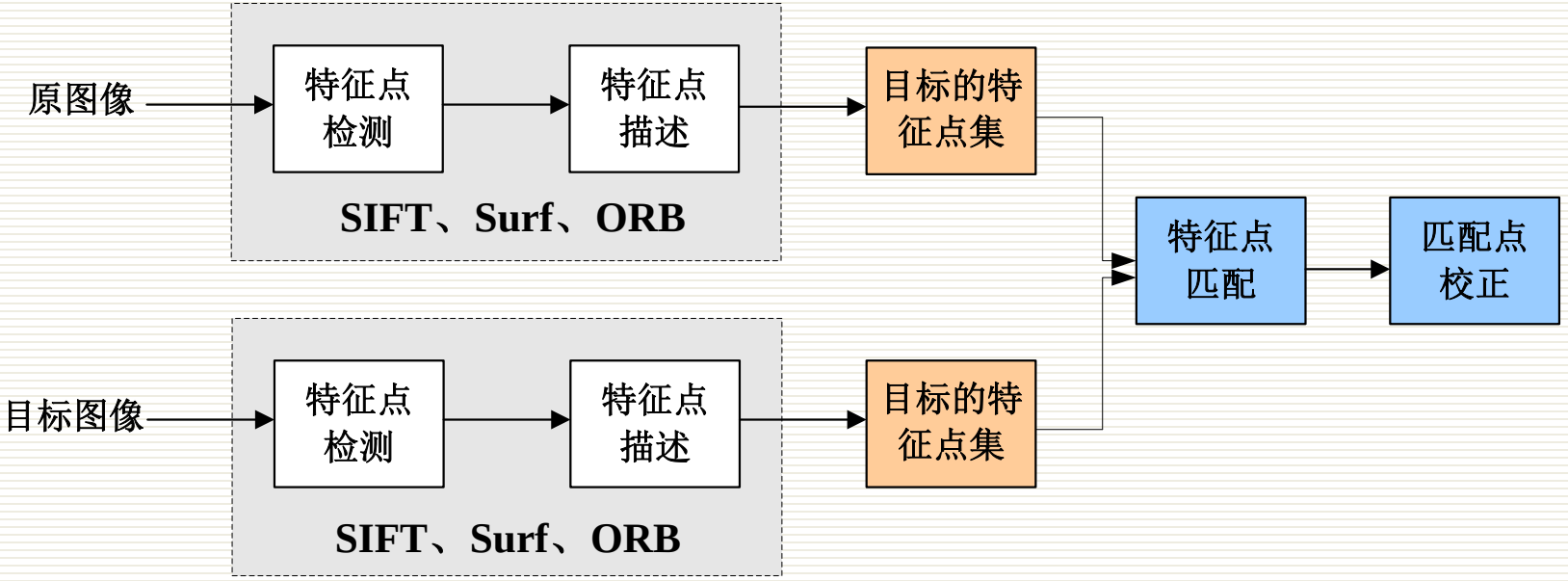
基于特征点匹配的目标检测

特征点匹配方法主要有三大步骤：

1、提取关键点；

2、对关键点附加描述信息（特征向量）；

3、通过两组特征点（附带上特征向量的关键点）的两两比较找出**相互匹配**的若干对特征点，也就建立了图像间的对应关系。



基于特征点匹配的目标检测

SIFT使用方法：

```
Ptr<SIFT> siftdetector = SIFT::create();
vector<KeyPoint> kp1, kp2;
siftdetector->detect(src1, kp1);
siftdetector->detect(src2, kp2);

Mat des1, des2;
siftdetector->compute(src1, kp1, des1);
siftdetector->compute(src2, kp2, des2);

BFMatcher matcher(NORM_L2,true);
vector<DMatch> matches;
matcher.match(des1,des2,matches);
```

基于特征点匹配的目标检测

ORB使用方法：

```
vector<KeyPoint> keypoints1, keypoints2;
Mat descriptors1, descriptors2;
Ptr<ORB> orb = ORB::create();

orb->detect(img1, keypoints1);
orb->detect(img2, keypoints2);

orb->compute(img1, keypoints1, descriptors1);
orb->compute(img2, keypoints2, descriptors2);

BFMatcher matcher(NORM_HAMMING,true);
vector<DMatch> matches;
matcher.match(obj_descriptors, scene_descriptors, matches);
```

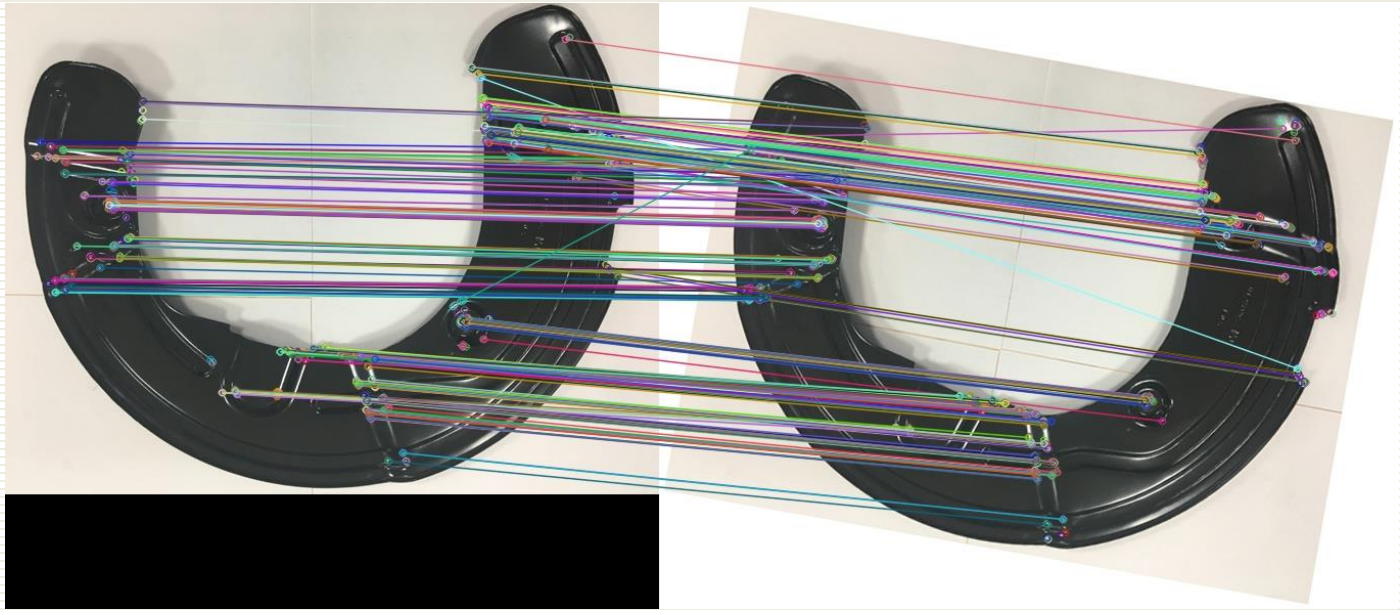
基于特征点匹配的目标检测

BRISK使用方法：

```
Ptr<BRISK> detector = BRISK::create();
vector<KeyPoint> kp1, kp2;

detector->detect(src1, kp1);
detector->detect(src2, kp2);

Mat des1, des2;
detector->compute(src1, kp1, des1);
detector->compute(src2, kp2, des2);

BFMatcher matcher(NORM_HAMMING,true);
vector<DMatch> matches;
matcher.match(obj_descriptors, scene_descriptors, matches);
```

ORB特征点检测

ORB特征点检测结果



采用RANSAC方法

滤除误匹配点后

ORB特征点检测

RANSAC(RANdom SAmple Consensus随机采样一致性算法)，是在一组含有“外点”的数据中，不断迭代，最终正确估计出最优参数模型的算法。

基本思想：

- 在样本N中随机采样K个点
- 对K个点进行模型拟合
- 计算其它点到该拟合模型的距离，并设置阈值，若大于阈值为外点舍弃，小于阈值为内点，统计内点个数。阈值为经验值，由具体应用和数据集决定。
- 以新的内点为基础，再次进行步骤2，得到新的拟合模型，迭代M次，选择内点数最多的模型，即为最优模型。

ORB特征点检测

H单应性矩阵描述两个平面的映射关系，平面中点的坐标是二维的。在特征匹配中，最终要得到一个3*3的单应性矩阵。通常令h33=1来归一化矩阵，因此单应性矩阵有8个自由度h11-h32，求这八个未知数，至少要包含四个匹配点对。

$$s\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

其中(x,y)表示目标图像角点位置,(x′ ,y′)为场景图像角点位置，s为尺度参数。

findHomography：计算多个二维点对之间的最优单映射变换矩阵 H，使用最小均方误差或者RANSAC方法函数功能：找到两个平面之间的转换矩阵。

```
Mat cv::findHomography (
InputArray srcPoints,
InputArray dstPoints,
int        method = 0,
double  ransacReprojThreshold = 3,
OutputArray  mask = noArray(),
const int         maxIters = 2000,
const double    confidence = 0.995 )
```

ORB特征点检测

具体步骤：

- 首先在得到的匹配点中，确定匹配点对(不共线)，其它匹配点为外点。
- 根据两组匹配点计算单应性矩阵。
- 根据此矩阵计算其它匹配点与该模型的投影误差，设置阈值，若小于为新内点，若大于则为外点，即误匹配对。因此通过计算出的单应性矩阵，就能实现一次误匹配点的剔除。
- 将所有的内点统计得到新的单应性矩阵，在此基础上再次进行步骤3，迭代M次，最终得到含有内点最多的模型，此时模型为最优模型，即最终所需要的单应性矩阵。

ORB特征点检测

检测出目标

两组点集获取单应性矩阵

第一组点投影到图像上，滤除错误点

画出目标框

？

图像校正

瑕疵检测

```
H12 = findperspectiveTransform(Mat(points1), points1t, H12);nsacReprojThreshold)
//确定两组点的单应性矩阵
```

编程作业

1、编写一种特征点匹配方法实现零件图像瑕疵检测

零件图像ORB特征提取结果：